

Manual POLYRATE – version 9.4.1

José C. Corchado, Yao-Yuan Chuang, Patton L. Fast, Wei-Ping Hu, Yi-Ping Liu, Gillian C. Lynch, Kiet A. Nguyen, Charles F. Jackels, Antonio Fernandez Ramos, Benjamin A. Ellingson, Benjamin J. Lynch, Vasilios S. Melissas, Jordi Villà, Ivan Rossi, Elena L. Coitino, Jingzhi Pu, Titus V. Albu
Department of Chemistry and Supercomputer Institute
University of Minnesota, Minneapolis, Minnesota 55455

Rozeanne Steckler
San Diego Supercomputer Center
P.O. Box 85608, San Diego, California 92138-5608

Bruce C. Garrett
Environmental Molecular Sciences Laboratory
Pacific Northwest National Laboratory, Richland, Washington 99352

Alan D. Isaacson
Department of Chemistry and Biochemistry
Miami University, Oxford, Ohio 45056

and Donald G. Truhlar
Department of Chemistry and Supercomputer Institute
University of Minnesota, Minneapolis, Minnesota 55455

Program version: 9.4.1
Program version date: June 20, 2006
Manual version date: June 20, 2006
Copyright 1989 – 2006

Abstract: POLYRATE is a computer program for the calculation of chemical reaction rates of polyatomic species (and also atoms and diatoms as special cases). POLYRATE–version 9.4.1 is an enhanced version of POLYRATE with improved capabilities for direct dynamics and curvilinear coordinates. The methods used are variational or conventional transition state theory and multidimensional semiclassical approximations for tunneling and nonclassical reflection. Generalized transition state theory calculations may be carried out using rectilinear coordinates, non-redundant curvilinear coordinates, or redundant curvilinear coordinates. Zero-curvature, centrifugal-dominant small-curvature, and large-curvature-version 3 or 4 (LC3 or LC4) tunneling methods are included, including tunneling into excited states in the LC3 and LC4 models, and optimized multidimensional tunneling is also included.

Rate constants may be calculated for canonical or microcanonical ensembles or for specific vibrational states of selected modes with translational, rotational, and other vibrational modes treated thermally. Bimolecular and unimolecular reactions and gas-phase, solid-state, and gas-solid-interface reactions are all included.

Potential energy surfaces may be analytic functions evaluated by subroutines, or they may be implicit surfaces defined by electronic structure input files containing energies, gradients, and force constants (Hessians) at selected points on a reaction path. The use of electronic structure calculations to calculate rate constants or other dynamical quantities without an analytic potential energy surface is called direct dynamics. Analytic surfaces may be used for variational transition state theory and any of the types of tunneling calculations, single-level and dual-level calculations based solely on electronic structure input files may be used for variational transition state theory and zero-curvature or small-curvature tunneling, and dual-level calculations based on using an analytic potential energy surface as the lower level and using an electronic structure input file as the higher level may be used for variational transition state theory and any of the types of tunneling calculations.

Reaction paths may be calculated by the Euler steepest-descents, Euler stabilization, Page-McIver, or variational reaction path algorithms. Vibrations away from the reaction path may be defined by rectilinear, nonredundant curvilinear, or redundant curvilinear coordinates. Generalized-transition-state dividing surfaces may be defined on the basis of gradient directions or by the re-orientation of the dividing surface algorithm.

POLYRATE supports six options for direct dynamics, namely (i) straight single-level direct dynamics, (ii) zero-order interpolated variational transition state theory (IVTST-0), (iii) first-order interpolated variational transition state theory (IVTST-1), (iv) interpolated variational transition state theory by mapping (IVTST-M), (v) variational transition state theory with interpolated single-point energies (VTST-ISPE), and (vi) variational transition state theory with interpolated optimized corrections (VTST-IOC). VTST-IOC and VTST-ISPE are examples of dual-level direct dynamics; and VTST-IOC is also called triple-slash dynamics (*///*), where the triple slash denotes higher-order corrections of the geometries at which energetic and Hessian corrections are calculated. Dual-level methods may be applied with electronic structure data for both levels (dual-level direct dynamics), or it may be applied with an analytical potential energy surface for the lower level and electronic structure data for the higher level. When dual-level methods are

employed with electronic structure data for both levels, the lower level may be either straight direct dynamics or IVTST-M.

This version of POLYRATE contains 96 test runs, and 38 of these are for direct dynamics calculations; 88 of the test runs are single-level runs, and 8 are dual-level calculations.

CONTENTS

TITLE PAGE AND ABSTRACT	1
CONTENTS	4
1. INTRODUCTION	12
2. REFERENCES	20
2.A. POLYRATE REFERENCES.....	20
2.B. REFERENCES FOR INCORPORATED CODE	21
3. AVAILABILITY AND LICENSING	22
4. POLYRATE SPECIFICS.....	23
4.A. POLYRATE authors	23
4.B. Hardware platforms	24
4.C. Compiler versions.....	27
5. PROGRAM AND MANUAL DESCRIPTION	29
5.A. Code description, portability, and special files.....	29
5.A.1. DATTIM subprogram	29
5.A.2. INCLUDE files	30
5.B. Description of the manual.....	32
5.B.1. Referencing conventions.....	33
5.B.2. Book chapters	33
6. THEORETICAL BACKGROUND FOR DYNAMICAL CALCULATIONS	34
6.A. Basic theory	34
6.B. Vibrations, Rotations, and Translations.....	37
6.C. Partition functions	39
6.D. Tunneling methods.....	41
6.E. Reactant and product properties	43
6.F. Saddle point properties.....	45
6.G. Reaction path	46
6.G.1. SADDLE option	47
6.G.2. NOSADDLE option.....	50
6.H. Reaction-path curvature.....	50
6.I. Transmission coefficients.....	52
6.J. Reaction rates including tunneling.....	53
6.K. Imaginary-frequency modes transverse to the reaction path	56
6.L. Equilibrium constant calculation.....	56
6.M. Omitting the product.....	57

6.N. Optimized multidimensional tunneling calculations.....	57
6.N.1. COMT method.....	57
6.N.2. μ OMT method.....	57
6.O. Interpolated VTST and mapped interpolation	58
6.O.1. IVTST-0 and IVTST-1	58
6.O.2. Interpolated VTST by mapping (IVTST-M).....	59
6.O.3. The mapped interpolation (MI) algorithm with the hooks.....	61
6.P. Interpolated optimized corrections (VTST-IOC), a form of dual-level VTST	62
6.Q. Interpolated algorithm for large-curvature tunneling calculations.....	63
7. PROGRAM STRUCTURE.....	64
7.A. Overview of program flow	64
7.B. Determining the sign of s	69
7.C. Restart options.....	70
8. CALCULATIONS USING THE HOOKS.....	72
8.A. Methods for interpolating beyond SLM and SLP with the hooks.....	72
8.B. Zero of energy	72
8.C. User-Supplied analytic potential energy surface.....	73
8.C.1. Interface.....	73
8.C.2. Utility routines.....	77
8.D. Calculations using an electronic structure program through hooks.....	81
9. CALCULATIONS BASED ON AN ELECTRONIC STRUCTURE INPUT FILE....	86
9.A. Relation of POLYRATE input to electronic structure package output.....	88
9.B. Orientation of the molecule at generalized transition states	89
9.C. Methods for interpolation of electronic structure data with files fu30 and fu40 ..	90
9.D. Methods for interpolation of electronic structure data with file fu31	92
10. SELECTED DETAILS OF THE CALCULATIONS	93
10.A. Extrapolation of the reaction-path data beyond the range of the original grid...	93
10.B. Options for anharmonicity	96
10.B.1. WKB anharmonicity based on a quadratic-quartic fit	96
10.B.2. WKB anharmonicity based on the true potential.....	97
10.B.3. Hindered rotor treatment	98
10.C. Alternate method for the calculation of numerical second derivatives	99
10.D. Reaction-path following methods	99
10.E. Interpolation of small-curvature effective reduced mass	101
10.F. Reaction-path curvature calculation.....	101
10.G. Solid-state and gas-solid interface reactions.....	102

10.H. VTST-IOC calculation, including ICR, ICL, and ICA schemes	102
10.I. VTST-ISPE calculation.....	107
10.J. Alternative methods for mapped interpolation.....	109
10.K. Quantized-reactant-state tunneling calculation.....	110
10.L. Curvilinear generalized normal mode analysis	110
10.M. IVTST0FREQ option.....	112
10.N. LCG4 option	116
11. DESCRIPTION OF INPUT	119
11.A. General description of file fu5 input	119
11.A.1. Section descriptions	121
11.A.2. GENERAL section	122
11.A.3. ENERGETICS section.....	131
11.A.4. SECOND section	134
11.A.5. OPTIMIZATION section.....	137
11.A.6. REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, AND START sections	143
11.A.7. PATH section	160
11.A.8. TUNNEL section	185
11.A.9. RATE section	194
11.B. General description of file fu29 input.....	206
11.B.1. E1GEOM Section	207
11.B.2. E1GRAD Section.....	207
11.B.3. E1HESS, P1HESS, P2HESS, R1HESS, R2HESS, and TSHESS sections.....	207
11.B.4. IVTST Section	208
11.B.5. P1FREQ, P2FREQ, R1FREQ, R2FREQ, and TSFREQ sections	209
11.C. General description of file fu30 input.....	210
11.D. General description of file fu31 input	219
11.D.1. Section descriptions	219
11.D.2 GENERAL section	221
11.D.3. REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and SADDLE sections	225
11.E. General description of file fu40 input.....	228
11.E.1. Section Descriptions.....	228
11.E.2 GEN40 section.....	229
11.E.3 R140, R240, P140, P240, WR40, and WP40 sections	235
11.E.4 SADDLE40 section.....	238

11.E.5 POINT40 section.....	240
11.F. General description of unit fu50 input	245
11.G. General description of unit fu51 input.....	256
11.G.1. Section Descriptions	256
11.G.2 ISPEGEN section.....	257
11.G.3 POINT section	260
12. GENERAL DESCRIPTION OF LCT FILES	262
13. INSTALLING AND USING POLYRATE.....	263
13.A. Instructions for a Unix environment	263
13.A.1. Installation.....	263
13.A.2. Testing.....	268
13.A.3. Using POLYRATE.....	270
13.B. Comments for expert users	273
13.C. Running in parallel using MPI	275
14. SAMPLE TEST RUNS.....	276
14.A. Chart of the sample param.inc files.....	276
14.B. Chart of the potential energy functions and data files	278
14.C. Descriptions of sample test runs.....	278
14.C.1. Test run ch5j1tr1	282
14.C.2. Test run ch5j1tr2.....	284
14.C.3. Test run ch5j1tr3.....	285
14.C.4. Test run ch5j1tr4.....	286
14.C.5. Test run ch5j1tr5.....	287
14.C.6. Test run ch5j1tr6.....	288
14.C.7. Test run ch5j1tr7.....	289
14.C.8. Test run ch5j1tr8.....	290
14.C.9. Test run ch5j1tr9.....	291
14.C.10. Test run ch5j1tr10	293
14.C.11. Test run ch5j1tr11	294
14.C.12. Test run ch5j2tr1	295
14.C.13. Test run ch5j1tr12.....	296
POLYRATE I/O FILES	296
POTENTIAL ENERGY SURFACE DATA FILE.....	296
14.C.15. Test run ch5j2tr3	298
14.C.16. Test run ch5j2itr1	299
14.C.17. Test run ch5j2itr2.....	300

14.C.18. Test run ch5fu29tr1	301
14.C.19. Test run ch5fu29tr2	302
14.C.20. Test run ch5fu30tr1	303
14.C.21. Test run ch5fu30tr2	304
14.C.22. Test run ch5fu30tr3	305
14.C.23. Test run ch5fu30tr4	306
14.C.24. Test run ch5fu30tr5	307
14.C.25. Test run ch5fu30tr6	308
14.C.26. Test run ch5fu31tr1	309
14.C.27. Test run ch5fu40tr1	310
14.C.28. Test run ch5fu40tr2	311
14.C.29. Test run ch5fu40tr3	312
14.C.30. Test run ch5fu40tr4	313
14.C.31. Test run ch5fu40tr5	314
14.C.32. Test run ch5fu40tr6	316
14.C.33. Test run ch5fu50tr1	317
14.C.34. Test run ch5fu50tr2	318
14.C.35. Test run ch5fu50tr3	319
14.C.36. Test run ch5fu51tr1	320
14.C.37. Test run ch5icfu30tr1	321
14.C.38. Test run ch5icfu30tr2	322
14.C.39. Test run ch5icfu31tr1	323
14.C.40. Test run ch5icfu40tr1	324
14.C.41. Test run ch5icfu40tr2	325
14.C.42. Test run clhbrtr1	326
14.C.43. Test run clhbrtr2	328
14.C.44. Test run clhbrtr3	329
14.C.45. Test run clhbrtr4	330
14.C.46. Test run clhbr1tr5	331
POLYRATE I/O FILES	331
14.C.48. Test run cmctr2	333
14.C.49. Test run cmctr3	334
14.C.50. Test run cwmctr1	335
14.C.51. Test run cwmctr2	336
14.C.52. Test run cwmctr3	337
14.C.53. Test run h3tr1	338

14.C.54. Test run h3tr2.....	339
14.C.55. Test run h3tr3.....	340
14.C.56. Test run h3tr4.....	341
14.C.57. Test run ho2tr1.....	342
14.C.58. Test run ho2tr2.....	343
14.C.59. Test run ho2tr3.....	344
14.C.60. Test run ho2tr4.....	345
14.C.61. Test run ho2tr5.....	346
14.C.62. Test run nh3tr1.....	347
14.C.63. Test run nh3tr2.....	348
14.C.64. Test run oh3tr1.....	349
14.C.65. Test run oh3tr2.....	350
14.C.66. Test run oh3tr3.....	351
14.C.67. Test run oh3tr4.....	352
14.C.68. Test run oh3tr5.....	353
14.C.69. Test run oh3tr6.....	354
14.C.70. Test run oh3tr7.....	355
14.C.71. Test run oh3tr8.....	356
14.C.72. Test run oh3tr9.....	357
14.C.73. Test run oh3tr10.....	358
14.C.74. Test run oh3fu30tr1.....	359
14.C.75. Test run oh3fu30tr2.....	360
14.C.76. Test run oh3fu30tr3.....	361
14.C.77. Test run oh3fu30tr4.....	362
14.C.78. Test run oh3fu30tr5.....	363
14.C.79. Test run oh3fu30tr6.....	364
14.C.80. Test run oh3fu31tr1.....	365
14.C.81. Test run oh3fu31tr2.....	366
14.C.82. Test run oh3fu40tr1.....	367
14.C.83. Test run oh3fu40tr2.....	368
14.C.84. Test run oh3fu40tr3.....	369
14.C.85. Test run oh3fu40tr4.....	370
14.C.86. Test run oh3fu40tr5.....	371
14.C.87. Test run oh3fu40tr6.....	372
14.C.88. Test run hnit1.....	373
14.C.89. Test run h2nit1.....	374

14.C.90. Test run ch4ohtr1	375
14.C.91. Test run ch4ohtr2	376
14.C.92. Test run ch4cltr1	377
14.C.93. Test run ch4cltr2	378
14.C.94. Test run ch4otr1	379
14.C.95. Test run ohcltr1	380
14.C.96. Test run ohcltr2.....	381
14.C.97. Test run ohcltr3.....	382
14.C.98. Test run ohcltr4.....	383
14.D. Summary of test runs.....	384
15. DESCRIPTION OF FILES IN RELEASE OF VERSION 9.4.1	386
15.A. FORTRAN files and subprograms, INCLUDE files, and PES subroutines.....	386
15.A.1. POLYRATE subprograms	387
15.A.2 User-supplied subprograms and INCLUDE files.....	425
15.A.3 FORTRAN files.....	425
15.B. Utility program detmi.f.....	427
15.C. Utility program hrotor.f	429
15.D. Utility programs findl.f and findb.f.....	432
15.E. Utility program rotate30.f, rotate31.f, and rotate40.f	432
16. UNIX C SHELL SCRIPTS.....	436
16.A. Installation script.....	436
16.B. Potential energy surface compiling scripts	436
16.C. Source code compiling script.....	437
16.D. Linking script.....	438
16.E. Test run scripts.....	438
17. FILE USAGE.....	439
18. MEMORY REQUIREMENTS.....	442
19. TEST RUNS AND TEST RUN TIMINGS.....	443
19.A. Extent of testing	443
19.B. Timings	443
20. FOOTNOTES TO TEXT	448
21. BIBLIOGRAPHY.....	451
22.A. Errata for "Generalized Transition State Theory" [Ref. 5, Sect. 21].....	457
22.B. Errata for " <i>Ab Initio</i> Reaction Paths and Direct Dynamics Calculations" [Ref. 20, Sect. 21].....	459

22.C. Errata for "Interpolated Variational Transition State Theory: Practical Methods for Estimating" [Ref. 26, Sect. 21].....	460
22.D. Errata for manual of version 1.1	460
22.E. Errata for manual of version 1.5.....	461
22.F. Erratum for manual of version 1.6.2.....	461
22.G. Errata for manual of version 1.7	461
22.H. Errata for manual of version 2.6	462
22.I. Erratum for manual of version 2.7.....	462
22.J. Errata for manual of version 4.0.....	462
22.K. Errata for manual of version 4.1	462
22.L. Errata for manual of version 4.5.1.....	463
22.M. Errata for manual of version 5.0.....	463
22.N. Errata for manual of version 5.0.1.....	463
22.O. Erratum for manual of version 5.1	463
22.P. Errata for manual of version 6.0.....	463
22.Q. Errata for manual of version 6.5	463
22.R. Errata for manual of version 6.6.....	464
22.S. Errata for manual of version 7.1	464
22.T. Erratum for manual of version 7.2.....	464
22.U. Erratum for manual of versions 7.8 – 7.9.1	465
22.V. Erratum for manual of versions 8.0.....	465
23. APPENDIX: POLYRATE REVISION HISTORY.....	466
23.A. Computer program revision history	466
23.B. ASCII manual revision history.....	525
23.C. PostScript and PDF manuals.....	525
23.D. Incompatibilities of POLYRATE–version 9.4.1 with previous versions.....	526
23.E. Program distribution history.....	529
23.E.1. Program libraries.....	529
23.E.2. Anonymous ftp.....	530
23.E.3. World Wide Web	530
23.E.4. Other	531
23.E.5. Current.....	531
24. ACKNOWLEDGMENTS.....	532
25. INDEX.....	533

1. INTRODUCTION

POLYRATE is a program for the analysis of reactants, products, and transition states of chemical reactions and for calculating variational transition state theory (VTST) rate constants and multidimensional semiclassical tunneling probabilities. The current version of the program can be applied to gas-phase, solid-phase, and gas-solid-interface reactions. The analysis of reactants and products can be performed for up to two reactants and two products. Generalized transition state theory calculations may be carried out using the harmonic approximation for vibrations with either rectilinear or curvilinear coordinates. A limited amount of anharmonicity may also be included. The VTST rate constants can be computed using canonical variational theory (CVT), improved canonical variational theory (ICVT), or microcanonical variational theory (μ VT). Conventional transition state theory (TST) may also be employed.

For thermal or ground-state reactions, tunneling probabilities can be computed using the zero-curvature tunneling method (ZCT), the small-curvature tunneling (SCT) method, the large-curvature tunneling (LCT) method, or an optimized multidimensional tunneling method (μ OMT or COMT). The SCT calculations are based on the centrifugal-dominant small-curvature semiclassical ground-state (CD-SCSAG) method, and the LCT calculations are based on the large-curvature ground-state method, version 3 (LCG3). For state-selected excited-state reactions only the ZCT and SCT methods are available in this version. In addition, for ground-state reactions of exoergic or thermoneutral reactions, LCT calculations may be summed over certain excited final states, and for endoergic reactions, LCT calculations may be summed over certain excited states of reactants. (These summations are a required part of the method when these contributions are not negligible). The recommended tunneling procedure when LCT calculations are carried out is the microcanonical optimized multidimensional tunneling method (μ OMT) in which, at each energy, the tunneling probability is approximated as the larger of the values calculated by the SCT and LCT methods.

The POLYRATE program computes rate constants by either single-level or dual-level approaches. The single-level calculations and the lower level of dual-level calculations may be based on either an analytic potential energy function, which is called the potential energy surface (PES) or potential energy function (PEF), or on various types of interpolations of electronic structure input data at selected points along the minimum energy path.

When using a PES, the user must supply a subroutine which, for a given set of Cartesian coordinates of each of the atoms, returns the potential energy and its first derivatives with respect to the Cartesian coordinates. This subroutine is called by a set of POLYRATE interface routines called hooks. (In particular, the “hooks” consist of subprograms PREP, PREPJ, EHOOK, GHOOK, HHOOK, and OHOOK plus a few utilities, e.g., YTRANS and YNEWT, that are discussed in Section 8.C.). When using electronic structure input data without an analytic surface, the user must supply, as an input file to the program, the geometries, energies, and force constants for the reactants, products, and saddle point, and optionally these quantities and the gradient at a sequence of points on the minimum energy path. This data is then used for dynamics calculations.

Either for reaction paths based on potential energy surfaces or for those based on electronic structure input files, the data along the reaction path may be “corrected” using higher-level data supplied by the user at selected points. A generic name for such dual-level VTST methods is VTST-IOC. We currently recognize three variants of the IC methods, namely IOC (interpolated optimized corrections), IOE (interpolated optimized energies), and ISPE (interpolated single point energies). The IOC and IOE methods are also called triple-slash (*///*) VTST. In the IOC method, the frequencies, potential energies, and moment-of-inertia determinants along the reaction path that are calculated from the potential energy surface are “corrected” using the results of higher-level electronic structure calculations (supplied by the user) at selected points. The O in IOC indicates that any corrections at stationary points are based on geometries optimized at the higher level. If corrections were made at nonstationary points, they would be based on reaction paths calculated at the higher level. The IOE approach is a subcase of IOC in which only the energies and moment-of-inertia determinants are corrected, but the corrected energies are based on geometries optimized at the higher level. The final dual-level method, called VTST-ISPE, corrects only the energies along the reaction path using higher-level electronic structure calculations (supplied by the user) at the stationary points and selected points along the path. In general the acronyms VTST-IOC and VTST-IOE or the triple slash notation is reserved for cases when the correction involves a higher-level geometry optimization of the transition state. The VTST-ISPE notation refers to using higher-level single-point energies at the lower-level saddle point or at the lower-level saddle point plus selected points along the lower-level reaction path. Thus VTST-IOE is expected to be more accurate than VTST-ISPE, even though one neglects the corrections to the frequencies.

In addition to the straight VTST method and to dual-level VTST-IOC and VTST-ISPE, the code also has single-level IVTST options. With the IVTST philosophy the user supplies the geometry, energy, and frequencies or Hessian at the stationary points and additionally at as many points along the path as are computationally feasible. POLYRATE has the functionality to perform VTST calculations with information at the stationary points and no points along the reaction path, called IVTST-0, with one point along the path on either side of the saddle point, called IVTST-1, or with an arbitrary number of points along the path on either side of the saddle point (as long as there is at least one nonstationary point on either side of the saddle point). The latter option is carried out by the mapped interpolation (MI) algorithm and is called IVTST-M. In the current version of POLYRATE, IVTST-M is available with only the default IVTST-M options when electronic structure input data is read from file fu30 or fu40, and it is available with all the IVTST-M options for files fu31 and the hooks. Note that IVTST-M can also be used as the lower level of a dual-level calculation, but in the present version of POLYRATE IVTST-0 and IVTST-1 cannot be used that way.

In addition to the use of the MI algorithm as the central element of the IVTST-M method with electronic structure input files, beginning with version 8.0 of POLYRATE we also for the use of the MI algorithm as an "extrapolation" algorithm when using the hooks for a calculation with a saddle point. If the hooks are used to interface with an electronic structure calculation, this provides an alternative way to perform IVTST-M direct dynamics calculations. If the hooks are used with an analytic potential, this replaces the previously used method for extrapolating the adiabatic potential and effective masses beyond the edge of the originally calculated grid. Finally we note that the energy part of the MI algorithm is used in the VTST-ISPE version of VTST-IC.

There is also an option, called the RESTART option, in which the input consists in part of reaction-path information calculated in a previous run of the program. The RESTART option is not available for dual-level calculations.

Derivative versions of POLYRATE, called MORATE, GAUSSRATE, GAMESOLRATE and AMSOLRATE, which compute rate constants from potentials and gradients calculated by electronic structure methods, are also available.

MORATE uses the semiempirical molecular orbital package MOPAC instead of an analytic potential energy function. The electronic structure input file option and MORATE are both examples of direct dynamics, i.e., rate or dynamics calculations based directly on electronic structure information without the intermediary of an analytic fit to the potential energy surface. The MORATE program is not distributed as part of the POLYRATE package, but it is available as a separate distribution from the U of M. At the time of release of POLYRATE–version 9.4.1 the latest version of MORATE is based on POLYRATE–version 8.5.1.

An interface to the GAUSSIAN94 or GAUSSIAN98 electronic structure packages based on POLYRATE version 9.1, called GAUSSRATE, is available as a separate distribution.

An interface to the MULTILEVEL package based on POLYRATE version 9.1, called MULTILEVELRATE, is available as a separate distribution.

An interface to the CHARMMRATE module of the CHARMM package based on POLYRATE version 9.0, called CRATE, is available as a separate distribution.

An interface to the GAMESOL2.2 solvation model package based on GAMESS and the solvation model SM5.42, called GAMESOLRATE, is available as a separate distribution.

AMSOLRATE is an interface with the AMSOL electronic structure package. AMSOLRATE can also calculate rate constants for reactions in liquid-phase solutions. AMSOLRATE can also be used for gas-phase calculations with semiempirical molecular orbital theory, but we recommend MORATE for that purpose because it produces less noisy frequencies.

An interface to the ACESII electronic structure package, called ACESRATE, is under construction.

The various direct dynamics methods obtain their input about the system in different ways:

- Single-level direct dynamics calculations may be carried out using file fu29, file fu30, file fu31, file fu40, or file fu50 for the electronic structure information.
- VTST-IOC dual-level direct dynamics calculations may be carried out using file fu30, fu31, or file fu40 for the lower-level information and file fu50 for the higher-level information.

- VTST-ISPE dual-level direct dynamics calculations may be carried out using file fu30, fu31, or file fu40 for the lower-level information and file fu51 for the higher-level information.
- Single-level calculations based on an analytic potential energy surface obtain the potential surface information from subprograms SETUP and SURF, which are called by the hooks.
- VTST-IOC dual-level calculations based on an analytic potential energy surface for the lower level obtain the potential surface information from subprograms SETUP and SURF (called by the hooks) and they obtain the higher-level electronic structure information from file fu50.
- VTST-ISPE dual-level calculations based on an analytic potential energy surface for the lower level obtain the potential surface information from subprograms SETUP and SURF (called by the hooks) and they obtain the higher-level electronic structure information from file fu51.

We note that files fu30, fu31, and fu40 accomplish essentially the same purpose. There are two differences between fu30 and fu40 is that file fu30 requires formatted data, whereas file fu40 is based on keyword input. The second difference is that there is a WRITEFU30 keyword, but again there is no WRITEFU40 keyword. File fu31 is also based on keyword input, but it differs from file fu40 in that certain IVTST-M options are supported with fu31 but not with fu40. A second difference is that there is a WRITEFU31 keyword, but there is no WRITEFU40 keyword. See also Sect. 11.D.1. At some point in the future we may stop supporting fu30, but for now we will keep it for the convenience of users of previous versions. New users are encouraged to ignore fu30 and to use fu31 instead.

Throughout this manual it will be useful to keep in mind that there are really several types of single-level VTST calculations, namely

1. Straight VTST
 - 1.A. Straight VTST through the hooks
 - 1.B. Straight VTST with file fu30 or file fu40
2. IVTST-M
 - 2.A. IVTST-M with file fu30 or fu40
 - 2.B. IVTST-M with file fu31
3. IVTST-0 or IVTST-1.

Calculations of type 1.A, 1.B, 2.A, and 2.B have many points of similarity, and in fact once POLYRATE has all the save grid arrays filled with data, these types of runs become identical. Further the MI algorithm that constitutes the central element of IVTST-M calculations (2.A or 2.B) can also be used in straight VTST (1.A or 1.B), but for straight VTST is considered as an algorithm for extending the save grid to improve numerical convergence rather than being considered as the main ingredient of the method. However

runs of type 3 do not use the save grid. In this manual we try to alert the reader whether a statement is true for one or another of these types of calculations, but not for all four types. However, the reader should be aware that sometimes we fail to mention this because it is assumed clear from the context or because the manual needs more work.

Dual-level calculations can be performed with type 1.A, 1.B, 2.A, or 2.B calculations as the lower level.

The main options are summarized in the following table:

Dynamics approach	hooks	fu29	fu30, fu31, or fu40	fu50
TST	TST	TST	TST	TST
VTST	TST/ZCT CVT CVT/ZCT CVT/SCT CVT/LCT ICVT ICVT/ZCT ICVT/SCT ICVT/LCT μ VT μ VT/ZCT μ VT/SCT μ VT/LCT		TST/ZCT CVT CVT/ZCT CVT/SCT ICVT ICVT/ZCT ICVT/SCT μ VT μ VT/ZCT μ VT/SCT	
IVTST-0		TST TST/ZCT		TST TST/ZCT
IVTST-1		TST TST/ZCT CVT CVT/ZCT CVT/SCT		
	hooks plus fu50		fu30, fu31, or fu40 plus fu50	
VTST-IOC	TST TST/ZCT CVT CVT/ZCT CVT/SCT CVT/LCT		TST TST/ZCT CVT CVT/ZCT CVT/SCT	

Table continued:

Dynamics approach	hooks plus fu50		fu30, fu31, or fu40 plus fu50	
VTST-IOC	ICVT ICVT/ZCT ICVT/SCT ICVT/LCT μ VT μ VT/ZCT μ VT/SCT μ VT/LCT		ICVT ICVT/ZCT ICVT/SCT μ VT μ VT/ZCT μ VT/SCT	
	hooks plus fu51		fu30, fu31, or fu40 plus fu51	
VTST-ISPE	ICVT ICVT/ZCT ICVT/SCT ICVT/LCT μ VT μ VT/ZCT μ VT/SCT		ICVT ICVT/ZCT ICVT/SCT μ VT μ VT/ZCT μ VT/SCT	

Note that whenever both SCT and LCT methods are available, then the μ OMT method is available as well.

Also note: When an analytical potential is used, the hooks call SETUP and SURF. However, the hooks capability is more general and they can be used to interface POLYRATE with electronic structure codes for direct dynamics; examples are the MORATE GAUSSRATE, GAMESOLRATE, and AMSOLRATE programs, which are interfaces to MOPAC, GAUSSIAN, GAMESOL, and AMSOL respectively.

2. REFERENCES

2.A. POLYRATE REFERENCES

The recommended reference for the current version of the code is to give Ref. 10. References 1, 4, and 9 are also recommended in certain cases because they provide published explanations of some of the methods used in the code. Other methods used in the code are explained in the references given in Section 21 of this manual.

1. A. D. Isaacson, D. G. Truhlar, S. N. Rai, R. Steckler, G. C. Hancock, B. C. Garrett, and M. J. Redmon, *Computer Physics Communications*, Volume 47, pp. 91-102 (1987). [POLYRATE-version 1.1]
2. D.-h. Lu, T. N. Truong, B. C. Garrett, R. Steckler, A. D. Isaacson, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, V. S. Melissas, and Donald G. Truhlar, *QCPE Bulletin*, Volume 11, pp. 13-14 (1991). [POLYRATE-version 2.5]
3. D.-h. Lu, T. N. Truong, V. S. Melissas, G. C. Lynch, Y.-P. Liu, B. C. Garrett, R. Steckler, A. D. Isaacson, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar, *QCPE Bulletin*, Volume 12, pp. 35-36 (1992). [POLYRATE-version 4.0]
4. D.-h. Lu, T. N. Truong, V. S. Melissas, G. C. Lynch, Y.-P. Liu, B. C. Garrett, R. Steckler, A. D. Isaacson, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar, *Computer Physics Communications*, Volume 71, pp. 235-262 (1992). [POLYRATE-version 4.0.1]
5. Y.-P. Liu, G. C. Lynch, W.-P. Hu, V. S. Melissas, R. Steckler, B. C. Garrett, D.-h. Lu, T. N. Truong, A. D. Isaacson, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar, *QCPE Bulletin*, Volume 13, pp. 28-29 (1993) [POLYRATE-version 5.0].
6. Y.-P. Liu, G. C. Lynch, W.-P. Hu, V. S. Melissas, R. Steckler, B. C. Garrett, D.-h. Lu, T. N. Truong, A. D. Isaacson, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar, *QCPE Bulletin*, Volume 13, p. 56 (1993) [POLYRATE-version 5.0.1].
7. R. Steckler, W.-P. Hu, Y.-P. Liu, G. C. Lynch, B. C. Garrett, A. D. Isaacson, D.-h. Lu, V. S. Melissas, T. N. Truong, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar, *POLYRATE program 6.0*, San Diego Supercomputer Center Technical Report No. GA-A21760 (1994).
8. R. Steckler, W.-P. Hu, Y.-P. Liu, G. C. Lynch, B. C. Garrett, A. D. Isaacson, D.-h. Lu, V. S. Melissas, T. N. Truong, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar, *QCPE Bulletin*, Volume 15, pp. 32-36 (1995) [POLYRATE-version 6.5].
9. R. Steckler, W.-P. Hu, Y.-P. Liu, G. C. Lynch, B. C. Garrett, A. D. Isaacson, D.-h. Lu, V. S. Melissas, T. N. Truong, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar, *Computer Physics Communications*, Volume 88, pp. 341-343 (1995). [POLYRATE-version 6.5]

10. J. C. Corchado, Y.-Y. Chuang, P. L. Fast, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, A. Fernandez Ramos, B. A. Ellingson, B. J. Lynch, V. S. Melissas, J. Villà, I. Rossi, E. L. Coitiño, J. Pu, T. V. Albu, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar, POLYRATE–version 9.4, University of Minnesota, Minneapolis, 2005.

2.B. REFERENCES FOR INCORPORATED CODE

1. The code for evaluating the Cartesian **B** matrix and **C** tensor for stretches, nondegenerate bends, and torsions for the general polyatomic case is from J. Cioslowski and M. Challacombe, ANHTRAX, QCPE Program No. 609 (Quantum Chemistry Program Exchange, Bloomington, IN, 1991).
2. The EF code for stationary point optimization is from J. Backer, F. Jensen, H. S. Rzepa, A. Stebbings, LOCATION OF TRANSITION STATES IN AMPAC AND MOPAC USING EIGENVECTOR FOLLOWING (EF), QCPE Program No. 597 (Quantum Chemistry Program Exchange, Bloomington, IN, 1990).

3. AVAILABILITY AND LICENSING

All versions are copyrighted.

Versions 1.6 and later are licensed. Licenses are available from authorized distributors. Or the program may be used in the installed and supported version at San Diego Supercomputer Center, which has a general site license. All license fees are waived for version 9.4.1, but the license must be signed. Copies of the license and instructions for obtaining the code are available at the U of M World Wide Web site, <http://comp.chem.umn.edu/polyrate>.

POLYRATE distribution at the U of M is currently handled by

Adam Chamberlin
Department of Chemistry
University of Minnesota
207 Pleasant Street SE
Minneapolis, MN 55455-0431
e-mail: software@comp.chem.umn.edu
<http://comp.chem.umn.edu/polyrate>
fax: (612) 626-9390

4. POLYRATE SPECIFICS

4.A. POLYRATE authors

Versions 1.0 - 1.1:	A. D. Isaacson, D. G. Truhlar, S. N. Rai, R. Steckler, G. C. Hancock, B. C. Garrett, and M. J. Redmon
Versions 1.5–1.5.2:	A. D. Isaacson, D. G. Truhlar, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. N. Truong, T. Joseph, B. C. Garrett, and R. Steckler
Versions 1.6–1.6.3:	A. D. Isaacson, D. G. Truhlar, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. N. Truong, T. Joseph, D.-h. Lu, B. C. Garrett, and R. Steckler
Versions 1.7–1.7.1:	A. D. Isaacson, D. G. Truhlar, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. N. Truong, T. Joseph, D.-h. Lu, V. S. Melissas, B. C. Garrett, and R. Steckler
Versions 2.0–2.5.2:	D.-h. Lu, T. N. Truong, B. C. Garrett, R. Steckler, A. D. Isaacson, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, V. S. Melissas, and D. G. Truhlar
Versions 2.6–3.0.1:	D.-h. Lu, T. N. Truong, V. S. Melissas, B. C. Garrett, R. Steckler, A. D. Isaacson, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar
Versions 3.0.2–4.5.1:	D.-h. Lu, T. N. Truong, V. S. Melissas, G. C. Lynch, Y.-P. Liu, B. C. Garrett, R. Steckler, A. D. Isaacson, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar
Versions 5.0–5.1:	Y.-P. Liu, G. C. Lynch, W.-P. Hu, V. S. Melissas, R. Steckler, B. C. Garrett, D.-h. Lu, T. N. Truong, A. D. Isaacson, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar
Versions 6.0–6.5:	R. Steckler, W.-P. Hu, Y.-P. Liu, G. C. Lynch, B. C. Garrett, A. D. Isaacson, D.-h. Lu, V. S. Melissas, T. N. Truong, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar
Version 6.6:	R. Steckler, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, M. Z. Gu, I. Rossi, B. C. Garrett, A. D. Isaacson, D.-h. Lu, V. S. Melissas, T. N. Truong, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar
Version 6.6.1:	R. Steckler, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, M. Z. Gu, I. Rossi, Y.-Y. Chuang, B. C. Garrett, A. D. Isaacson, D.-h. Lu, V. S. Melissas, T. N. Truong, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar

- Version 7.0–7.1: R. Steckler, Y.-Y. Chuang, E. L. Coitiño, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, M. Z. Gu, I. Rossi, P. Fast, S. Clayton, V. S. Melissas, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar
- Version 7.1.1–7.2: R. Steckler, Y.-Y. Chuang, E. L. Coitiño, P. L. Fast, J. C. Corchado, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, M. Z. Gu, I. Rossi, S. Clayton, V. S. Melissas, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar
- Version 7.3–7.4.3: R. Steckler, Y.-Y. Chuang, P. L. Fast, E. L. Coitiño, J. C. Corchado, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, M. Z. Gu, I. Rossi, S. Clayton, V. S. Melissas, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar
- Version 7.8–7.9.1: J. C. Corchado, Y.-Y. Chuang, P. L. Fast, Jordi Villà, E. L. Coitiño, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, M. Z. Gu, I. Rossi, S. Clayton, V. S. Melissas, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar
- Version 8.0: Y.-Y. Chuang, J. C. Corchado, P. L. Fast, Jordi Villà, E. L. Coitiño, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, M. Z. Gu, I. Rossi, S. Clayton, V. S. Melissas, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar
- Version 8.1–8.5.1: Y.-Y. Chuang, J. C. Corchado, P. L. Fast, Jordi Villà, W.-P. Hu, Y.-P. Liu, G. C. Lynch, C. F. Jackels, K. A. Nguyen, M. Z. Gu, I. Rossi, E. L. Coitiño, S. Clayton, V. S. Melissas, B. L. Lynch, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar
- Version 8.6–8.7.2: J. C. Corchado, Y.-Y. Chuang, P. L. Fast, Jordi Villà, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, V. S. Melissas, B. J. Lynch, I. Rossi, E. L. Coitiño, A. Fernandez-Ramos, J. Pu, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar
- Version 8.7.3–9.3: J. C. Corchado, Y.-Y. Chuang, P. L. Fast, Jordi Villà, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, V. S. Melissas, B. J. Lynch, I. Rossi, E. L. Coitiño, A. Fernandez-Ramos, J. Pu, T. V. Albu, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar
- Version 9.3.1–9.4.1: J. C. Corchado, Y.-Y. Chuang, P. L. Fast, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, A. Fernandez Ramos, B. A. Ellingson, B. J. Lynch, V. S. Melissas, J. Villà, I. Rossi, E. L. Coitiño, J. Pu, T. V. Albu, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar

4.B. Hardware platforms

Versions 1.0–1.5: VAX 11/780 (VMS 4.3), Cray-2 (UNICOS)

- Versions 1.6–1.7: VAX 11/780 (VMS 4.3), Cray-2 and Cray X-MP (both UNICOS)
- Version 2.0: VAX 11/780 (VMS 4.3), Cray-2 and Cray X-MP (both UNICOS 5.0), Cray Y-MP (UNICOS 5.1.8)
- Version 2.1: VAX 11/780 (VMS 4.3), Cray-2 (UNICOS 5.1.8), Cray X-MP (UNICOS 5.1)
- Version 2.5: VAX 11/780 (VMS 4.3), Cray-2 (UNICOS 5.1.9), Cray X-MP (UNICOS 5.1), Cray Y-MP (UNICOS 5.1.10), IBM 3090 600J (AIX), IBM RS/6000 model 530 (AIX), Silicon Graphics IRIS-4D/320GTX (IRIX System V.3), Sun SparcStation 4/360 (SunOS 4.0.3)
- Version 2.6: VAX 11/780 (VMS 4.3), Cray-2 (UNICOS 5.1.9), Cray X-MP-EA (UNICOS 5.1.9), IBM 3090 600J (AIX), Silicon Graphics IRIS-4D/310GTXB (IRIX System V 3.3.1)
- Version 2.7: VAX 11/780 (VMS 4.3), Cray-2 (UNICOS 5.1.9), Cray Y-MP (UNICOS 5.1.9), Cray X-MP-EA (UNICOS 5.1.9), IBM 3090 600J (AIX), Silicon Graphics IRIS-4D/310GTXB (IRIX System V 3.3.1)
- Version 3.0: VAX 11/780 (VMS 4.3), Cray-2 (UNICOS 5.1.9), Cray X-MP-EA (UNICOS 5.1.9), IBM 3090 600J (AIX), Silicon Graphics IRIS-4D/310GTXB (IRIX System V 3.3.1), Sun SparcStation 4/360 (SunOS 4.1.1)
- Versions 3.0.7, 3.9.1, 3.9.2: Cray-2 (UNICOS 5.1.9), Cray X-MP-EA (UNICOS 5.1.9)
- Version 4.0: Cray-2 (UNICOS 6.1), Cray X-MP-EA (UNICOS 6.1), Cray Y-MP (UNICOS 6.1.5a), IBM RS/6000 model 550 (AIX 3.1.5), Silicon Graphics IRIS-4D/310GTXB (IRIX System 4.0.1), Sun SparcStation 4/360 (SunOS 4.0.3), VAX 6000-510 (VMS 5.4), VAX 11/780 (VMS 4.3)
- Version 4.0.1: Cray-2 (UNICOS 6.1), Cray X-MP-EA (UNICOS 6.1), Cray Y-MP (UNICOS 6.1.5a), IBM RS/6000 model 550 (AIX 3.1.5), Silicon Graphics IRIS-4D/440VGX, IRIS-4D/310GTXB, IRIS-4D/320VGX, IRIS-4D/210VGX, IRIS-4D/35VGX, IRIS/INDIGO (all IRIS computers use IRIX 4.0.1 operating system), Sun SparcStation 4/360 (SunOS 4.0.3), VAX 6000-510 (VMS 5.4), VAX 11/780 (VMS 4.3)
- Version 4.1: Cray-2 (UNICOS 6.1), Cray X-MP-EA (UNICOS 6.1), IBM RS/6000 model 550 (AIX 3.1.5)
- Version 4.2: Cray-2 (UNICOS 6.1), Cray X-MP-EA (UNICOS 6.1), Cray Y-MP (UNICOS 6.1.5a), IBM RS/6000 model 550 (AIX 3.1.5), Silicon Graphics IRIS-4D/310GTXB (IRIX 4.0.1), Sun SparcStation 4/360 (SunOS 4.0.3)
- Version 4.3: IBM RS/6000 model 550 (AIX 3.1.5)

- Version 4.4: Cray-2 (UNICOS 6.1), Cray X-MP-EA (UNICOS 6.1), Cray Y-MP (UNICOS 6.1.5a), IBM RS/6000 model 550 (AIX 3.2), Silicon Graphics IRIS-4D/310GTXB (IRIX 4.0.1), Sun SparcStation 4/360 (SunOS 4.0.3)
- Version 4.5: Cray-2 (UNICOS 6.1), Cray X-MP-EA (UNICOS 6.1), IBM RS/6000 model 550 (AIX 3.2)
- Version 5.0: Cray-2 (UNICOS 6.1), Cray X-MP-EA (UNICOS 6.1), Cray Y-MP (UNICOS 6.1.7), Cray M92 (UNICOS 6.1), IBM RS/6000 model 550 (AIX 3.2), Silicon Graphics IRIS-4D/310GTXB (IRIX 4.0.1), Sun SparcStation IPX (SunOS 4.1.2), DEC 3000/500 AXP (OSF/1)
- Version 5.0.1: Cray-2 (UNICOS 6.1), Cray X-MP-EA (UNICOS 7.C.2), Cray C90 (UNICOS 7.C.2), IBM RS/6000 model 550 (AIX 3.2), DEC 3000/500 AXP (OSF/1), HP 730 and HP 735 (HP-UX 9.01)
- Version 5.1: Cray-2 (UNICOS 7.0.5), Cray X-MP-EA (UNICOS 7.C.2), Cray Y-MP (UNICOS 7.C.3), Cray C90 (UNICOS 7.C.2), IBM RS/6000 model 550 (AIX 3.2), Silicon Graphics IRIS-4D/310GTXB (IRIX 4.0.5), Sun SparcStation IPX (SunOS 4.1.2), HP 730 and HP 735 (HP-UX 9.01)
- Version 6.0: Cray-2 (UNICOS 7.0.5), Cray X-MP-EA (UNICOS 7.C.3), Cray C90 (UNICOS 7.C.3), IBM RS/6000 model 550 (AIX 3.2.3), Silicon Graphics IRIS-4D/35TG (IRIX 4.0.5), Silicon Graphics IRIS-Indigo R4000 (IRIX 5.2), Sun SparcStation IPX (SunOS 4.1.2), Sun SparcStation LX (SunOS 4.1.2)
- Version 6.2: Cray-2 (UNICOS 7.0.5), Cray X-MP-EA (UNICOS 7.C.3), Cray C90 (UNICOS 7.C.3), IBM RS/6000 model 550 (AIX 3.2.5), Silicon Graphics IRIS-4D/310GTXB (IRIX 4.0.5), Silicon Graphics IRIS-Indigo R4000 (IRIX 5.2), Sun SparcStation IPX (SunOS 4.1.2)
- Version 6.5: Cray-2 (UNICOS 7.0.5), Cray X-MP-EA (UNICOS 7.C.3), Cray C90 (UNICOS 7.C.3), IBM RS/6000 model 550 (AIX 3.2.5), DEC Alpha 3000/300 LX (OSF/1 v2.0), Silicon Graphics IRIS-Indigo R4000 (IRIX 5.2), Sun SparcStation IPX (SunOS 4.1.2)
- Version 6.6: Cray-2 (UNICOS 8.0.3), Cray X-MP-EA (UNICOS 8.0.3), Cray C90 (UNICOS 8.0.3), IBM RS/6000 model 550 (AIX 3.2.5), DEC Alpha 3000/300 LX (OSF/1 v2.0), Silicon Graphics IRIS-Indigo R4000 (IRIX 5.2), Silicon Graphics IRIS-Power Challenge L R8000 (IRIX 6.1), Sun SparcStation IPX (SunOS 4.1.2), Power Macintosh 7100/66 (Mac OS 7.5.2)
- Version 6.6.1: Cray-2 (UNICOS 8.0.3), Cray C90 (UNICOS 8.0.3), IBM RS/6000 model 550, 590 (AIX 4.1), Silicon Graphics IRIS-Indigo R4000 (IRIX 5.3), Silicon Graphics IRIS-Power Challenge L R8000 (IRIX 6.1), Sun SparcStation IPX (SunOS 5.4)
- Version 7.0: Cray C90 (UNICOS 8.0.3), IBM RS/6000 models 550 and 590 (AIX 4.1), Silicon Graphics IRIS-Power Challenge L R8000 (IRIX 6.1), Sun SparcStation IPX (SunOS 5.4), DEC Alpha 3000/300 (OSF/1)

- Version 7.1: Cray C90 (UNICOS 8.0.3), IBM RS/6000 models 550 and 590 (AIX 4.1), Silicon Graphics IRIS-Power Challenge L R8000 (IRIX 6.1)
- Version 7.1.1: Cray C90 (UNICOS 8.0.3),
- Version 7.2: Cray C90 (UNICOS 8.0.3), IBM RS/6000 models 550 and 590 (AIX 4.1), and Silicon Graphics IRIS-Power Challenge L R8000 (IRIX 6.1)
- Version 7.3–7.3.1: Cray C90 (UNICOS 8.0.3), IBM RS/6000 models 550 and 590 (AIX 4.1), and Silicon Graphics Indigo-2 R10000 (IRIX 6.2)
- Versions 7.4–7.4.3: Cray C90 (UNICOS 8.0.3), IBM RS/6000 models 550 and 590 (AIX 4.1), Silicon Graphics Indigo-2 R10000 (IRIX 6.2), and Dell XPS Pro200n (Linux, RedHat 4.2)
- Version 7.8–7.9.2: Cray C90 (UNICOS 8.0.3), IBM RS/6000 models 550 and 590 (AIX 4.1), Silicon Graphics Indigo-2 R10000 (IRIX 6.2), and Micron Millennia Pro (RedHat Linux 4.2, kernel 2.0.30).
- Version 8.0: IBM RS/6000 model 590 (AIX 4.3), Silicon Graphics Indigo-2 R10000 (IRIX 6.2), and Dual Intel Pentium II-333 (RedHat Linux 4.3, kernel 2.0.33).
- Version 8.1–8.5.1: IBM SP with PowerPC 604e processors (AIX 4.3.1) and Silicon Graphics Power Challenge L with R10000 (IRIX 6.5).
- Version 8.6–8.7.3: IBM SP with Power3 processors (AIX 4.3.4.0) and SGI Origin 2000 R12000 (IRIX 6.5.12f).
- Version 8.9–9.1: IBM SP with Power3 processors (AIX 4.3.4.0), SGI Origin 2000 R12000 (IRIX 6.5.12f), Sun Blade 2000 (Solaris 8), and Compaq XP1000 AlphaStation (Tru64 4.0f).
- Version 8.9–9.1: IBM SP with Power3 processors (AIX 4.3.4.0), SGI Origin 2000 R12000 (IRIX 6.5.12f), Sun Blade 2000 (Solaris 8), and Compaq XP1000 AlphaStation (Tru64 4.0f).
- Version 9.4-9.4.1: See Table in Section 4.C

4.C. Compiler versions

- Versions 7.4–7.4.3: Cray C90 (cft77 Version 6.0.4.19), IBM RS/6000 models 550 and 590 (XL Fortran Version 3.2), Silicon Graphics Indigo-2 R10000 (Fortran 77 GCS Version 6.2), and Dell XPS Pro200n (f2c-gcc, kernel 2.0.30).
- Version 7.8–7.9.2: Cray C90 (cft77 Version 6.0.4.19), IBM RS/6000 models 550 and 590 (XL Fortran Version 3.2), Silicon Graphics Indigo-2 R10000 (Fortran 77 GCS Version 6.2), and Micron Millennia Pro (f2c-gcc, kernel 2.0.30).

Version 8.0: IBM RS/6000 model 590 (XL Fortran Version 3.2), Silicon Graphics Indigo-2 R10000 (Fortran 77 GCS Version 6.2), and Dual Intel Pentium II-333 (f2c-gcc, kernel 2.0.33).

Version 8.1–8.5.1: IBM SP with PowerPC 604e processors (XL Fortran Version 6.0.1) and Silicon Graphics Power Challenge L with R10000 (MIPSpro 7.2.1.3m).

Version 8.6–8.7.3: IBM SP with Power3 processors (XL Fortran 6.1.0.3) and SGI Origin 2000 R12000 (MIPSpro 7.3.1.2m).

Version 8.9–9.1: IBM SP with Power3 processors (XL Fortran 6.1.0.3), SGI Origin 2000 R12000 (MIPSpro 7.3.1.2m), Sun Blade 2000 (Sun WorkShop[tm] 6 update 2 Fortran 77 5.3), and Compaq XP1000 AlphaStation (Compaq Fortran V5.4-1283-46ABA).

Version 9.4-9.4.1:

Platform	Operating System	Compilers
IBM Regatta, Power4 procs	AIX 5.2	XL Fortran version 8.1.1
		g77 from gcc version 3.4.1
IBM Netfinity, Dual Pentium III procs	Redhat Enterprise Linux 3	Portland group fortran compiler version 5.2
		g77 from gcc version 3.3.3
SGI Altix, Itanium 2 procs	SuSE Linux	Intel ifort version 9.0
		g77 from gcc version 3.2.3

5. PROGRAM AND MANUAL DESCRIPTION

5.A. Code description, portability, and special files

The POLYRATE program is written in ANSI-standard FORTRAN 77 with the INCLUDE and DO WHILE extensions. Another exception to the ANSI standard in the POLYRATE program is that all the subprogram names are in lowercase. If the user's FORTRAN compiler does not allow the INCLUDE extension, the only way this program can be compiled successfully on that system is by replacing the INCLUDE statements with the data in the INCLUDE files in each subprogram.

The subprogram DATTIM, which obtains the date and the wall clock time, is the only machine-specific subroutine in the code; however, versions are supplied for a variety of machines.

5.A.1. DATTIM subprogram

The subprogram DATTIM obtains the current date and wall clock time for stamping the POLYRATE output files by calling machine-specific intrinsic FORTRAN functions which return the date and the wall clock time. We supply different versions of this subprogram for a variety of machines. They are dattim.altix for SGI Altix, dattim.compaq for Compaq AlphaStation, dattim.cray for the Cray-2, Cray X-MP, Cray Y-MP, Cray M92, and Cray C90, dattim.iris for the IRIS-4D, IRIS-Indigo, SGI Power Challenge, and SGI Origin, dattim.sun for the SUN SparcStation and UltraSparc workstation, dattim.ibm for the IBM RS/6000 series, dattim.dec for the DEC 3000, dattim.hp for the HP 730 and 735, and dattim.linux for systems running the Linux operating system. In addition, we supply a version of this subprogram, dattim.f90, which uses the standard FORTRAN 90 routine for date and time, DATE_AND_TIME.

One exception to the lack of machine specificity is that some Unix operating systems have no such FORTRAN intrinsic functions. To overcome this problem, a C subprogram, in particular dateclock.c, has been written to obtain the date and the wall clock time using Unix utilities.

If none of the `dattim.machine` files in the distribution package are compatible with the operating system on which POLYRATE will be installed, there are three options:

1. If the operating system is Unix, the user can try using the `dateclock.c` subroutine which obtains the date and time using Unix utilities.
2. The user can write a `dattim` subprogram which makes the appropriate intrinsic function call(s).

5.A.2. INCLUDE files

The POLYRATE program requires three INCLUDE files for compilation: `param.inc`, `percon.inc`, and `common.inc`. When running POLYRATE under an operating system that is case-sensitive, these file names must be in lower case. In any subprogram that requires more than one INCLUDE file, the INCLUDE files must be listed in the order that they are mentioned above since the variables in one depend on those in another. The INCLUDE file `param.inc` contains all the parameters that can be modified by the user. The INCLUDE file `percon.inc` contains the parameters which are permanent constants for a particular version of POLYRATE. The INCLUDE file `common.inc` contains all the common blocks used by POLYRATE to pass information between the various subprograms.

Note that there are four sample files that can be used as `param.inc` file. They are `param1.inc`, `param2.inc`, `param3.inc`, and `param4.inc`. The `param.inc` file in the distribution package is identical to the `param4.inc` file. The user may edit this file or overwrite it with one of the examples. See Subsection 13.A.2 for details on using these include files to compile the source code.

Below is a description of the parameters in the `param.inc` file.

parameter	description
FU#	<p>parameters which link I/O units to poly.fu# files, where # is 1, 2, 3, 5, 6, 8, 14, 15, 18, 19, 20, 21, 22, 24, 25, 26, 27, 28, 30, 31, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 60, 65, 71, 72, 73, 74, 75, 76, 77, or 78.</p> <p>FU# can be a one or two digit integer which is any allowed I/O unit on the computer.</p> <p>For example, if FU5 is set equal to 5, I/O unit 5 is used for the POLYRATE main input.</p> <p>A full description of all the input and output files and the corresponding FORTRAN unit numbers is given in Section 17.</p>
INMM	<p>maximum value for the ratio between the number of gradient save points and the number of Hessian save points (used only with fu31).</p>
MAXPS	<p>the maximum number of product states (in the exoergic direction of reaction) into which an LCG3 quasidiabatic mode p can be excited. (The minimum is 1, and this value should be set to equal to one to minimize storage space unless the QRST option is invoked for tunneling in a unimolecular reaction.)</p>
MXWKB	<p>the maximum number of excited states allowed for the reaction-coordinate mode in a quantized-reactant-state tunneling calculation.</p>
NATOMS	<p>the maximum number of atoms.</p>
NSDIM	<p>the maximum number of grid points ($G + 3$) used for MI interpolation.</p>
NSDM	<p>the maximum number of points along the minimum energy path at which Hessian-dependent reaction-path information can be saved.</p>
NSDML	<p>the maximum number of points along the minimum energy path at which the information relevant to the large-curvature tunneling (LCT) calculation can be saved. (Note that NSDML must be set equal to or greater than NSDM if LCT calculations are to be carried out. Otherwise, NSDML can be any integer greater than 0.)</p>
NMSPEC	<p>the maximum number of extra points at which a generalized normal mode analysis may be performed; see SSPECIAL in the unit fu5 input. (The default is 20. The minimum is 1.)</p>
NPOTPT	<p>the maximum number of pieces of information that can be passed back to POLYRATE from an analytic potential energy surface through the common block SURFCM, i.e.,</p> <pre>COMMON /SURFCM/ POTINF (NPOTPT)</pre> <p>NPOTPT must be a number between 1 and 99; see Subsection 8.C.1</p>

The parameters used in the four sample INCLUDE files are listed below.

	param1.inc	param2.inc	param3.inc	param4.inc
NATOMS	6	6	9	11
NSDIM	701	701	901	901
NSDM	701	701	901	901
NSDML	1	701	1	901
MAXPS	1	5	1	5

In all of these files, INMM is set to 10, MAXWKB is set to 50, NMSPEC is set to 20, NPOTPT is set to 1, and FU# is set equal to #, e.g., FU5 is set equal to 5.

5.B. Description of the manual

Section 6 provides an overview of the theoretical background for the dynamical calculations. Section 7 gives information related to the program structure. Sections 8 and 9 describe the calculations using a potential energy surface and an electronic structure input file, respectively. Section 10 provides selected details of the calculations, especially for procedures that are not fully described in publications. Section 11 describes the complete POLYRATE input and selected aspects of the output. Section 12 describes files that give extra details of LCT calculations. Section 13 contains instructions on how to install, test, and use the POLYRATE program. Section 14 gives sample runs (the POLYRATE test suite) that may be useful in understanding the input or testing the program. Section 15 lists the files in the program distribution, and it has a brief explanation of each subprogram. Section 16 contains instructions on using the Unix scripts that are provided. Section 17 describes file usage. Section 18 contains tables of compiling and linking commands and memory requirements on various computers. The memory requirements were obtained using the "size" command on the various Unix computers. Section 19 contains test run timing information. Section 20 provides other references used as footnotes in the text, and Section 21 is the POLYRATE bibliography.

Much of the mathematical formulation underlying the dynamical calculations is presented in a book chapter entitled "Variational Transition State Theory with Multidimensional Tunneling," which is Ref. 59 in the bibliography (Section 21), and should be consulted along with this manual.. "Generalized Transition State Theory" (GTST), which is Ref. 5, is also a useful chapter, although it is older and the material in Ref. 59 supersedes it. If

this book is unavailable in your library, you may obtain a complimentary copy of this chapter from the senior author (DGT). A complete compilation of all known errors in this reference, some errors in other papers, and in previous manuals are given in Section 22.A. In addition, an appendix (Section 23) contains the POLYRATE revision history.

5.B.1. Referencing conventions

Reference numbers in this manual refer to references in Sections 20 and 21. In all cases the reference number is accompanied by the section number to avoid confusion.

5.B.2. Book chapters

Reference 5 in Section 21 is sometimes referred to simply as "the GTST book chapter". Reference 59 in Section 21 is sometimes referred to as "the VTST/MT book chapter." These two book chapters explain the methods used in POLYRATE.

6. THEORETICAL BACKGROUND FOR DYNAMICAL CALCULATIONS

This section provides an overview of the dynamical theory underlying the calculation procedures. Further details of the methods for gas-phase bimolecular and unimolecular reactions may be found in the "Generalized Transition State Theory" (GTST) book chapter (Ref. 5, Sect. 21). Details of new methods developed after that chapter was written are described in Section 10 of this manual, if not fully published, or in the additional references in Section 21.

6.A. Basic theory

As described in the Introduction, the program can be used to calculate rate constants for reactions of one or two reactant molecules to form one or two product molecules, using a variety of generalized transition state theory methods. In order to simplify the discussions in the remainder of this introductory overview section, we consider only the calculation of forward and reverse thermal rates using both conventional transition state theory (TST) and canonical variational theory (CVT). Improved canonical variational theory (ICVT) and microcanonical variational theory (μ VT) are described in detail in the GTST book chapter. The applications of VTST methods to gas-phase unimolecular or bimolecular reactions or gas-solid-interface reactions or the calculation of adiabatic and diabatic state-selected rate constants are described in the references included in Section 20.

The conventional transition-state-theory canonical rate constant involves a symmetry number [Ref. 1, Sect. 20] times kT/h times a ratio of a partition function evaluated at the saddle point to one evaluated at the reactants, and, optionally, times a temperature-dependent transmission coefficient, which in this program, takes into account tunneling and non-classical reflection effects. The partition functions are evaluated as products of electronic, vibrational, and rotational partition functions, and for the reactants of a bimolecular reaction, a relative translational partition function per unit volume. In POLYRATE, vibrational partition functions are defined such that they tend to a Boltzmann factor of the zero point energy at low temperature (not to unity as is common in statistical mechanics books), and the corresponding local zero of energy is used consistently for the rest of the calculations.

To calculate rate constants by the CVT method various items of information are needed along the reaction path. In POLYRATE, the reaction path is always taken as the minimum energy path (MEP) through mass-scaled (i.e. isoinertial) coordinates [Refs. 2-5, Sect. 20]. Following Fukui [Ref. 5, Sect. 20], one may call this the "intrinsic" reaction path. (In POLYRATE, we always call it the MEP.) The distance along the MEP is called s , or the reaction coordinate. For reactions with a saddle point, the MEP is specifically defined to be the path of steepest descent through mass-scaled Cartesian coordinates from the saddle point ($s = 0$) to the reactants (at which $s = -\infty$ for a bimolecular reaction, or s equals a finite negative number for a unimolecular reaction) and the path from this saddle point to the products (at which $s = +\infty$ for a bimolecular reaction, or s equals a finite positive number for a unimolecular product). We define a sequence of generalized transition states corresponding to the reaction coordinate to be hypersurfaces which are orthogonal to the MEP [Ref. 6, Sect. 20]. All quantities calculated for a generalized transition state that intersects the MEP at a particular value s of the reaction coordinate are labeled by that value of s (as well as the temperature T when they depend on temperature). The generalized TST canonical forward rate constant $k_f^{\text{GT}}(T, s)$ is then evaluated like the conventional one but with saddle point properties replaced by generalized transition-state (GT) ones. Thus, at each generalized transition state along the reaction path the following information is (or may be, depending on the options chosen) needed:

$V_{\text{MEP}}(s)$	—	potential along the minimum energy path,
$ \mathbf{I}(s) $	—	determinant of the moment of inertia tensor,
$\mathbf{X}(s)$	—	geometry of the generalized transition state,
$\nabla V_{\text{MEP}}(s)$	—	the gradient of the generalized transition state,
$\mathbf{L}_m(s)$	—	eigenvector of the generalized normal mode m ,
$\omega_m(s)$	—	generalized normal mode frequencies.

In addition, depending on the anharmonicity option and the method chosen to include tunneling, one may also require some or all of the following additional functions:

$k_m^{(3)}(s)$	—	principal cubic force constants for the generalized normal modes,
$k_m^{(4)}(s)$	—	principal quartic force constants for the generalized normal modes,
$B_{mF}(s)$	—	reaction-path curvature coupling [Ref. 7, Sect. 20] between the reaction coordinate (mode F) and generalized normal

mode m . Sometimes we refer to the set of $B_{mF}(s)$ with $m=1, 2, \dots, F-1$, as the curvature vector \mathbf{B}_F .

Note that “principal” force constants are diagonal in the normal mode index; that is why $k_3(s)$ and $k_4(s)$ are short one-dimensional arrays. The functions listed above and the reaction path itself can be obtained through the hooks, which can in turn be interfaced to an analytic potential energy surface or an electronic structure package, or alternatively this data (or information which can be used to calculate this data) can be obtained from electronic structure calculations carried out separately from POLYRATE. In the latter case the MEP information is input only on a relatively sparse grid of s values, and the program interpolates this information to arbitrary s values. This option is called the electronic structure input file option (POTENTIAL = *unit30*, *unit31*, or *unit40*).

The isoinertial coordinate system used by POLYRATE is defined by eq. (2) of Ref. 1 of Sect. 21 or, equivalently, by eq. (2) of Ref. 5 of Sect. 21. These coordinate systems have the same reduced mass, μ , for all degrees of freedom. Physical observables and the MEP are independent of μ , but numerical values of s are not. Thus it is essential to specify μ if s values or plots of barriers vs. s are considered. If μ is set equal to 1 amu, then the numerical value of s in mass-scaled coordinates (which have units of length) is the same as the numerical value of the distance along a mass-weighted reaction coordinate (which has units of mass^{1/2} length) if the mass weighting is amu^{1/2}. Recommendation to users who do not understand μ ; always set it equal to 1 amu.

The reverse rate constant is obtained from the forward one by detailed balance. The canonical variational theory [Ref. 6, Sect. 20] forward rate constant is obtained by minimizing the generalized transition state theory rate constant with respect to s :

$$k_f^{CVT}(T) = \min_s k_f^{GT}(T, s) = k_f^{GT} \left[T, s_*^{CVT}(T) \right] \quad (1)$$

Generalized-transition-state-theory rate constants can also be written in terms of the generalized transition-state (GT), standard-state (0), forward (f) free energy of activation, $\Delta G_f^{GT,0}(T, s)$. The location of the variational transition state is then determined by maximizing the free energy of activation with respect to s .

Section 6.C gives a brief explanation of the evaluation of the partition functions and Section 6.D summarizes methods used for transmission coefficients.

6.B. Vibrations, Rotations, and Translations

This section gives a brief explanation of how POLYRATE uses the SPECIES keyword (see Section 11.A.6) to determine the number of rotations and vibrations used in calculating the partition functions and the number of degrees of freedom which are projected out by the projection operator (when the PROJECT option is on, see Section 11.A.6) for the given reactant, product, transition state, or generalized transition state. The table below gives the valid values of the SPECIES keyword and a description of what the values mean, along with the number of vibrations and rotations in the partition function for each type of species, and the number of modes which are projected out to remove translations, rotations, and the reaction coordinate.

Description	SPECIES keyword value	Partition Function		Projection Operator			
		Vibrations	Rotations	Translations	Rotations	Reaction Coordinate	Mandatory?
atomic reactant or product	<i>atomic</i>	0	0	N.A.
linear reactant or product	<i>linrp</i>	$3N - 5$	2	3	2	0	optional
nonlinear reactant or product	<i>nonlinrp</i>	$3N - 6$	3	3	3	0	optional
solid-state reactant or product	<i>ssrp</i>	$3N$	0	0	0	0	prohibited
linear transition state	<i>lints</i>	$3N - 6$	2	3	2	0	optional
nonlinear transition state	<i>nonlints</i>	$3N - 7$	3	3	3	0	optional
solid-state transition state	<i>ssts</i>	$3N - 1$	0	0	0	0	prohibited
linear GTS	N.A.	$3N - 6$	2	3	2	1	mandatory
nonlinear GTS	N.A.	$3N - 7$	3	3	3	1	mandatory
solid-state GTS	N.A.	$3N - 1$	0	0	0	1	mandatory

N.A., Not Applicable; N , number of atoms

For an atomic reactant or product the value *atomic* should be specified for SPECIES. Since the species is a single atom there are no vibrations or rotations, nor is there any Hessian to consider projecting. Therefore the PROJECT keyword (see Section 11.A.6) can not be used with an atomic species.

Gas-phase reactants and products that are not atomic and transition states can be either linear or nonlinear, specified by *linrp*, *nonlinrp*, *lints*, and *nonlints* respectively. After removing the 2 rotations and three translations for a linear molecule there are $3N - 5$ vibrations for the linear species, and after removing the 3 rotations and 3 translations for a nonlinear molecule there are $3N - 6$ vibrations for a nonlinear species. A gas-phase transition state has one additional mode removed due to the reaction coordinate and therefore has $3N - 6$ real vibrations for a linear molecule and $3N - 7$ real vibrations for a nonlinear molecule. The number of vibrations for a generalized transition state are calculated in the same manner as a transition state. Projecting the Hessian at a transition state is not required, but sometimes it leads to a more stable calculation of vibrational frequencies and eigenvectors, so it is a recommended option to project out translations and rotations at a reactant, product, or transition state. For generalized transition states (points on the reaction path where the gradient does not vanish), projecting out the reaction coordinate is mandatory if the resulting frequencies are to represent motions orthogonal to the reaction coordinate. In such cases it is also mandatory to project out translations and rotations, if they exist (they always exist for gas-phase cases, but not for solid-state cases).

Solid-state species are different than gas-phase species as concerns the number of vibrations. For solid state species there are no rotations or translations which are removed because a rotation or a translation of a species relative to a surface changes the potential energy of the whole system. Therefore there are $3N$ vibrations for a reactant or product species and $3N - 1$ for a transition state or generalized transition state species. The reaction coordinate is removed for a solid-state species in the same manner as it is for a gas-phase species.

6.C. Partition functions

The electronic partition function for a given species is computed quantum mechanically by summing over the Boltzmann factors of the electronic energy levels of the species

times their degeneracies. In practice, since the energy spacing between electronic multiplets is generally quite large, we usually include in the summation only those levels comprising the energetically lowest multiplet, e.g., the $J = 0, 1,$ and 2 levels of $O(^3P)$.

The rotational partition function is approximated by the classical limit [Refs. 8 and 9, Sect. 20]. For a linear reactant, product, transition state, or generalized transition state, the partition function is evaluated from the unique moment of inertia. For a nonlinear species, the product of the three principal moments of inertia is required and is conveniently obtained from the determinant of the moment of inertia tensor. All symmetry numbers are left out of all rotational partition functions; the symmetry numbers need only to be included in the overall symmetry number mentioned above [Ref. 1, Sect. 20].

For bimolecular reactions, the relative translational partition function per unit volume is calculated classically.

For vibrational partition functions, a separable approximation is assumed, whereby the overall vibrational partition function is a product of independent normal modes of a reactant, product, or saddle point or the generalized normal modes of a generalized transition state. Vibrational modes are treated classically or quantized, either quantum mechanically or semiclassically. Anharmonic methods are limited to intramode anharmonicity in normal modes expressed as linear combinations of rectilinear coordinates and do not include mode-mode coupling as a supported option at this time. Classical vibrational partition functions are evaluated using the classical expression for the harmonic oscillator. Quantum mechanical vibrational partition functions are evaluated by the harmonic approximation, by summing Boltzmann factors using quantum or semiclassical energy levels.

Frequencies can be scaled for partition function calculations in CVT calculations when the harmonic approximation is employed. The scaling is not used in the calculation of transmission coefficients (either by CAG or tunneling).

6.D. Tunneling methods

Different semiclassical methods are used to evaluate transmission coefficients depending on the degree of coupling between the MEP and other degrees of freedom through the MEP curvature. There are three general types of transmission coefficients: zero-curvature tunneling (ZCT), small-curvature tunneling (SCT) and large-curvature tunneling (LCT). The methods included in this version for each type of transmission coefficient for thermal reactions are as follows:

- ZCT: minimum-energy-path semiclassical adiabatic ground-state (MEPSAG) method [Refs. 4 and 10, Sect. 20 and Ref. 5, Sect. 21].
- SCT: centrifugal-dominant small-curvature semiclassical adiabatic ground-state (CD-SCSAG) method [Refs. 29 and 31, Sect. 21].
- LCT: large-curvature ground-state version-3 (LCG3) method [Refs. 5, 21, 29, 30, and 34, Sect. 21] and large-curvature ground-state version-4 (LCG4) method [Ref. 54, Sect. 21].

The current theory of these tunneling methods appears in “Variational Transition State Theory with Multidimensional Tunneling,” which is Ref. 59. The final version (CD-SCSAG, which replaced the original SCSAG) of the SCT method is described in Refs. 29 and 31 in Sect. 21, and the final versions of the LCT method are described in Refs. 29, 30, 34 (LCG3 including tunneling into excited states), and Ref. 54 (LCG4 including tunneling into excited states) in Sect. 21. The final versions of these methods were used in all references of Sect. 21 that have reference number 30 or higher. A critical element of the SCT calculation is that the reduced mass, μ (See Section 6.A), is replaced by an effective mass $\mu_{eff}(s)$ in calculating the tunneling probabilities. See Section 6.I of this manual for further discussion.

Notice that the G in MEPSAG, CD-SCSAG, LCG3, and LCG4 refers to the fact that the transmission coefficient for thermal reactions is always calculated for the ground (G) state of the reactants in the exoergic direction of a reaction, as discussed elsewhere [Ref. 13, Sect. 20 and Refs. 29, 30, and 34 of Sect. 21]. The transmission coefficient for the endoergic direction of a reaction is then obtained by detailed balance, which requires that the transmission coefficients for thermal reactions be the same in both directions. The LCG3 and LCG4 methods can include tunneling into excited states of the product in the exoergic direction, and the algorithm in POLYRATE is general enough to treat even cases that involve a well on the product side in the exoergic direction. (Wells on the reactant

side do not require special considerations in this regard.) The ground-state approximation for the transmission coefficient is equivalent to assuming that the energy-dependent transmission probability curve for each successive quantized level of the transition state has the same functional dependence on energy as that for the ground state but is shifted up in energy by the excitation energy evaluated at the variational transition state [Ref. 47, Sect. 20].

When reaction-path curvature is large, it may be *necessary* to include excited states of the product of the exothermic direction in order to obtain accurate results, and the user of POLYRATE is responsible for checking this by checking convergence with respect to the argument of the EXCITED keyword in the TUNNEL section.

VTST methods are also applicable to state-selected reactants if the quantum number of the selected mode remains adiabatic, to a good approximation, all the way to the transition state. Excited-state tunneling probabilities may be defined using similar methods to those mentioned above, and they are called MEPSA, CD-SCSA, and LC3 or LC4, i.e., we don't include the G when the method is applied to excited states of reactants. These abbreviations and the ZCT, SCT, and LCT abbreviations are also useful when we wish to refer to the semiclassical method without reference to the initial state. The present version of POLYRATE allows the MEPSA and CD-SCSA options for excited-state reactions but does not allow LC3 or LC4, which is not coded in such generality.

Regions of large reaction-path curvature can promote vibrational nonadiabaticity. The partial-reaction-path (PRP) adiabaticity approximation [Ref. 13A, Sect 21] assumes adiabaticity in the excited-state vibration up to the first point along the reaction coordinate where an appreciable local maximum in the reaction-path curvature occurs, which we denote s_+ . The vibration is assumed to relax nonadiabatically into the ground vibrational state after s_+ . The adiabatic potential for tunneling is the excited-state one up to s_+ and the ground-state one for $s > s_+$. Excited-state PRP calculations can be implemented using the STATE and STATEOPT keywords in the RATE section. An example of an excited-state PRP calculation is provided in test run ohcltr4.

The multidimensional tunneling methods used in POLYRATE have been tested extensively against accurate quantal results [Ref. 53, Sect 20].

6.E. Reactant and product properties

The first step in the calculation is to determine the properties of the reactants and products. We will specifically discuss reactant A (the first reactant, which may be an atom or molecule) as an example in this subsection; the methods for the other reactant, in the case of a bimolecular reaction, or for products are analogous.

The calculation begins by determining the equilibrium geometry and energy for reactant A on the potential energy surface. The reactant geometry is optimized by using either the Newton-Raphson (NR) method with Brent line minimization [Ref. 14, Sect. 20], the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [Ref. 15, Sect. 20], or the Eigenvector Following (EF) method [Refs. 54-58, Sect. 20] to search over a restricted set of atomic Cartesian coordinates. These options are not available for the program packages MORATE and AMSOLRATE, in which the geometry optimization is only carried out in the electronic structure program (which is specified using option HOOK). To do this, the user must fix (somewhat arbitrarily) 6 of the Cartesian coordinates (5 for a linear species) and search over the rest. For example, the user might fix atom 1 at the origin, atom 2 on the x-axis, and the atom 3 in the x-y plane. (In the case of solid-state or gas-solid-interface reactions, there are no translations or rotations because the moving atoms are adsorbed on or embedded in a stationary lattice; thus for such cases, the full set of atomic Cartesians of the movable atoms is used). Starting from an initial guess, an improved estimate for the position of the minimum is obtained by taking a step ΔR_0 , where ΔR_0 is obtained by solving a simultaneous set of equations (given in the GTST book chapter) involving the matrix of second derivatives of the potential energy and the gradient. Once ΔR_0 is obtained by direct calculation of the inverse Hessian matrix (NR method) or estimated using information from the previous geometry optimization steps (BFGS method), a new guess for the position of the minimum is obtained using either the NR step or a variable step size which is determined by the Brent line minimization method [Ref. 16, Sect. 20] (BFGS method). For the EF method, the step is determined by the hessian. If the hessian has the wrong number of negative eigenvalue, then the partitioned rational function (P-RFO) approximation is used. If the hessian has one negative eigenvalue, then Newton-Raphson step is used. The detailed EF algorithm is given in [Ref. 56, Sect. 20].

For all methods, convergence is reached when the maximum component of the gradient in the space of optimized coordinates is less than a prescribed amount GCOMP. In order

to increase the stability of the optimization, the maximum component of the step, ΔR_0 , is not allowed to be larger than the prescribed amount SCALE for the NR and BFGS methods. However, the SCALE keyword has no effect on the EF method. The accurate Hessian is recalculated every HREC optimization steps. In the case of the NR method, if HREC is not equal to 1, the Hessian is kept frozen until recalculated. The BFGS method is the default. The EF method has a few extra options, namely, RMAX, RMIN, OMIN, DDMAX, and DDMAXTS, to control the size of the trust region and tune up the optimization procedure.

Next, the origin is moved to the center of mass of reactant A. The moment of inertia or the product of the principal moments of inertia is then determined. Finally, a normal mode analysis is performed on reactant A. Since, in mass-scaled Cartesians, motion in any direction is associated with the same reduced mass (that is why we call these isoinertial coordinates), this is easily accomplished by diagonalizing the force constant matrix. The force constant matrix is calculated by numerical first derivatives of the analytic or input first derivatives of the potential. In the case of gas-phase reactions, we can optionally project out the contamination of the translational and rotational motions in the force constant matrix at the stationary point before diagonalizing it to yield 6 zero eigenvalues (5 for a linear molecule) corresponding to overall translation and rotation. The remaining eigenvalues are all positive, corresponding to bound normal mode frequencies, and their associated (normalized) eigenvectors L_m give the normal mode directions. After the frequencies are calculated, the frequencies and corresponding eigenvectors are placed in “canonical order.” Canonical order is defined as follows:

First define G:

$$\text{Stationary point: } G = R + T = \begin{array}{l} 6 \text{ for nonlinear molecule,} \\ 5 \text{ for linear molecule,} \\ 0 \text{ for gas-solid interface.} \end{array}$$

$$\text{Reaction path: } G = R + T + 1$$

Then the frequencies are ordered as follows:

1. The G frequencies with the smallest absolute values go last, ordered in order of decreasing magnitude.
2. Of the remaining frequencies, first put the real ones, in decreasing order, then put the imaginary ones, in order of increasing magnitude.

By computing or reading the third and fourth derivatives of the potential energy along these normal mode directions, anharmonicity may be included separately in each of the

modes, as discussed elsewhere [Ref. 1, Sect. 21]. Depending on the method chosen for treating the anharmonicity, the zero point energy is then computed for reactant A by summing up the zero point energies in each of the normal modes.

The calculations in this section are then repeated for the other reactant (if it exists) and for product(s). From these results, the adiabatic energy of reactants, the classical endoergicity for the reaction, ΔE , and the product adiabatic energy are obtained.

Notice that the normal mode analysis can only be carried in rectilinear coordinates for reactants and products, whereas the saddle point and the nonstationary points can be treated either in rectilinear or in curvilinear coordinates.

6.F. Saddle point properties

The calculation of the saddle point geometry and properties is next. The saddle point position is determined by means of the Newton-Raphson method, the BFGS method, or the EF method as described in Subsection 6.E. The default for transition state searches is the Newton-Raphson method. This optimization provides the saddle point's classical barrier height $V_{\text{MEP}} (s = 0)$. Next, the origin is moved to the center of mass of the full N -atom system. The moment of inertia is obtained as discussed for reactants. Next the force constant matrix is calculated in mass-scaled Cartesians, and optionally the contributions of the overall translational and rotational motions are projected out. The normal mode analysis can be carried out in either the rectilinear or curvilinear coordinates for the saddle point. In the case of using rectilinear coordinates, a diagonalization of the $3N \times 3N$ matrix of force constants in mass-scaled Cartesian coordinates is performed. Upon diagonalization, we again obtain 6 zero eigenvalues (5 for a linear molecule, 0 for solid state), $3N-7$ ($3N-6$ or $3N-1$) positive eigenvalues corresponding to bound normal mode frequencies, and one negative eigenvalue corresponding to the imaginary frequency whose eigenvector provides the reaction coordinate direction at the saddle point. After the frequencies are calculated, the frequencies and corresponding eigenvectors are placed in canonical order (which is defined in Section 6.E). The imaginary frequency is used at a later stage to compute the Wigner tunneling correction (see Subsection 6.J). By computing or reading the third and fourth derivatives along the normal mode directions as discussed above, anharmonicity can be included separately in each of the $3N-7$ ($3N-6$ for a linear molecule, or $3N-1$ for solid state) bound normal modes (see Subsection 10.B of this manual), and the zero point

energy (ZPE) for the transition state is computed from the sum of the zero point energies of each of the modes. The ground-state adiabatic energy at the saddle point is then calculated as:

$$V_a^G(s = 0) = V_a(n = 0, s = 0) = V_{\text{MEP}}(s = 0) + ZPE(s = 0) \quad (2)$$

The normal mode analysis can also be carried using redundant (or nonredundant) curvilinear coordinates. In this case, Wilson's *FG* matrix method is used to obtain the projected force constant matrix in internal coordinates. See Section 10.L. for further discussion of curvilinear internal coordinates.

6.G. Reaction path

In this section, we discuss how the reaction path (or, more precisely, a set of points along an approximation to the reaction path) and various properties along it are determined. In the present version of POLYRATE, the reaction path is always taken as the minimum-energy path (i.e. the path of steepest descents) in mass-scaled coordinates; this is called the MEP. Note that the path of steepest descent is the same in any iso-inertial coordinate system. [Ref. 49, Sect. 20] There are four cases, and the current version of POLYRATE allows three of them:

In START	In PATH	
STATUS = 0	SADDLE	Reaction has a saddle point. Initial guess of its geometry is provided in the START section of the fu5 input file. The geometry of the saddle point is optimized. The first step is followed along the imaginary frequency normal mode (if FIRSTSTEP is <i>nmode</i>) or with the cubic starting algorithm based on this mode (if FIRSTSTEP is <i>cubic</i>).
STATUS = 2	SADDLE	This is the same as above except it assumes that the geometry in the START section is already fully optimized, and it skips the optimization step.
STATUS = 0	NOSADDLE	not supported
STATUS = 2	NOSADDLE	Reaction does not have a saddle point. It is a strictly downhill association reaction. The reaction path is started by following the negative gradient at the

geometry provided in the START section.

Most of the test runs in the current version of POLYRATE correspond to STATUS = 0/SADDLE. An example of STATUS = 2/SADDLE is test run oh3tr1. All fu29, fu30, fu31, and fu40 test runs use STATUS = 2/SADDLE. An example of STATUS = 2/NOSADDLE is ho2tr2.

6.G.1. SADDLE option

Taking the saddle point as the first point on the reaction path, an initial step of prescribed length is taken according to one of two options. Either it is taken based on a cubic expansion of the potential energy surface [Ref. 17, Sect. 20] using the normalized eigenvector corresponding to the imaginary frequency and the curvature vector, both of which are calculated at the saddle point, or it is taken in the direction given solely by the normalized eigenvector corresponding to the imaginary-frequency normal mode at the saddle point, as was done in the original version of the program [Ref. 1, Sect. 21] described in the GTST book chapter.

In the original straight VTST calculations [Ref. 18, Sect. 20; Ref. 1, Sect. 21] the MEP was calculated by the Euler steepest-descents (ESD) method in which repeated steps of length Δs are then taken in the direction of the negative of the normalized gradient. The whole process is then repeated for the other direction of leaving the saddle point, thus providing the entire reaction path. The distance between successive points along the reaction path (δs) is called the reaction path stride or the gradient step size, and the set of such points is called the fine grid or the gradient grid. Thus, at each point on the fine grid both the potential and gradient are required.

To reduce the numerical error in the above approach due to wandering about the true steepest-descent path, the program allows for a stabilization algorithm to be employed. This algorithm [Ref. 18, Sect. 21] may be considered to be a modified version of the prescriptions of Ishida et al. [Ref. 19, Sect. 20] and Schmidt et al. [Ref. 20, Sect. 20]. It is called ES1 (Euler stabilization method, version 1), and it is described elsewhere [Ref. 18, Sect. 20]. In the original version of the ES1 algorithm, the stabilization step was omitted whenever the vector difference between successive normalized gradients was less than 10^{-8} , but now this is only a default value and the user is free to choose among other

possible values which might be more appropriate for the reacting system under study. This is controlled by the keyword DIFFD.

The ES1 method involves two auxiliary parameters, the stabilization step size DELTA2 and the tolerance in the allowed magnitude DIFFD of the vector difference between successive normalized gradients, and the accuracy achieved for a given amount of computer time can be sensitive to these values. These values have been approximately optimized in a systematic study of methods for calculating minimum energy paths (see Ref. 27 in Sect. 21). In that study, "standard" values were determined for both parameters. These "standard" values are: a) the stabilization step size DELTA2 equal to the gradient step size SSTEP, and b) the tolerance in the allowed magnitude of the vector difference between successive normalized gradients equal to 0.01. When the ES1 method is used with these choices for the stabilization step size and the tolerance in the allowed magnitude of the vector difference between successive normalized gradients, the algorithm is called the optimized ES1 method or ES1*.

The program also allows one to use the Page-McIver algorithm [Ref. 17, Sect. 20], rather than the ESD or Euler stabilization method, to compute the reaction path. In our implementation of the Page-McIver method, the first step away from the saddle point is again taken as described above, and subsequent steps are taken using a local quadratic approximation (LQA) as described in [Ref. 17, Sect. 20] or in the hard-copy manual for version 1.5.

POLYRATE can also use the variational reaction path algorithm based on the Euler steepest-descents method, called VRP(ESD) [Ref. 49, Sect. 21], to construct the reaction path.

Methods for calculating the reaction path are discussed further in Sect. 10.D.

In order to obtain the partition functions needed to evaluate $k^{\text{GT}}(T, s)$, a generalized normal mode analysis is performed at regular intervals along the reaction path. This is done at the points referred to as the save grid. The generalized normal mode analysis can be carried out in either rectilinear (Cartesian) or curvilinear coordinates.

The algorithm used for rectilinear coordinate calculations is discussed by Miller, Handy, and Adams [Ref. 7, Sect. 20]. In particular, for $s \neq 0$ the rectilinear or Cartesian generalized normal modes orthogonal to the reaction path are obtained from the projected force constant matrix

$$\mathbf{F}^P(s) = [1 - \mathbf{P}(s)]\mathbf{F}(s)[1 - \mathbf{P}(s)], \quad (3)$$

where $\mathbf{F}(s)$ is the full ($3N \times 3N$) Cartesian force constant matrix evaluated at a point on the MEP, and $[1 - \mathbf{P}(s)]$ is a projector used to project the direction along the reaction path as well as overall rotations and translations out of the space associated with the non zero eigenvalues. Diagonalization of $\mathbf{F}^P(s)$ gives 7 zero and $3N-7$ positive eigenvalues (6 zero and $3N-6$ positive eigenvalues for linear molecules, or 1 zero and $3N-1$ positive eigenvalues for adsorbed species) corresponding to the bound generalized normal modes orthogonal to the reaction coordinate. After the frequencies are calculated, the frequencies and corresponding eigenvectors are placed in canonical order (which is defined in Section 6.E). If the REORDER keyword is specified, the code will attempt to correlate the modes from point to point in terms of their physical characters rather than their magnitudes. Thus the first mode would be the mode with the highest overlap with the first mode at the previous point, rather than the highest-frequency mode, and so forth. The REORDER option would be useful for studying diabatic correlations of vibrational modes. Once the frequencies have been reordered anharmonicity corrections, the zero point energy, and the ground-state adiabatic energy are computed as they were at the saddle point.

The curvilinear generalized normal mode analysis is carried out by first transforming the Cartesian force constant matrix into a curvilinear coordinate system, then projecting out the gradient components, and finally diagonalizing the projected curvilinear force constant matrix according to Wilson **GF** method [Refs. 36 and 37 of Sect. 20] as described in Refs. 41, 43, 45, and 46 of Sect. 21. Generalized normal mode frequencies, the zero point energy, and the ground-state adiabatic energy are then computed as they were at the saddle point. Methods for calculating the curvilinear generalized normal mode frequencies are discussed further in Sect. 10.M.

Section 10 and the GTST book chapter give details of the WKB method for computing anharmonic vibrational energies.

Starting with POLYRATE version 4.0, vibrational modes corresponding to torsional motion can be treated with the hindered-internal-rotator approximation [Refs. 24, 29, and 34, Sect. 21].

6.G.2. NOSADDLE option

The NOSADDLE option is used only for strictly downhill association reactions. It requires the user to input an initial guess for a point on the MEP such that the variational transition state lies between this point and the product, which is the associated molecule. Let $s = 0$ denote the initial guess, and let s^P denote the value of s at the product. Typically, one will compute the MEP only up to some value s_2 that is less than s^P . Furthermore, the first few points are typically not useful because it takes a few steps for the path generated by the steepest descent algorithm to become sufficiently close to the MEP. If this occurs by $s = s_1$, where $0 \leq s_1 \leq s_2$, then it is the user's responsibility to ensure that the variational transition state lies between s_1 and s_2 .

6.H. Reaction-path curvature

The curvature of the reaction path is used in both the SCT and LCT calculations. The curvature of the reaction path is a vector with $F - 1$ components, where F is the number of internal degrees of freedom ($F = 3N - 6$ for a gas-phase reaction with a nonlinear reaction path, $F = 3N - 5$ for a gas-phase reaction with a linear reaction path, and $F = 3N$ for a solid-state reaction employing a fixed lattice, as in the embedded-cluster method). In this manual the components of the curvature vector are denoted B_{mF} , $m = 1, 2, \dots, F-1$, where m labels a generalized-normal-mode direction.

When using an analytic potential energy surface or electronic structure input data from unit fu30 or unit fu40, there are three choices for how to compute B_{mF} . The first two choices are based on eq. (98) of the GTST book chapter [Ref. 5, Sect. 21]. This formula involves a numerical derivative of the negative gradient, and this derivative may be computed by central or one-sided finite differences. The third choice is to use eq. (48) of a paper by Page and McIver [Ref. 17, Sect. 20]. Notice that eq. (98) of the GTST book chapter is the same as eq. (47) of Page and McIver paper. An advantage of eq. (48) is that it does not involve numerical differentiation of the gradient. Both formulas require the generalized-normal-mode eigenvectors so the curvature is calculated on the Hessian

grid (as opposed to the gradient grid). The evaluation of the curvature by taking the derivative of a quadratic fit to the gradient at three points (i.e., a central finite difference) is specified by the option *dgrad* of the keyword CURV; the evaluation by taking the derivative of a one-sided two-point fit to the gradient is called *oneside*, and the evaluation by eq. (48) of Page and McIver is called *dhess* because it involves a derivative of the Hessian. Although in principle, *dhess* can be used even when the reaction path is calculated by the ESD method, this can be dangerous because the accuracy of a numerical derivative requires a small step size. When the ESD method is used to calculate the reaction path, the gradient may be known on a much finer grid than the Hessian grid, and numerical differentiation of the Hessian may be inaccurate. If the *dhess* option is used with an ESD integration scheme, one should be especially careful to converge the Hessian with respect to INH.

These methods all break down (become unstable) at or very near to the saddle point. Thus the code does not obtain the curvature vector directly at the saddle point. Instead it obtains the value of B_{mF} at the saddle point by linear interpolation of the absolute values at the closest negative- s save grid point and the closest positive- s save grid point. That is, if the saddle point is s_{NS} , one obtains the curvature at the saddle point as an average of the values at s_{NS-1} and s_{NS+1} if INTMU equals 1, and one obtains the curvature at the saddle point as an average of the values at s_{NS-2} and s_{NS+2} if INTMU equals 3. The curvature at the saddle point is used to calculate $\mu_{eff}(s)$ at s_{NS} . There is an option to compute $\mu_{eff}(s)$ at s_{NS} , s_{NS-1} , and s_{NS+1} by linear interpolation of the values at s_{NS-2} and s_{NS+2} . This option corresponds to setting INTMU equal to 3 in the PATH section, this is currently the default.

When using the electronic structure data from unit fu31, the method for computing B_{mF} is automatically chosen based on whether or not a gradient is available at the closest point on both sides of the Hessian point. If gradients are available on both sides of the point then the three gradients are fit to a quadratic function and the derivative of this function is calculated and then evaluated at the Hessian grid point. If the gradient is only available on one side then the derivative is evaluated by one-sided difference. For obtaining the curvature vector at the saddle point the procedure described above is also used. However, if the user selects to perform a direct interpolation of the exponent in eq. 10 of Ref. 48 in Sect. 21, the IVTST-M calculation will obtain μ_{eff} at the saddle point from a spline fitting of the exponent as a function of the mapping variable z . See Section 6.N.2 for more IVTST-M details.

In contrast, when data is obtained from unit fu29, which is used for IVTST-1 calculations, the curvature at the saddle point is computed using an alternative formula presented as equation (49) in the Page and McIver article [Ref. 17 of Sect. 20], and its value at the "extra point" is computed again using eq. (48) of the same reference. The value at the saddle point is actually used in the IVTST-1 calculation, whereas the value at the extra point is computed for informational purposes only.

See Section 10.F for further discussion of the curvature calculations.

6.I. Transmission coefficients

The temperature-dependent ground-state transmission coefficient is the ratio of the Boltzmann average of the quantum energy-dependent ground-state transmission probability to the Boltzmann average of the energy-dependent transmission probability corresponding to classical-reaction-coordinate motion with other degrees of freedom in the ground state. Both transmission probabilities are zero for an energy lower than the maximum of the ground-state energy of the reactants and of the products. For ICVT and μ VT, the threshold energy for the classical transmission probability is the maximum of the adiabatic ground-state energy $V_a^G(s)$, i.e., V_a^{AG} ; it is $V_a^G(s=0)$ for TST and $V_a^G(s=s_*^{CVT})$ for CVT. One can factor the ground-state transmission coefficient into two factors. The first factor is a ratio of the Boltzmann average of the quantum transmission probability to the Boltzmann average of the classical transmission probability with the threshold energy at the maximum of the adiabatic ground-state energy. The second factor is a ratio of the Boltzmann average of the classical transmission probability with the threshold energy at the maximum of the adiabatic ground-state energy to the Boltzmann average of the classical transmission probability with the threshold at the value that is consistent with the level of variational theory being applied. In the FORTRAN unit fu6 output, the first factor is denoted as κ^{ZCT} , κ^{SCT} , κ^{LCG3} , κ^{LCG4} , and the second factor is denoted as $\kappa^{TST/CAG}$ and $\kappa^{CVT/CAG}$ for the TST and CVT levels of theory, respectively. Since this CAG factor is exactly 1 in the case of ICVT and μ VT, it is not listed in the output. The final result is denoted as, e.g., $\kappa^{CVT/ZCT}$ for a transmission coefficient evaluated with the ZCT (or MEPSAG) method at the CVT level.

The CAG factor is *required* for consistency in TST and CVT calculations (but not in ICVT and μ VT calculations), although it is often close to unity. The CAG correction was introduced in Refs. 10 and 44 of Sect. 20, and the reader is referred to those references for further discussion.

The MEPSAG, CD-SCSAG, and LCG3 or LCG4 transmission coefficients are simply denoted ZCT, SCT, and LCT, respectively, in recent papers, as discussed in Section 6.D above. The ZCT method should be considered simply as an approximation to SCT to be used when insufficient information is available to compute the reaction-path curvature. The SCT transmission coefficient is always larger than (or equal to) the ZCT one. The SCT transmission coefficient is expected to be larger than and more accurate than the LCT one when reaction-path curvature is small over the classically forbidden segment of the reaction path; the LCT transmission coefficient is expected to be larger than and more accurate than the SCT one when reaction-path curvature is large in the classically forbidden section of the $V_a^G(s)$ curve. These two situations are discussed in more detail in discussion remarks at Faraday Symposium 29 on Potential Energy Surfaces and Organic Reaction Paths [Ref. 45 in Sect. 20], and they are illustrated (in color) in an earlier review article [Ref. 46 in Sect. 20].

Whenever there is enough information available to calculate LCT transmission coefficients there is also enough data available to calculate SCT ones (although the converse is not true). In such cases the recommended best estimate is obtained by the microcanonical optimized multidimensional tunneling (μ OMT) approximation that is discussed in Section 6.M.2 below.

The optimized multidimensional tunneling approximations are reviewed below. In particular, Subsection 6.M.1 reviews the canonical optimized multidimensional tunneling (COMT) method, and Subsection 6.M.2 discusses the microcanonical optimized multidimensional tunneling method (μ OMT).

6.J. Reaction rates including tunneling

In this subsection, we describe the calculation of the reaction rate constants as discussed in Subsection 6.A.

For a given temperature T , various transmission coefficients $\kappa(T)$ are computed using methods selected by the user, all of which lead to different approximations to the rates. Next, the s -independent quantities are determined, i.e., the electronic, translational, rotational, and vibrational partition functions for reactant(s) and product(s), as well as the electronic partition function for the generalized transition state and the prefactor for the rate constant.

Next, for each s at which a generalized normal mode analysis was performed the program computes the rotational and vibrational partition functions of the generalized transition state to get the generalized free energy of activation at this value of s . Using the nearest points to the maximum, parabolic and quartic fits are used to obtain the interpolated maximum of $\Delta G_f^{GT,0}(T,s)$. The final location and generalized free energy of activation at this maximum are obtained from the quartic fit (the parabolic fit is just for comparison).

The calculation of the kappa factors is described next. The Wigner tunneling correction for conventional TST is also computed, but only for informational purposes. It is not a recommended method because it is just the first two terms in a slowly convergent or divergent series. For thermal reaction rates, the MEPSAG, CD-SCSAG, and LCG3 or LCG4 transmission probabilities are averaged over a thermal distribution and divided by the thermal average of the classical transmission probabilities. As mentioned previously, CD-SCSAG, LCG3, AND LCG4 methods take into account the coupling between the reaction path curvature and the vibrational modes. In this version of POLYRATE, the LCG3, LCG4, and CD-SCSAG methods treat vibrational modes harmonically. (The current version of the program will not perform LCG3 or LCG4 calculations if anharmonic options other than the hindered-rotator method are selected. In the case of CD-SCSAG, zero-point energies and vibrational partition functions can be treated using one of the various anharmonicity methods, and the CD-SCSAG kappa factor will change due to the change in the adiabatic ground state potential. However, the CD-SCSAG method is still based on a harmonic approximation.) The procedures for calculating the transmission probabilities and transmission coefficients for the MEPSAG method are described completely in the GTST book chapter, and the description of the CD-SCSAG method can be found in Refs. 29 and 31 of Sect. 21. For the LCG3 or LCG4 calculation the basic idea is in the GTST book chapter and additional details of the implementation are described in Refs. 21, 29, 31, and 34 of Sect. 21. (Note that the value of s along an LCG3 or LCG4 tunneling path is a continuous function of progress along the path, as

described correctly in the GTST book chapter and not as described in Ref. 21 of Sect. 21.)

Early versions of POLYRATE used a method called SCSAG. This has now been replaced by CD-SCSAG. The two methods are identical if the reaction-path curvature has only one nonzero component; otherwise they differ, and CD-SCSAG is based on a more correct analysis of the geometry of corner-cutting tunneling.

In the POLYRATE program, the Kronrod quadrature for MEPSAG and CD-SCSAG tunneling calculations is performed using the independent variable s and abscissa g_i rather than y and y_i ; see page 99 of the GTST book chapter.

Furthermore, for both the $\theta(E)$ integral over s , eq. (94), and the Boltzmann average, eq. (91) of the GTST book chapter, the range of integration is divided into 1-3 equal segments, and the same order of Kronrod quadrature is used in each. For the $\theta(E)$ integral we use NSEG2 segments with $(2*NQTH+1)$ -point Kronrod quadrature in each, and for the Boltzmann integral we use NSEG segments with $(2*NQE+1)$ -point Kronrod quadrature in each. Kronrod's method provides $NQTH$ -point and NQE -point Gauss-Legendre quadrature results as well. It is important to check the convergence of these integrals by comparing the N -point and $(2N+1)$ -point results.

The CD-SCSAG and LCG3 or LCG4 transmission probabilities are also used to calculate optimized multidimensional tunneling approximations called COMT and μ OMT, as discussed in Sections 6.M.1 and 6.M.2.

Using the various generalized transition state approximations and $\kappa(T)$ factors, various forward and reverse rate constants at temperature T can be obtained, e.g., the combination of an ICVT rate constant with an MEPSAG tunneling correction yields an ICVT/MEPSAG rate constant.

If the VTST-IOC correction is selected, the program updates all the energy, frequency, and determinant of moment of inertia information to the corrected values and recalculates the tunneling probabilities. A new set of corrected rate constants is then obtained.

If the VTST-ISPE correction is selected, the program spline fits the differences of the lower-level energies and the higher-level energies using the IVTST-M algorithm. The

interpolated values are then added on to the lower-level and a new set of corrected rate constants is obtained. See Section 10.J for a detailed description.

6.K. Imaginary-frequency modes transverse to the reaction path

When a generalized normal mode eigenvalue is negative, i.e., the harmonic frequency of a mode transverse to the reaction path is imaginary (recall that the eigenvalues are squares of the frequencies), the program takes the following actions:

1. When an independent-normal mode method is being used, the vibrational partition function of that mode is set equal to 1.0×10^{10} .
2. The zero-point energy of that mode, as required for the calculation of the vibrationally adiabatic ground-state potential curve for tunneling calculations, is set equal to zero.
3. The turning point distance of that mode is set equal to 1.0×10^{10} bohr for the LCG3 or LCG4 calculations.
4. For the CD-SCSAG method, no contribution from this mode is included whether the harmonic or anharmonic approximation is selected.

6.L. Equilibrium constant calculation

In the current version of the code this calculation is the last to be performed. However, the calculation is carried out in two different subroutines, TSOUT and FINOUT. If a TST-only calculation is requested, then all the rate calculations are carried out in the subprogram TSRATE, which calls TSOUT. TSOUT prints the rate constants to the fu6 file and the fu15 (summary) file, and it calculates and prints the equilibrium constants using the TST rate constants.

The subprogram RATE, which calls FINOUT, calculates the rate constants with canonical variational theory (CVT). FINOUT computes and prints final rate constants, activation energies, and equilibrium constants (from the CVT rate constants).

In both cases the equilibrium constants are the last quantities calculated in the respective subprograms.

6.M. Omitting the product

POLYRATE can be run without calculations on the product if reverse rate constants and equilibrium constants are not calculated. This requires specifying the FORWARDK keyword in the RATE section of the fu5 input.

What the code will do then depends on whether there is a WELLP section in the fu5 input. If there is a WELLP section, the interpolation extends to the product-side well as described at Section 3.1 of Ref. 48 of Section 21. If there is not a WELLP section, then the interpolation extends to the last nonstationary point on the product side. Thus $V_{\text{MEP}}(s)$ is cut off (down to the product value) at that point, and if tunneling extends into the region beyond the last nonstationary point on the product side, it will be overestimated.

6.N. Optimized multidimensional tunneling calculations

6.N.1. COMT method

The canonical optimized multidimensional tunneling (COMT) temperature-dependent transmission coefficient is the larger of the thermally averaged temperature-dependent transmission coefficients calculated by the CD-SCSAG method and the LCG3 or LCG4 method at a given temperature. The COMT method was first used in Ref. 28 of Sect. 21 (where it was simply called MT). It was first called COMT in Ref. 34 of Sect. 21.

6.N.2. μ OMT method

The microcanonical optimized multidimensional tunneling (μ OMT) temperature-dependent transmission coefficient is the ratio of the Boltzmann average of the μ OMT quantum transmission probability to the Boltzmann average of the classical transmission probability, where the μ OMT quantum transmission probability is the larger of the energy-dependent transmission probabilities evaluated by the CD-SCSAG method and the LCG3 or LCG4 method at a given energy. This method was introduced in Ref. 34 of Sect. 21. See Also Ref. 45 in Sect. 20.

6.O. Interpolated VTST and mapped interpolation

Calculating $V_{\text{MEP}}(s)$, $|\mathbf{H}(s)|$, and $\omega(s)$ as needed for variational transition-state theory and tunneling calculations from *ab initio* electronic structure theory can be computationally demanding. Thus, it is sometimes desirable to interpolate these quantities based on electronic structure data only at the reactant, product, saddle point, and optionally a small number of additional points (close to the saddle point) along the MEP. Variational transition-state theory based on interpolated data at a few points on the MEP [Ref. 43 in Sect. 20 and Refs. 26, 32, and 48 in Sect. 21] is called interpolated VTST or IVTST.

Three implementations of IVTST are included in this version of the POLYRATE program. The first is zero-order IVTST (IVTST-0) and the second is first-order IVTST (IVTST-1) using the global interpolation approach. The third is IVTST-M which is described in Section 6.N.2 and 6.N.3. Permutation of frequencies is not included in any of these methods. TST rates and ZCT transmission coefficients can be calculated with IVTST-0; in addition CVT rates and SCT transmission coefficients can be calculated with IVTST-1. Moreover, IVTST-M may be used for TST, CVT, ICVT, μ VT, and CUS rates and ZCT and SCT transmission coefficient calculations.

6.O.1. IVTST-0 and IVTST-1

IVTST-0 and IVTST-1 calculations [Ref. 26, Sect. 21] can be performed for bimolecular reactions by using the IVTST0 or IVTST1 keywords in the GENERAL section. Alternatively, IVTST-0 calculations can be performed using file fu50. Presently IVTST-0 and IVTST-1 methods are only available for bimolecular reactions with two reactants and two products.

The IVTST-0 method assumes that information is only available at three locations on the potential energy surface, namely at the reactant (R), saddle point, and product (P). Details of the interpolation procedures can be found in Ref. 26 of Sect. 21. The IVTST-0 rate constant is the conventional TST rate constant with ZCT tunneling. Note that ZCT tunneling is the only available tunneling method for IVTST-0. ZCT tunneling calculations for zero-order IVTST are carried out using

$$V_a^G(s) = V_{\text{MEP}} + \text{ZPE}(s),$$

where $V_{\text{MEP}}(s)$ is represented by an Eckart function and $\text{ZPE}(s)$ is obtained from interpolated ω values.

The IVTST-1 method requires information for one extra point ($s = s_1$) near the saddle point in addition to the saddle point, R, and P. The $V_{\text{MEP}}(s)$ curve for the IVTST-1 calculation is also approximated by an Eckart function with a constraint that the Eckart function goes through the extra point at $s = s_1$. To determine the location of the variational transition state, the forward standard-state free energy of activation profile, $\Delta G_{\text{f}}^{\text{GT},0}(T,s)$ is calculated using the interpolated $V_{\text{MEP}}(s)$, $|\mathbf{I}(s)|$, and $\omega(s)$. The SCT transmission coefficients can be obtained for first-order IVTST using the interpolated $\mu_{\text{eff}}(s)$.

IVTST-0 calculations can also be performed by using the keyword ICOPT in the GENERAL section and the keyword 0IVTST in the unit fu50 input (see Subsection 11.F). In the current version, this method is available only for bimolecular reactions with two products, and therefore the keywords MEPTYPEP and MEPTYPEP for the unit fu50 input should both be set to 2. In addition, the keywords MEPTYPEP and MEPTYPEP should not be set to *well*, and the keyword FREQMAT should not be used. The hindered rotor approximation is currently the only anharmonicity option that can be used with IVTST-0 when using file unit fu50 for the IVTST-0 calculation. At this time there are no anharmonicity options for IVTST-0 nor IVTST-1 calculations when using file unit fu29.

For the IVTST-0 calculations, only conventional transition state theory rate constants and ZCT transmission coefficients can be calculated. Variational transition state theory at the CVT level and both ZCT and small-curvature tunneling (SCT) transmission coefficients can be obtained for IVTST-1. Details can be found in Ref. 26 of Sect. 21.

6.O.2. Interpolated VTST by mapping (IVTST-M)

IVTST-M is a method for obtaining the information required for a VTST calculation all along the reaction-path from the information available at an arbitrary, but small, number of points. The only limitations on using the IVTST-M algorithm are that a saddle point exists and reaction path information is available at least one nonstationary point on each side of the saddle point. In addition, as in all IVTST methods, information is required for 3 – 5 stationary points (reactants, saddle point, products, and optionally one or two

wells). The level of IVTST-M is denoted by two numbers, H and G , indicating respectively the number of nonstationary points at which Hessians are available and the number of nonstationary points at which gradients are available. Thus, IVTST-M-6/54 indicates that the interpolation is based in 6 nonstationary Hessian points, at which geometries, energies, gradients, and Hessians are available, and 48 points (which is 54 minus 6) for which only geometries, energies, and gradients are available. The mapping procedure transforms $V_{\text{MEP}}(s)$, $\Pi(s)$, $\omega(s)$, and $B_{m,F}(s)$ from functions of s into functions of z where z is a new variable which always has a finite value, whereas s can have infinite values. The parameter z is determined as

$$z = \frac{2}{\pi} \arctan\left(\frac{s - s_0}{L}\right), \quad (1)$$

with s_0 and L being two parameters obtained from the forward and reverse barrier heights [Ref. 48 in Sect. 21].

The new $V_{\text{MEP}}(z)$, $\Pi(z)$, $\omega(z)$, and $B_{m,F}(z)$ functions are interpolated by using splines under tension.

The method employs different approaches for the interpolation of $V_{\text{MEP}}(s)$, depending on the number of reactants and products in the reaction. If the reaction is unimolecular in either the forward or reverse direction, a cubic polynomial for estimating 10 energies between the reactant (or product) and the first (or last) point on the reaction path (or between the reactant and the first point and between the product and the last point) is calculated before carrying out the spline under tension fit. The user can choose not to calculate the 10 extra energies; however, the default is to calculate them. If the forward reaction is bimolecular, the interpolation on the reactant side of the saddle point is based on the energies from the input plus 10 additional energies estimated from an Eckart polynomial with a range parameter which is a function of s . Similar considerations apply on the product side if there are two products.

If there are wells on the reactant and/or product side, the interpolation can be done using the information on these stationary points with the unimolecular procedures, which at least in principle can make the interpolation more accurate, especially when the well is deep and/or close to the saddle point. But if a well is present, it can also be ignored, and the interpolation can be carried out using reactants and/or products. If the well is ignored, then since for bimolecular reactions (or bimolecular sides of the reaction path) the interpolation is based on an Eckart potential, no point on the reactant side of the reaction

path can be used that has an energy lower than the reactants energy, and no point on the product side can be used that is more stable than the products.

The IVTST-M algorithm has been implemented to be used with the hooks and the electronic structure input files, unit fu30, fu31, and fu40. The method is fully operational when using the hooks and the unit fu31 input file. When using units fu30 and fu40, only the IVTST-M default options are available. To use the IVTST-M algorithm with unit fu30 LOPT(2) must be set equal to -1, and with fu40 MAXLPTS must be set equal to -1.

Whenever IVTST-M calls TSPACK for a spline-under-tension fit, it sets the variable IENDL equal to 3. This means that the two extra pieces of data needed for the spline fit are generated by the ENDSLPL subroutine using a local spline under tension.

6.O.3. The mapped interpolation (MI) algorithm with the hooks

As mentioned in Section 1 of this manual, the mapped interpolation (MI) algorithm can be used in two ways by POLYRATE. The first way is to perform IVTST-M calculations using electronic structure input file fu30, fu40, or fu31. This method is described fully in Sections 6.N.2, 10.J, and 11.D of this manual. The second way is in conjunction with the hooks; this requires some additional comments, which are given in the rest of this section.

A calculation with the hooks involves two stages. At the end of the first stage, one has stored s , $V_{\text{MEP}}(s)$, and the gradients on the fine grid and s , $V_{\text{MEP}}(s)$, $\omega(s)$, $B_{mF}(s)$, and $|\mathbf{I}(s)|$ are stored on the save grid. (The grids are defined in Sect. 7.A) Both grids extend from $s = \text{SLM}$ on the reactant side to $s = \text{SLP}$ on the product side, and there are $G + 1$ points (including the saddle point but excluding reactants and products) on the fine grid and $H + 1$ points (including the saddle point but excluding reactants and products) on the save grid. Then mapped interpolation is carried out to add $\text{EXNSTEP} * \text{EXSTEP}$ points on each side of the grids. Furthermore SLM is replaced by $\text{SLM} - \text{EXNSTEP} * \text{EXSTEP}$, and SLP is replaced by $\text{SLP} + \text{EXNSTEP} * \text{EXSTEP}$. As should be clear from Section 11.A.7, EXNSTEP and EXSTEP may actually be different on the two sides.

6.P. Interpolated optimized corrections (VTST-IOC), a form of dual-level VTST

In VTST calculations with approximate potential energy surfaces, it is sometimes desirable to correct the classical energies, vibrational frequencies, and moment-of-inertia determinants of some points along the MEP (e.g., the reactant, saddle point, product, van der Waals complex, ion-molecule complex, etc.) with accurate (higher-level) results. This is accomplished by applying correction procedures to the calculated energy, frequencies, and moment-of-inertia determinant along the MEP. The corrections are calibrated such that the corrected results match the accurate values at those selected points, and they correspond to interpolating these corrections at other points. When the corrections involve data from higher-level optimizations of the stationary points, the method is called VTST-IOC, a special case of VTST-IC. The original correction procedure for frequencies is denoted ICA, and two improved procedures are denoted ICR and ICL. (In general we recommend the ICL procedure.) The ICA procedure was introduced in version 5.1 of POLYRATE, and the ICL and ICR procedures were introduced in version 7.0. All three procedures are based on the correction at 3 points, in particular, the saddle point and two stationary points, one on each side of MEP.

There are also two procedures for correcting $V_{\text{MEP}}(s)$ in IOC calculations: the original method, which was called NOECKART in versions 7.2 – 7.9.2 of POLYRATE but now is called the single-Eckart (SECKART) method of interpolating V_{MEP} , and the newer method, which was called ECKART in versions 7.2 – 7.9.2 but now is called double-Eckart (DECKART). The ECKART option was introduced in version 7.2. As implied by the names of the correction, is either a single Eckart function or a difference of two Eckart terms. If there is an appreciable well on either side of MEP, it is recommended that one use the molecular complex corresponding to the well as the stationary point on that side of the saddle point; otherwise it is recommended that one use the reactant or the product. However, if the well on the MEP is not very deep or if accurate data for the complex corresponding to the well are not available, the user can neglect the well and choose an appropriate functional form for the correction function as if there were no well at all.

As discussed in Section 1, the VTST-IOC method can be used either with an analytic potential energy surface or with electronic structure input data for the lower-level surface. In either case, the information calculated from the potential energy surface or read from the electronic structure input file will be corrected by the higher-level values.

The functional forms we use for the correction functions are an Eckart function, a hyperbolic tangent, a cut-off gaussian, or a cut-off hyperbolic tangent, depending on the shape of MEP under consideration and the relative correction values at the 3 stationary points. See Refs. 36 and 44 of Sect. 21 for details. Details for the ICA procedure are available in Subsection 10.H and in Ref. 36 of Sect. 21 and details for the ICR and ICL procedures are available in Subsection 10.H and in Ref. 44 of Sect. 21. Details of the double Eckart procedure are available in Ref. 44 of Sect. 21.

If the higher-level data is obtained at electronic structure level X, and the lower-level data is obtained at level Y, the VTST-IOC calculation may be denoted X///Y. The triple slash notation is similar to the double slash notation that is widely used in electronic structure calculations. In particular, // means “at a stationary point geometry optimized by...” whereas /// is loosely translated as “along a reaction path calculated by...”. More specifically, though, we reserve the IOC and triple slash notations for algorithms where the higher-level information supplied to “correct” the transition state data involves a higher-level geometry optimization. If only single-point energies are used, the ISPE and double slash notation should be used.

6.Q. Interpolated algorithm for large-curvature tunneling calculations

In LCT calculations, for each energy, a tunneling path is defined as a straight line in isoinertial coordinates between two reaction-coordinate turning points lying on the MEP, on the reactant side and on the product side of the maximum of the vibrationally adiabatic potential. This straight-line is divided into three zones, two of them are in the reactant and product valleys, respectively, and they are vibrationally adiabatic, and another zone describes the middle zone of the straight-line path and is vibrationally nonadiabatic. In the interpolated LCT (or ILCT) algorithm, every time the potential energy is needed along the nonadiabatic segment of the tunneling path, this energy is calculated by means of the spline fit to a number of already calculated points. Details of the methods are available in Ref. 55 of Sect. 21.

7. PROGRAM STRUCTURE

7.A. Overview of program flow

As mentioned in the Introduction, the POLYRATE program has several possible ways to obtain the reaction-path information necessary for rate constant calculations. The first option is to calculate this information directly from a set of interface subprograms called hooks. The second is to read this information from an electronic structure input file (fu30, fu31, or fu40) consisting of geometries, gradients, Hessians, etc. at points on the MEP obtained from electronic structure calculations. The third option is to read the information from a restart file which has been saved from a previous POLYRATE run. The fourth option is to read in higher-level “corrections” at selected points along the MEP; these corrections are read from file fu50 for VTST-IOC and from file fu51 for VTST-ISPE. Finally, for the IVTST calculations, electronic structure data at selected points may be read from file fu29 (IVTST-0 and IVTST-1) or fu50 (IVTST-0). These options are implemented in the POLYRATE program as follows.

First, the MAIN program reads all the keyword input. These keywords are analyzed for possible conflicts (i.e., unsupported combinations of options) in subroutine OPTION. If it is a restart run then the program reads in the restart information in RESTOR, and the rate constant calculation proceeds as described in the last paragraph of this section. Otherwise, all other input data from unit fu5 are read in and checked by READ5 and REDGEO.

The GENERAL section of keyword input (Section 11.A.2) is always read first. Depending on the options chosen the properties of the reactants, products, and generalized transition states are calculated by calls to the hooks or read in from an electronic structure input file (or both). The user can choose at most one of the two switch keywords IVTST0 and IVTST1, or use the DL keyword. This choice plus the choice made for the POTENTIAL keyword in the ENERGETICS section (Section 11.A.3) determines the basic outline of the rest of the calculations as indicated in the following table. That table assumes that the hooks call SURF and SETUP, which is the standard way to use the hooks when POLYRATE is used as a stand-alone code. (Subprograms SURF and SETUP are described in Subsection 8.C, and a more complete description of the hooks is provided in Subsection 8.C).

DL	IVTST0	IVTST1	POTENTIAL	Action
<i>none</i>	off	Off	<i>hooks</i>	Single-level calculation based on whole reaction path calculated from an analytic PES.
<i>none</i>	off	Off	<i>unit30, unit31, or unit40</i>	Single-level direct dynamics based on whole reaction path calculated from an electronic structure input file on unit fu30, fu31, or fu40
<i>none</i>	on	Off	<i>unit29 or unit50</i>	Single-level direct dynamics IVTST-0 calculation based on electronic structure input data from unit fu29 or fu50
<i>none</i>	off	On	<i>unit29</i>	Single-level direct dynamics IVTST-1 calculation based on electronic structure input data from unit fu29.
<i>ic</i>	off	Off	<i>hooks</i>	Dual-level direct dynamics with lower-level data from analytic PES and higher-level data from unit fu50
<i>ic</i>	off	Off	<i>unit30, unit31, or unit40</i>	Dual-level direct dynamics with lower-level data from unit fu30, fu31, or fu40 and higher-level data from unit fu50
<i>ispe</i>	off	Off	<i>hooks</i>	Dual-level direct dynamics with lower-level data from analytic PES and higher-level data from unit fu51
<i>ispe</i>	off	Off	<i>unit30, unit31, or unit40</i>	Dual-level direct dynamics with lower-level data from unit fu30, fu31, or fu40 and higher-level data from unit fu51

In the analytic potential energy surface case, the equilibrium geometries and properties of the reactant and product species are optionally obtained in subroutine REACT. For polyatomic species, REACT calls POLYAT, and the user must supply an estimate of the equilibrium geometry. POLYAT calls NEWT to perform a Newton-Raphson search for the location where the first derivatives of V are zero, CENTER to place the origin at this location and evaluate the determinant of the moment of inertia tensor, and NORMOD to do a vibrational analysis. The vibrational analysis involves determining the normal mode eigenvectors and frequencies, and, if desired, one of several procedures for including anharmonicity is invoked in ANHARM. If a diatomic species is treated as a Morse oscillator (DIATOM keyword) REACT calls DIATOM, and the equilibrium geometry, dissociation energy, and range parameter must be supplied by the user.

Next, the program will optionally determine the saddle point geometry and properties. This is done in SADDLE and MAIN, which in turn call the same subroutines used by POLYAT. The program will handle 0 or 1 saddle points. For one saddle point the user must supply an approximate geometry for its location.

When using the hooks, POLYRATE starts either at the saddle point just determined or from an input starting-point geometry, then computes the minimum energy path (MEP) by a choice of several methods (done in the PATH section of file fu5). The distance along the MEP is called s ; the user can make s be zero at the saddle point, negative in the reactant region, and positive in the product region. The initial step can be based on a cubic expansion of the potential energy surface determined from the normalized eigenvector of the imaginary-frequency normal mode and from the curvature vector, both of which are calculated at the saddle point, or the initial step may be taken along the normal coordinate for the unbound motion at the starting point. When the default ESD method is used (RPM keyword set equal to *esd*), the negative of the gradient in isoinertial coordinates is followed using a fixed gradient step size DEL (specified by the SSTEP keyword). At regular intervals (specified by the INH keyword) along the MEP, the program computes the generalized-transition-state properties necessary to evaluate the generalized free energy of activation and to perform semiclassical tunneling corrections. (We emphasize that both DEL and the regular intervals are measured in the isoinertial coordinates. Thus the units are sometimes called mass-scaled bohrs instead of bohrs. Thus if REDM, specified by the SCALEMASS keyword which is described in Subsection 11.A, is changed, DEL and the regular intervals will have different meanings.) Note that the grid

determined by DEL is called the fine grid or the gradient grid. The grid defined by the INH keyword is called the save grid or Hessian grid. This is discussed further in Subsection 10.D. The moment of inertia calculation and Miller-Handy-Adams projections (eq. 31 of Ref. 5) or curvilinear-reaction-coordinate projections (eq. 17 of Ref. 41) are carried out in PROJCT; the vibrational analysis and the calculation of the effective mass for tunneling are done in NORMOD. In addition, if there is a saddle point the mapped interpolation (MI) algorithm can be used to interpolate the energy, the frequencies, the determinant of the moment of the inertia tensor, and the curvature components to extend one or both ends of the MEP for tunneling calculations. (See Sections 6.N.3 and 8.A for more details.) When there is no saddle point a simple exponential extrapolation of the classical and adiabatic energies and the other quantities used for tunneling calculations can be used at one or both ends of the MEP. (See Sect. 10.A for more details.) The EXFIRST and EXSECOND keywords in the PATH section control interpolation (extrapolation) options. Interpolation (extrapolation) is especially useful in situations where a minimum is reached in the classical potential before the adiabatic barrier is low enough for computing converged tunneling probabilities. Notice that extending the grid is sometimes called extrapolation, but it is also called interpolation because one used the reactant or product information that technically it is indeed interpolation.

When the MEP calculation is completed, the reaction-path information can be saved by writing it to unit fu1 for later restart runs. We note that there is an upper limit of NSDM points at which the reaction path information can be saved in a given POLYRATE executable file. The parameter NSDM is defined in the INCLUDE file param.inc (see Subsection 5.A.2).

In the electronic structure input file case, the geometries, energies, vibrational frequencies or harmonic force constant matrices, and (optionally) higher-order force constants of the reactant(s), product(s), saddle point, and generalized transition states are read in from unit fu30, fu31, or fu40. (A more detailed statement of the available input options is given in Section 9.) The number of points on the MEP available from electronic structure calculations is usually small; thus when converging tunneling probabilities it is sometimes necessary to extend the reaction path beyond the last input data point on one or both sides of the MEP. This is done by specifying larger values for the SLM and SLP keywords in the PATH section. The special interpolation procedures are described in Subsections 9.C and D.

The generalized free energy of activation is then computed at each of the points on the save grid (at which the reaction path information was saved), and the CVT rates are calculated for an input set of temperatures (done in RATE). We find the maximum generalized free energy of activation by local interpolation using 5-point Lagrangian interpolation. The result calculated with 3-point Lagrangian interpolation is printed as a check. ICVT calculations are carried out similarly if requested.

Various tunneling corrections can also be included by a call to KAPVA. The final output of forward and reverse reaction rate constants with and without the various tunneling corrections, as well as output of activation energies, analyses of bottleneck properties, equilibrium constants, and the ratio of the CVT to the conventional transition state theory rate constants, are provided by FINOUT.

If the VTST-IOC option is selected (DL keyword set to *ic* or ICOPT keyword specified), the program will then update (done in ZOCUPD) all the MEP information based on corrections calculated from the higher-level values read from the unit fu50 input file and calculate a new set of rate constants based on the corrected values by the same procedures as described above.

If the VTST-ISPE option is selected (DL keyword set to *ispe*), the program will update the energies along the MEP based on a spline fit to the differences of the lower-level energies and the higher-level energies read from the unit fu51 input file. A new set of rate constants is then calculated, based on the corrected energies, by the same procedures described above.

If the IVTST-0 method is selected, after the MAIN program reads all the information provided at reactants, products, and saddle point, subroutine GIVTST is called. Then the process of generating the information required at points along the reaction path by interpolation is started using specific routines for IVTST instead of the ones employed for calculations based on analytic potentials or fu30, fu31, and fu40 input files. If the IVTST-1 method is selected, the electronic structure information at the stationary points plus the data at an extra point located close to the transition state is read by the MAIN program from file fu29 and then the subroutine GIVTST is called.

7.B. Determining the sign of s

If there is a saddle point, the reaction coordinate s is, by convention, positive on the product side of the saddle point and negative on the reactant side. In the program, however, the default is that s is defined as positive in the direction of the unbound normal mode at the saddle point, and it is defined as negative in the opposite direction. When one is using the hooks, the definition of the sign of s does not affect the calculated rate constants when LCG3 or LCG4 calculations including tunneling into excited states are not carried out, but it does affect the LCT probabilities when contributions from tunneling into excited states are included, since the current coding of that case requires consistency with the convention. When one is using fu30, fu31, or fu40, the choice of SIGN also affects whether SCT calculations or LCT tunneling into the ground state are correct, as explained in the next paragraph. Also, confusion might arise when one specifies the range of the reaction path to be calculated or discusses the location of the optimized variational transition state. We therefore advise users to override the default, if necessary, to follow the established convention of s being positive on the product side. The following procedure is recommended to ensure this. First, make a run without computing the path (do not include the PATH section) and without computing the properties of reactants and products (do not include REACT1, REACT2, PROD1, and/or PROD2 sections), and look at the normal mode eigenvector corresponding to the normal mode with imaginary frequency at the saddle point. If this vector points towards products, set the SIGN variable keyword in the PATH section to *product* in full runs. If not, set it to *reactant* (or omit it, since *reactant* is the default). There is another keyword, IDIRECT, in the PATH section which determines the direction of the first step from the saddle point in calculating the MEP. If SIGN is set correctly and if IDIRECT is set to 1 (which is the default), the MEP toward the product side (which is determined by the keyword SIGN discussed above) will be calculated first and the reactant-side MEP will be calculated after the limit on the product-side MEP is reached. If SIGN is set correctly and if IDIRECT is set to -1, the reactant-side MEP will be calculated first instead. The value of IDIRECT should not affect the calculated results.

In practice, the SIGN variable works differently with the hooks and with files 30, 31, and 40. When one is using the hooks, SIGN actually changes the sign of the value of s that is assigned to given geometry on the reaction path. When one is using the electronic structure input file options, one cannot change the sign of s . However, it is still necessary that SIGN be set correctly. If SIGN is set wrong, then the reaction path curvature at the

saddle point will not be computed correctly. The correct setting for SIGN when one is using fu30, fu31, or fu40 is determined by the following rule: The positive direction of the imaginary frequency normal mode at the transition state must point to geometries that are associated with positive values of s in the input data (even if positive s corresponds to reactants rather than products); if this alignment is not observed then the run should be repeated with SIGN changed to the other choice.

7.C. Restart options

There are two ways to restart calculations in POLYRATE.

The first restart option is controlled by the RESTART keyword in the GENERAL section. This option is designed to allow calculations at additional temperatures without recalculating the information stored on the save grid. This option uses fu1. If the VTST-IOC option is used with RESTART, one can change the lower level but not the higher level, i.e., the fu50 file cannot be changed for the RESTART run. Furthermore, if the original run was not a dual-level run, the restarted run cannot be a dual-level run.

The second restart option allows one to restart calculations without repeating all of the work for the stationary points (reactants, products, and/or saddle point). This restart option is controlled by the STATUS keyword in the REACT1, REACT2, PROD1, PROD2, and START section. Each of the input sections for the stationary points (REACT1, REACT2, PROD1, PROD2, and START) allows one to input optimized geometries, energies, and frequencies from previous jobs. The user also has the option of only calculating reactants, or only calculating products, or only calculating a saddle point to obtain the optimized geometries, energies, and frequencies for the given stationary point before proceeding with the rest of the calculation. Thus it is possible to optimize the stationary points in separate runs, and then read in the data for those stationary points in later runs to continue. To do this, the user can run POLYRATE with only the *GENERAL, *ENERGETICS, *SECOND, *OPTIMIZATION, *REACT1, and *REACT2 sections in the fu5 file. If the calculation is a supermolecule calculation then both reactant 1 and reactant 2 must be specified in the *REACT1 and *REACT2 sections. However, if the calculation is not a supermolecule calculation then it is possible to calculate reactant 1 and reactant 2 separately using only the *REACT1 section and leaving out the *REACT2 section. Note that when calculating only reactant 2, the information for reactant 2 is placed in the *REACT1 section and the *REACT2 section is left out. The same scenario holds for

products, i.e., one can include *REACT1 or *REACT1 and *REACT2 but leave out *START. If the user wishes to optimize the saddle point without optimizing the reactants and products then the saddle point information is placed in the *START section and POLYRATE is run with only the *GENERAL, *ENERGETICS, *SECOND, *OPTIMIZATION, and *START sections in the fu5 file. An fu61 file, which contains the optimized properties, is created after every POLYRATE run. The information on this file can be used to build a complete fu5 file or update an existing fu5 file with optimized information. Thus, effectively, one can optimize each stationary point separately or in any combination of stationary points, add the newly optimized information to an existing or a new fu5 file, and restart the calculation to optimize the remaining stationary points or complete the full calculation. See Section 11.A.6.

8. CALCULATIONS USING THE HOOKS

8.A. Methods for interpolating beyond SLM and SLP with the hooks

This section describes the procedures used when interpolating between the SLM end point on the save grid and the reactants and between the SLP end point on the save grid and the products, respectively. There are two procedures that are used.

The first procedure is used when the reaction being studied has a saddle point. (In this case the SADDLE keyword in the PATH section is specified or taken as the default.) Reactions with saddle points either do not involve interpolation, or they use the mapped interpolation (MI) algorithm to extend the reaction path beyond SLM and SLP. The options for this interpolation are controlled by the EXFIRST and EXSECOND keywords in the PATH section. The MI algorithm is described in Sections 6.N.2, 6.N.3, and 10.J and in Ref. 48 of Sect. 21.

The second procedure is used when the reaction being studied does not have a saddle point. For this type of reaction the NOSADDLE keyword must be specified in the path section. Reactions without saddle points always use a two-point extrapolation to extend the reaction path beyond SLM and SLP. The options for this extrapolation are controlled by the EXFIRST and EXSECOND keywords in the path section. The two-point extrapolation is described in Section 10.A.

8.B. Zero of energy

In POLYRATE the variable EZERO always denotes the energy that corresponds to reactants in the potential energy subroutine, and it is used to convert the potential energy to the zero of energy that is used in the rest of POLYRATE. EZERO refers to the energy of a single reactant (for a unimolecular reaction) or the sum of energies of infinitely separated reactants (for a bimolecular reaction). In the language of potential energy surfaces, EZERO denotes the potential energy of the reactant or reactants at their classical equilibrium geometries. In the language of electronic structure calculations, EZERO denotes the sum of the electronic energy and the nuclear repulsion (for *ab initio* calculations). It is necessary to understand this variable in order to understand the input

options available with the EZERO keyword in the GENERAL section; the rest of this section provides the necessary explanation.

In POLYRATE the user must ensure that EZERO is set equal to a value consistent with the potential energy subprogram (when an analytic potential energy subprogram is used) or with the electronic structure data (when the electronic structure input file option is used). This can be accomplished in one of two ways: (i) by setting the EZERO keyword to *calculate* and letting the program calculate the zero of energy, or (iii) by setting the EZERO keyword to *read* and directly inputting a nonzero value for EZERO which is consistent with the potential energy subprogram.

Setting the EZERO keyword to be equal to *calculate* causes the program to calculate the zero of energy after the reactants have been optimized to their equilibrium geometries at "infinite" separation. Because the zero of energy is not computed until the optimization procedure, all energies for the reactant species which are printed to the long output file on unit fu6 before the zero of energy is computed will have an incorrect zero of energy. For a unimolecular system the zero of energy is set equal to the potential energy of the system at its optimized reactant equilibrium geometry. For a bimolecular system the zero of energy is set equal to the sum of the potential energies of the two reactants after both reactants have been optimized to their equilibrium geometries. This can be done with the SUPERMOL option by placing the two reactants at a large separation distance given implicitly by the two geometries specified by the GEOM keyword in the REACT1 and REACT2 sections of the unit fu5 input data file. (Note that, when the SUPERMOL option is used, this sum is given by the energy at the end of the second reactant's optimization).

8.C. User-Supplied analytic potential energy surface

8.C.1. Interface

The protocol of the subprograms for evaluating the potential energy from an analytic function is as follows. The user must supply two subroutines, SETUP and SURF, which accept and return all information from and to POLYRATE in atomic units. The subroutine SETUP has one integer type variable, $n3tm$, in its parameter list:

subroutine setup (N3TM)

In the argument list of the call statement to SETUP the POLYRATE code passes the variable N3TM where:

N3TM = $3 \times \text{NATOMS}$
(This variable is used to dimension the arrays for the Cartesian coordinates and the derivatives of the energy with respect to the Cartesian coordinates for a given system.)

NATOMS = maximum number of atoms in the POLYRATE code
(This variable is set in the INCLUDE file param.inc.)

SETUP may optionally open data file(s) which contain the potential parameters, read these parameters, set up any necessary data, and close any file(s) which were opened. SETUP is called once and only once in each complete run of the program.

Note that SURF and SETUP are called by the hooks. In particular, SETUP may be called by PREP and/or PREPJ, and SURF may be called by EHOOK, GHOOK, HHOOK, and OHOOK. No other routines call SURF and SETUP.

Users should dimension arrays in SETUP carefully since inconsistency of array dimensioning is the major potential source of error when passing information between subprograms. Also, users should resolve any conflict in the assignment of the input and output I/O units between POLYRATE and SETUP. For the I/O unit numbers used by POLYRATE, see Section 17. As noted there, unit 4 is reserved for use by the potential routines, i.e., by SETUP. If SETUP requires more than one file, unit 7 is recommended for the second file. Alternatively the FU# parameters in the param.inc INCLUDE file can be modified to avoid conflicts between the FORTRAN units used in POLYRATE and those used by SETUP and SURF. In addition, users should not name subroutines with the same name as an existing POLYRATE subroutine. A list of subroutine names and a short description of the subroutines used by POLYRATE are given in subsection 15.A.1.

The subroutine SURF evaluates the potential and its first derivatives as functions of the Cartesian coordinates, and it has 4 elements in its parameter list. These are V, X, DX, and N3TM, as follows:

```
subroutine surf (V, X, DX, N3TM)
```

where:

- V = potential energy in hartrees
- X = array of dimension N3TM which contains the Cartesian coordinates in bohrs of all the atoms in the system
- DX = array of dimension N3TM which contains the first derivatives (in hartrees per bohr) of the energy with respect to each of the Cartesian coordinates. DX should be determined analytically if possible as all higher-order derivatives needed by POLYRATE are calculated numerically from these first derivatives.
- N3TM = as defined above

The arrays X and DX must be of dimension N3TM within the subroutine SURF.

The order of the coordinates is (x, y, z) for atom 1, (x, y, z) for atom 2, ..., (x, y, z) for atom N, where the order of the atoms is determined by the order used in the ATOMS list keyword in the GENERAL section. (see Subsection 11.A.2).

Note that whenever the potential is read from FORTRAN unit fu30 file, SURF and SETUP are not needed and may be replaced by their dummy versions, for example, those included in the dumpot.f file.

During the calculation of the MEP, the quantities V, X, and DX are optionally written (using the PRSAVERP keyword in the PATH section) by POLYRATE to the file linked to the FORTRAN unit fu6 for each point at which reaction path information is saved (each save grid point). Since the user is often interested in the values of additional quantities calculated by subroutine SURF (such as interatomic distances, curvilinear internal coordinates, or various contributions to the potential energy) at these save points, we also provide the common block

```
parameter (npotpt = #)  
common/surfc/ potinf(npotpt)
```

where:

- # = an integer value set in the param.inc file (See Subsection 5.A.2)

in the POLYRATE program as a means of passing such information from subroutine SURF to the main program, in which the array POTINF is optionally written to the file linked to the FORTRAN unit fu6. (This is controlled by the use of the POTINF keyword in the PATH section). The maximum number of pieces of data from the potential which may be printed at each save grid point is 99. Unlike the parameter lists described above, the use of the common block SURFCM in the user-supplied SURF subroutine is user-driven. The information stored in the array POTINF is printed exactly as stored. Note that values assigned to the array POTINF have no effect on any values calculated by POLYRATE. Also note that except for the common block SURFCM, the user should avoid using common block names in subroutines SETUP and SURF that match those used in the POLYRATE program. To make this easier, all common block names end in COM or CM. All common blocks used by POLYRATE are given in the file common.inc and are summarized in the table below:

amascm	Anhcom	cerdem	curvcm
cutfcm	Cvcicm	der2cm	dmylcm
dtdscm	Eigcom	emaxcm	ezer0cm
finocm	f1com	geocm	geopcm
hnrtcm	Iblacm	icrvcm	icutcm
ieigcm	Igeocm	igepcm	if1com
imascm	Imincm	ipgcom	iphscm
iprmcm	Ipthcm	iqdcm	iqdkcm
iqstem	Iratcm	irccm	irescm
irpgcm	Irphcm	isplcm	itmpcm
itx2cm	itx3cm	itx5cm	ivacom
ivrcom	Iwbzcm	iwbkcm	ixtcm
izoccm	Kchrcm	kflcm	kintcm
lenecm	Lprdcn	ltuncm	lzoccm
massem	Mincom	npocom	outpcm
pathcm	Pcvcm	pefcm	perccm
pgcom	Phascm	prefcm	projcm
ptbcm	Qdkacm	qdtcm	ratecm
rescom	Rphacm	rphcm	rphgcm
rphwcm	Rstocm	r40acm	r40ccm
r40icm	r40lcm	splhcm	splncm
sqstem	stopcm	sum1cm	surfcm
tempcm	thdecn	Trnpcm	vacom
txxcm	tx1cm	tx2cm	tx3cm
tx5cm	varcom	Vtmucm	vtusem
wkbcom	wkbzcm	Xtraem	zoccm
efprecm	fcfacpj	itjac	intr
jac	ifrqsc1	fcscale	prova
spiccm			

8.C.2. Utility routines

The POLYRATE program assumes that the user-supplied potential energy function subprogram SURF accepts the values of the atomic Cartesian coordinates and returns the potential energy and the derivatives of the potential energy with respect to the atomic

Cartesians. However, not all potential energy functions are written in terms of atomic Cartesians; some potential energy functions are written in terms of bond lengths. With this in mind, four utility subprograms are provided with POLYRATE. Two of the utility subprograms are for three-body systems and the other two are for four-body systems.

Two types of utility subprograms are provided. One type converts atomic Cartesians to bond lengths, and the other uses the chain rule to convert the derivatives of the energy with respect to the bond lengths to derivatives with respect to the atomic Cartesians.

The subprograms COORD#, where # = 3 for a three-body system and # = 4 for a four-body system, convert the atomic Cartesians to bond lengths. In the case of COORD3, nine Cartesians are converted to three bond lengths. The order of the Cartesians and the bond lengths are as follows:

X(1) — X(3):	X, Y, Z for atom A
X(4) — X(6):	X, Y, Z for atom B
X(7) — X(9):	X, Y, Z for atom C
R(1):	R(A–B)
R(2):	R(B–C)
R(3):	R(C–A)

The Cartesians are passed to COORD3 through the parameter list (X, N3TM) (X and N3TM are defined in Subsection 8.C.1) and the bond lengths are passed to the potential subprogram through the common block POTCM:

```
/potcm/ r(3), energy, dedr(3)
```

COORD3 assumes that all the coordinates (Cartesians and bond lengths) are in atomic units. The subprogram also assumes that the potential is written such that the bond specifications are as defined above. For example, if the potential for ClHBr is written such that

R(1):	R(Cl–H)
R(2):	R(H–Br)
R(3):	R(Br–Cl)

the user must define Cl as the first atom (atom A), H as the second atom (atom B), and Br as the third atom (atom C) in the input data in unit fu5.

The subprogram COORD4 does the same calculation as COORD3 but for a four-body system. In COORD4 the coordinates are defined as follows:

X(1) — X(3):	X, Y, Z for atom A
X(4) — X(6):	X, Y, Z for atom B
X(7) — X(9):	X, Y, Z for atom C
X(10) — X(12):	X, Y, Z for atom D
R(1):	R(A–B)
R(2):	R(A–C)
R(3):	R(A–D)
R(4):	R(B–C)
R(5):	R(B–D)
R(6):	R(C–D)

The subprograms CHAIN#, where # = 3 for a three-body system and # = 4 for a four-body system, convert the derivatives of the energy with respect to the bond lengths to derivatives with respect to the atomic Cartesians. In the case of CHAIN3, three derivatives with respect to the bond lengths are converted to nine derivatives with respect to the atomic Cartesians. The order of the derivatives is as follows:

DEDR(1):	DEDR for the bond length A–B
DEDR(2):	DEDR for the bond length B–C
DEDR(3):	DEDR for the bond length C–A
DX(1):	DEDX for atom A
DX(2):	DEDY for atom A
DX(3):	DEDZ for atom A
DX(4):	DEDX for atom B
DX(5):	DEDY for atom B
DX(6):	DEDZ for atom B
DX(7):	DEDX for atom C
DX(8):	DEDY for atom C
DX(9):	DEDZ for atom C

The derivatives with respect to the bond lengths are passed through the common block POTCM as defined above, and the Cartesian information is passed through the parameter list (X, DX, N3TM). This subprogram assumes the same definitions as COORD3.

The subprogram CHAIN4 performs the same calculation as CHAIN3, but for a four-body system with the following coordinate definitions:

DEDR(1):	DEDR for the bond length A–B
DEDR(2):	DEDR for the bond length A–C
DEDR(3):	DEDR for the bond length A–D
DEDR(4):	DEDR for the bond length B–C
DEDR(5):	DEDR for the bond length B–D

DEDR(6):	DEDR for the bond length C-D
DX(1):	DEX for atom A
DX(2):	DEY for atom A
DX(3):	DEZ for atom A
DX(4):	DEX for atom B
DX(5):	DEY for atom B
DX(6):	DEZ for atom B
DX(7):	DEX for atom C
DX(8):	DEY for atom C
DX(9):	DEZ for atom C
DX(10):	DEX for atom D
DX(11):	DEY for atom D
DX(12):	DEZ for atom D

These subprograms can be used with any three- or four-body potential which is written in terms of bond lengths. For example, for a three-body potential made up of two subprograms, PREPOT and POT, where PREPOT does the initializations for the potential, i.e. performs the role of SETUP, and POT does the actual potential and derivative evaluations given the geometry in terms of bond lengths, the user could provide SETUP and SURF routines to link POLYRATE with PREPOT and POT.

The subprogram SETUP would simply call PREPOT. For example:

```

subroutine setup (n3tm)
implicit double precision (a-h, o-z)
call prepot
return
end

```

The SURF subprogram would first call COORD3, then call POT, and then call CHAIN3 before returning to the POLYRATE program. For example:

```

subroutine surf (v, x, dx, n3tm)
implicit double precision (a-h, o-z)
common /potcm/ r(3), energy, dedr(3)
dimension x(n3tm), dx(n3tm)
call coord3(x, n3tm)
call pot
v = energy
call chain3 (x, dx, n3tm)
return
end

```

8.D. Calculations using an electronic structure program through hooks

A very useful feature of the POLYRATE program is its ability to link an electronic structure program to obtain MEP information directly. All energetic information needed by POLYRATE for single-level runs or for the lower level of a dual-level run is obtained in one of two ways. The first way is via a set of general hook routines. The second way is from an electronic structure file (fu29, fu30, fu31, or fu40). This section of the manual describes the hook routines.

The hook routines are actually quite general. They can be used to call an analytical potential surface, and this special case is described in section 8.C. Alternatively they could be used for direct links to electronic structure programs (as in ACESRATE, GAUSSRATE, GAMESOLRATE, MORATE, or AMSOLRATE) or in any way that the user finds useful. In the case of the connection to an electronic structure program, the user is responsible for obtaining or writing the necessary routines that interface the hook routines and the electronic structure program. In some cases, e.g., ACESRATE, GAUSSRATE, GAMESOLRATE, MORATE, and AMSOLRATE such interface routines have been prepared by POLYRATE developers, but users are invited to prepare their own interface routines as well.

We note that historically POLYRATE called SURF and SETUP directly. Beginning with version 7.0, the hooks were inserted as an interface between the POLYRATE program and the source of potential information. The goal is for POLYRATE to be completely blind as to what happens on the other side of this interface. We refer to this as complete modularity between the dynamics and the potential, and we believe that this modularity has been fully accomplished since version 7.3.

The hook routines are PREP, PREPJ, EHOOK, GHOOK, HHOOK, and OHOOK. The source code for these routines is located in the src subdirectory in hooks.f. This subdirectory also contains six utility subprograms (YTRANS, YSECEN, YSECEP, YDERV2, YDER24, and YNEWT). The role of each of the hooks is described below. Before proceeding with these individual descriptions, we first make a few general comments. The basic concept of the hooks is as follows: Whenever POLYRATE needs an energy, it calls EHOOK. Whenever POLYRATE needs a gradient (or an energy and a gradient at the same geometry) it calls GHOOK. Whenever POLYRATE needs a Hessian (or a Hessian plus an energy and/or gradient, all at the same geometry), it calls HHOOK. POLYRATE has its own geometry

optimization routines, and it may employ them using energy, gradient, and/or Hessian information from calls to EHOOK, GHOOK, and HHOOK. However, in many cases it is more efficient to call external optimization routines. These can be accessed by calls to OHOOK. Finally, we note that PREP and PREPJ are available for initialization tasks, i.e., for passing geometry-independent information to the other side of the interface.

The versions of the hooks distributed with POLYRATE itself only call SURF and SETUP, except for HHOOK, which calls GHOOK. This does not convey the full usefulness of the hooks. The hooks are better appreciated by studying one of the direct dynamics codes built on POLYRATE, e.g., GAUSSRATE, MORATE, or AMSOLRATE. But the POLYRATE versions of the hooks provide clear examples of their use that can be understood by all users, independent of whichever electronic structure codes they are familiar with.

Since version 7.3, HHOOK computes Hessians by taking numerical derivatives of gradients obtained by calls to GHOOK. The code is identical to the code for this purpose on the POLYRATE side of the interface. As mentioned in the previous paragraph, this does not convey the full power of this hook. This hook is designed to allow the user to take advantage of analytic Hessians if they are available. If analytical Hessians are available, the user should modify HHOOK to use them. A similar generic version of OHOOK is also provided.

PREP:

PREP is the driver for all preparation routines needed to set up the hooks. This routine is called once after input file fu5 has been read in and before the table of input is written to the output file. This call occurs before any calculations are done. This routine may be used to call electronic structure preparation routines. File fu70 is reserved for use by PREP.

To interface with a new electronic structure program one just needs to add a new block to the 'if block' contained in this subroutine with a call to the appropriate routine. Note that this is an optional hook. Not all electronic structure programs need initialization. When PREP is not needed, one should use a dummy routine with a simple return command inside.

PREPJ:

PREPJ is the species-dependent driver for all preparation routines needed to set up the hooks. Any initializations that are dependent on whether the reactants, products, saddle point, or generalized transition state is computed are done here. PREPJ is called up to six times, namely it is called once each at the starts of the sections for REACT1, REACT2, PROD1, PROD2, START, and PATH. In other words, PREPJ is called before each respective reactant, product, or saddle point is studied and again before the reaction path is computed. A variable, JTYPE, specifies which particular species is being computed; see Section 17 on file usage.

To interface with a new electronic structure program one just needs to add a new block to the 'if block' contained in this subroutine with a call to the appropriate routine. Note that this is an optional hook. Not all electronic structure programs need any species dependent initializations. When the PREPJ is not needed, user should prepare a dummy routine with a simple return command inside.

EHOOK, GHOOK, HHOOK, and OHOOK always communicate with the POLYRATE side of the interface using mass-scaled Cartesians in bohrs and energies in hartrees. In the versions of the hooks distributed with POLYRATE, either mass-scaled Cartesians or unscaled Cartesians may be used for communicating with SURF. In the versions of SURF that are distributed with POLYRATE, the energy is computed using the Cartesian coordinates, so the hooks first convert to unscaled Cartesians before calling SURF. Similar transformations are required for gradients and Hessians; for example, SURF returns gradients in unscaled coordinates, but GHOOK converts them to mass-scaled coordinates.

EHOOK:

EHOOK is the energy interface subprogram of POLYRATE.

To interface with a new electronic structure program, one needs to add a new block to the 'if block' with a call to an interface routine that accepts Cartesian coordinates (unscaled) and returns the energy. If the *read* option is chosen for the EZERO keyword, then the zero of energy should be consistent with the EZERO value specified in the input file. Remember that the user must also specify the new interface potential by adding another option to the POTENTIAL keyword and implementing the new possible value of the associated IPOT variable in the FORTRAN code.

GHOOK:

GHOOK is the gradient (first derivative) interface subprogram of POLYRATE.

To interface with a new electronic structure program, one needs to add a new block to the 'if block' with a call to an interface routine that accepts Cartesian coordinates (unscaled) and returns the energy and first derivatives. The developer of a new interface must also specify the new interface potential by adding another option to the POTENTIAL keyword and implementing the new possible value of the associated IPOT variable.

HHOOK:

HHOOK is the Hessian (second derivative) interface subprogram of POLYRATE.

For the second derivatives, the potential is specified by the HESSCAL keyword of the SECOND section of the fu5 input file and is coded in FORTRAN using the IPOT variable. If the keyword option *ghook* is used in the HESSCAL keyword, numerical difference methods from POLYRATE are applied; if the *hhook* option is used, then the second derivatives are taken from HHOOK. The latter option is designed to allow HHOOK to obtain analytic second derivatives from electronic structure routines. In the current version of POLYRATE, the *ghook* and *hhook* options are essentially the same. If the *hhook* option is used, the second derivatives are computed with the hooks routines YSECEN and YSECEP, whereas if the *ghook* option is used, the second derivatives are computed with POLYRATE subroutines SECCEN and SECCEP. However, the numerical methods used in YSECEN and YSECEP are the same as those in SECCEN and SECCEP. HHOOK must convert the Hessian to mass-scaled coordinates before returning to the POLYRATE side of the interface.

To interface with a new electronic structure program, one needs to remove the subroutine calls in the ISPOT equals to 1 block and add a new block with a call to an interface routine that accepts Cartesian coordinates (unscaled) and returns the energy, first derivatives, and Hessian matrix. Note one must ensure that the gradient and Hessian conform to the same orientation. See subsection 11.A.4 for the description of the keyword options used to calculate second derivatives.

OHOOK:

OHOOK is the geometry optimization interface subprogram of POLYRATE. This routine accepts an initial guess to the mass-scaled Cartesian coordinates of the full system and returns the optimized geometry in mass-scaled Cartesian coordinates and the energy. A variable, IOP, is used to specify which chemical species is being optimized: for IOP = 1, or 2; the chemical species is reactant 1 or 2, for IOP = 3, or 4; the chemical species is product 1 or 2; for IOP = 5, the chemical species is the saddle point.

The geometry optimization is specified by the OPTMIN and OPTTS keywords and is coded in FORTRAN using the IGPO variable. If the keyword option *bfgs* or *nr* is used for either of these keywords, IGPO is 0, and the optimization is performed by NEWT, which is a POLYRATE subroutine. The OHOOK option (IGPO = 1) is designed to allow the OHOOK routine to call an electronic structure package directly. In the generic version of the OHOOK routine, the option calls YNEWT, which is supplied as part of the generic hooks. (Notice that YNEWT is a slightly simplified version of the POLYRATE routine NEWT.) To interface with a new electronic structure program, one must replace the call to YNEWT when IGPO equals to 1 with a call to an interface routine that accepts Cartesian coordinates (unscaled) and returns the optimized geometry and energy. Notice that, in the current version of POLYRATE, the method used to optimize the minima and saddle point should be the same (i.e. either both from POLYRATE or both from electronic structure package). For the description of the keyword options used in geometry optimization, see subsection 11.A.5.

9. CALCULATIONS BASED ON AN ELECTRONIC STRUCTURE INPUT FILE

One major option of the POLYRATE program is the ability to use MEP information obtained from electronic structure calculations in the form of an input file instead of using an analytic potential energy surface. This input file will contain MEP information for a finite range of the reaction path, namely s in the interval [SLM, SLP]. It is the user's responsibility to ensure that these input geometries do indeed lie on the MEP or a good approximation to it.

When one of the electronic structure files (fu30, fu31, or fu40) is used the POLYRATE program is never supposed to make calls to the potential energy surface subprograms SETUP and SURF. However, to satisfy all external references when linking the program the subprograms SETUP and SURF must be provided. A potential energy surface file, dumpot.f, distributed with the POLYRATE program has been designed specifically for use with this option of the code. This potential energy surface file contains dummy SETUP and SURF routines which print a warning message and stop if subprogram SURF is called.

When the reaction-path information is calculated from electronic structure calculations and is used as input for dynamical calculations, the necessary information (listed later in this paragraph) is read from a separate data file which is linked to the FORTRAN unit fu30, fu31, or fu40 and is keyed to the normal input in the file which is linked to the FORTRAN unit fu5. Throughout this manual, unit fu30, unit fu31, and unit fu40 are called the electronic structure input files. The input data in unit fu5 remains the same as when the calculation is done with an analytic potential energy surface except that the POTENTIAL variable keyword in the GENERAL section is set to *unit30*, *unit31*, or *unit40*. A line-by-line description of the MEP input for unit fu30 is provided in Section 11.C, a description of the input when using unit fu31 is given in Section 11.D., and a description of the input when using unit fu40 is provided in Section 11.E. Using the information from unit fu30, unit fu31, and unit fu40, the quantities $V(s)$, $|\mathbf{I}(s)|$, $\omega(s)$, $k_3(s)$, $k_4(s)$, and $B_{mF}(s)$ are interpolated to locations on the save grid and are stored in save grid arrays to be used for Lagrangian interpolation as needed for finding variational transition states and as needed for tunneling calculations. The type of information used to obtain the tables of the MEP quantities is as follows:

Potential

$V_{\text{MEP}}(s)$ is always read directly.

Determinant of the moment of inertia tensor

$|\mathbf{I}(s)|$ is always computed from the geometry of the complex. For reactants and products at infinite separation $|\mathbf{I}(s)|$ is infinite, so for large $|s|$ the program interpolates the inverse of $|\mathbf{I}(s)|$ instead of $|\mathbf{I}(s)|$ itself. To obtain $|\mathbf{I}(s)|$ for finite s it is necessary to read the $3N$ Cartesian coordinates $R(s)$ or the mass-scaled ones $x(s)$.

Gradients

The gradient $\nabla V_{\text{MEP}}(s)$ must be read in. The gradient is denoted DX. The code has options to read either the gradient itself or the negative of the gradient, which is the force. In addition, the gradient or force may be in either unscaled or mass-scaled coordinates.

Frequencies

$\omega(s)$ can be read directly or computed from the force constant matrix $\mathbf{F}(s)$. If $\mathbf{F}(s)$ is read, both $\omega(s)$ and the coefficient matrix \mathbf{L} which diagonalizes \mathbf{F} (or the projected \mathbf{F}) are obtained. A total of $3N-7$ (for a nonlinear generalized transition state), $3N-6$ (for a linear generalized transition state), or $3N-1$ (for solid-state or gas-solid interface generalized transition state) frequencies or a $3N \times 3N$ force constant matrix is read.

Anharmonicities

Either the cubic and quartic force constants $k_3(s)$ and $k_4(s)$ are read directly, or they are omitted. If read, a total of $3N-7$ (nonlinear), $3N-6$ (linear), or $3N-1$ (solid-state or interface) of each will be read. Even if $k_3(s)$ and $k_4(s)$ are not read, anharmonicities can still be included by using the Morse I approximation.

Reaction-path curvature

The reaction-path curvature components $B_{mF}(s)$ can either be read in or they can be computed from the gradients. These curvature components are needed for SCT calculations but are not used for CVT or ZCT calculations.

The user should always supply the following information for all points on the Hessian grid: s , V_{MEP} , the geometry, the gradient, and either the frequencies or the Hessian. Depending on the options chosen, one or more of the following should also be supplied for all points on the Hessian grid: the frequencies, the Hessian (i.e., the matrix of second derivatives of the potential), the cubic force constants, the quartic force constants, and the curvature components. Alternatively, the curvature components can be calculated by the program. For this purpose, when using files fu30 or fu40, the program can use only the gradients at the save points or it can use the gradient at up to two additional s values on either side of each Hessian point. However only the latter option is available when using file fu31.

The user must also supply information about the reactants and the products, if it is needed.

9.A. Relation of POLYRATE input to electronic structure package output

GAUSSIAN94 prints the forces on the standard output file, and it prints the gradients on the formatted check point file; however, on the latter the gradients are incorrectly labeled as forces. We prefer to take the gradients from the formatted checkpoint file because GAUSSIAN's output format on the formatted checkpoint file is more similar to POLYRATE's input format (for fu30, fu31, and fu40), so the user can directly cut and paste.

A different issue arises when one uses the GAMESS electronic structure package. GAMESS writes the gradient in both un-normalized and normalized form. POLYRATE requires the un-normalized gradient if one uses curvilinear coordinates, so the user should be sure to take the gradient from the correct place in the GAMESS output.

9.B. Orientation of the molecule at generalized transition states

When an electronic structure calculation program is used to determine the optimized structures for the generalized transition states, the electronic structure code sometimes translates and/or rotates the molecule to a “standard”, a “Z-matrix”, or other orientation. However, POLYRATE requires precisely the orientation obtained by following the minimum energy path in mass-scaled Cartesians. For example, if the ESD method is used, the points along the path should be related by equations (26) and (27) of the GTST book chapter [Ref. 5 of Section 21]. Using an incorrect orientation can introduce errors in the gradients and curvatures and can result in errors in the tunneling calculations. The user of POLYRATE is urged to be very cautious in understanding these issues if he or she uses an electronic structure package that re-oriens the molecule. Special keywords are sometimes required to prevent re-orientation.

For example, for single-point calculations that ask for a gradient or a Hessian or both, when the GAUSSIAN94 input has the geometry in Cartesian coordinates and the NOSYMM keyword is specified, then GAUSSIAN will not re-orient the molecule, and the gradients and Hessians that it prints out in the formatted checkpoint file (Test.FChk file) are consistent with the orientation of the input geometry. Notice that it is impossible to avoid re-orientation of geometry if the Z-matrix is given in internal coordinates, since GAUSSIAN94 automatically generates a set of Cartesian “Z-matrix” coordinates in which the first atom is placed at the origin, the second one on the Z axis, and the third one in the X-Z plane. This can be avoided using Cartesian coordinates for defining the input geometry. A second aspect of GAUSSIAN94 the user should be aware of is that even when using Cartesian coordinates for the input geometry in single-point calculations, if the NOSYMM keyword is not specified, both the gradient and the force constants matrix written to the *formatted checkpoint file* will correspond to a geometry translated and re-oriented according to what GAUSSIAN94 calls the “standard orientation”. However, notice that the forces and the force constants matrix printed out in the GAUSSIAN94 *long output file* are still consistent with the initial input geometry orientation; thus they can be used for dynamics without introducing errors in the calculation.

Re-orientation issues are more complicated when one considers reaction paths. It appears to be impossible to generate a reaction path with GAUSSIAN94 without having the geometries re-oriented if the transition state geometry required to perform a reaction path calculation is input in internal coordinates. However, if the geometry of the transition

state is given in Cartesian coordinates, and if the computed force constants are also read in Cartesian coordinates from a checkpoint file corresponding to a frequency calculation performed including NOSYMM in the keyword list, the reaction path will have the correct orientation to be used in a direct dynamics calculation. Whether the NOSYMM keyword is or is not used in the actual reaction path calculation does not affect the origin or the orientation of the respective points.

An alternative method that we have used in the past to generate the reaction paths is to employ the electronic structure package GAMESS. However, although GAMESS can be used to generate a non-re-oriented reaction path, we could not find a convenient way to prevent GAMESS from re-orienting when it calculates the gradients and Hessians. (We did find a way to avoid re-orientation, but it is inconvenient because it re-orders the atoms.) One solution is to use GAMESS to generate un-re-oriented geometries along the reaction path, and then use GAUSSIAN (with Cartesian coordinates input and NOSYMM) to generate gradients and Hessians in a formatted checkpoint file. Our test of GAMESS were done with the 9-11-96 version.

9.C. Methods for interpolation of electronic structure data with files fu30 and fu40

This section describes the procedures used with file fu30 when LOPT(2) is positive or with file fu40 when MAXLPTS is positive.

For values of s between those on the input grid, the MEP quantities are obtained by n -point Lagrangian interpolation. The maximum number of points used in the interpolation is input to the program; the actual number of interpolation points is the minimum of this input value [LOPT(2) in unit fu30] and the total number of points in the grid. Thus, if only 3 points are used in the grid, only up to second order Lagrangian interpolation can be used.

For unimolecular reactions, association reactions, or reactions with wells, the reactant, product, or well geometry (or both) will occur at finite values of s . In these cases Lagrangian interpolation is also used for values of s approaching the reactant, product, or well s value.

In runs using analytic potential energy functions, there are two grids: (i) the fine grid, also called gradient grid, with step size $\delta s = \text{SSTEP}$ and (ii) the save grid, where

frequencies and the Hessian are calculated and stored, with step size $\Delta s = \text{INH} * \text{SSTEP}$. See also sections 7, 10.D, and 11.A.7.

In runs using file fu30 or file fu40, there are three grids which are similar to the two grids described above; however their use is somewhat different; also their step sizes are determined differently. First is the gradient grid. This consists of all the s values at which gradient information is read in records 35, 38, or 41 of unit fu30 or the equivalent records of unit fu40. This grid is not required to be evenly spaced, so δs does not exist, and the SSTEP value read from fu5 is ignored. Second is the basic input grid, which is called the sparse grid in Ref. 14 of Section 21. When using file fu30, this is the same as the gradient grid if $\text{LOPT}(6) \leq 2$; it is a subset of the gradient grid if $\text{LOPT}(6)$ is 3 or ≥ 4 . The basic input grid is the grid of points for which one reads energies in record 31 of unit fu30 and either frequencies (record 46) or Hessians (record 44). Third is the save grid. Input electronic structure information and calculated reaction path curvatures are interpolated from the basic input grid plus saddle point information to the save grid by n -point Lagrangian interpolation. The step size for the save grid is $\Delta s = \text{INH} * \text{SSTEP}$. This interpolation can also be done by the IVTST-M method (For unit fu30, $\text{LOPT}(2) = -1$; for unit fu40, the MAXLPTS keyword in *GEN40 is set to -1) instead of by n -point Lagrangian interpolation. When using the IVTST-M method with files fu30 and fu40, the interpolation can only be calculated with the default IVTST-M options.

A comment on the calculation of $B_{mF}(s)$ may be helpful here. When an analytic potential is used, and the default value (*dgrad*) is accepted for the CURV keyword, $B_{mF}(s)$ is calculated by a quadratic fit of points on the gradient grid; this fit is carried out in subroutine BCALC. In this fit, one uses points with $s \leq 0$ to calculate $B_{mF}(s)$ for $s \leq -\delta s$, and one uses points with $s \geq 0$ to calculate $B_{mF}(s)$ for $s \geq +\delta s$. Then one interpolates between $-\delta s$ and $+\delta s$ to get $B_{mF}(s = 0)$. Notice that none of the quadratic fits involves data with both $s > 0$ and $s < 0$. Thus in eq. (98) [see Section 22.A, item 15], $d \left\{ \text{sign}(s) \right\} v_{i\gamma}(s) / ds$ becomes $[\text{sign}(s)] dv_{i\gamma}(s) / ds$. In contrast, when an electronic structure input file is used, files fu30, fu31, or fu40, one calculates $B_{mF}(s)$ by a quadratic fit to points on the whole gradient grid in subroutine RPHB. Since it is now possible for a given 3-point quadratic fit to involve both negative and positive s , the $\text{sign}(s)$ function must be implemented before the fit is carried out because $[\text{sign}(s)] v_{i\gamma}(s)$ is a continuous function, but $v_{i\gamma}(s)$ is not.

9.D. Methods for interpolation of electronic structure data with file fu31

This section concerns the procedures used with file fu31. The fu31 file is currently incompatible with the IVTSTOFREQ option. In version 9.4.1, the ISPE and FREQSCALE options are compatible. The default versions of these procedures may also be used with file fu30 by setting LOPT(2) = -1 or with file fu40 by setting MAXLPTS = -1.

In runs using file fu31 there are also three grids, similar to those described above for files fu30 and fu40, which store input and interpolated information. The gradient and the basic input grids for file fu31 are the same as described above for files fu30 and fu40. However, when using file fu31, the input grid information is always interpolated onto the save grid using IVTST-M instead of by n -point Lagrangian interpolation. After data is interpolated to the save grid, the rest of the calculation is the same as when using file fu30 or fu40 or when using the hooks.

IVTST-M is described in Sections 6.N.2 and 10.J and in detail in Ref. 48 of Sect. 21.

10. SELECTED DETAILS OF THE CALCULATIONS

Most of the methods used by the program are described in detail, including equations, in the GTST book chapter (Ref. 5 in Sect. 21). All methods not discussed in the GTST book chapter are described in the present section of the manual. See also Refs. 14, 29, 30, and 47 in Sect. 21.

Subsections 10.A-10.L describe in detail various options that may be used with either of the two kinds of calculations—those based on an analytic potential energy surface or those based on an electronic structure input file.

10.A. Extrapolation of the reaction-path data beyond the range of the original grid

In some cases the variational transition state lies in the interval [SLM, SLP], which is the range of the save grid, but the tunneling calculation requires information over a wider range of s to converge.

For reactions with a saddle point this is handled almost automatically by the mapped interpolation (MI) algorithm, which allows one to extend the range of the reaction path used without calculating more energies, gradients, or Hessians. For reactions with a saddle point the MI algorithm is available to interpolate the energy, the determinant of the moment of inertia, the frequencies, and the curvature components between SLM and the reactants or SLP and the products. This may be thought of as an interpolation between SLM and reactants and an interpolation between SLP and products, or it may be thought of as an extrapolation beyond SLM toward reactants and beyond SLP toward products. (That is why we sometimes call it interpolation and sometimes call it extrapolation.) The end result is that all information is interpolated over the entire range from reactants to transition state to products. The keywords EXFIRST and EXSECOND in the PATH section control the options for the interpolation. If the EXFIRST and EXSECOND keywords are not specified then no interpolation is performed. The MI algorithm is explained in Sections 6.N.2, 6.N.3, 10.J, and 11.A.7 and in Ref. 48 of Sect. 21.

For reactions without a saddle point POLYRATE includes a simple two-point extrapolation to extrapolate the vibrationally adiabatic ground-state potential and effective reduced

mass, μ_{eff} , from the MEP end points to the reactants and/or products. *The asymptotic extrapolation procedures that may be used for reactions without a saddle point are described in this subsection.*

When using units fu30 and fu40 the two-point extrapolation is used by default to extrapolate between the last save grid point and SLM or SLP. However, the IVTST-M algorithm can be specified to interpolate between the last save grid point and SLM or SLP by setting LOPT(2) = -1 for unit fu30 and by setting MAXLPTS = -1 for unit fu40. File fu31 always uses the IVTST-M algorithm to interpolate from the last save grid point to SLM or SLP. See Sections 6.N.2 and 10.J and Ref. 48 of Sect. 21 for more details about the IVTST-M algorithm. In all cases, when using the electronic structure file input options (units fu30, fu31, or fu40), the extrapolation or interpolation is applied using data on the save grid, not the basic input grid (i.e., not the sparse grid). The extrapolation or interpolation is specified by increasing the range of SLM and SLP in the path section.

Precise calculation of the transmission coefficients through the adiabatic barrier for reactions involving two reactant or product species often requires calculation of the ground-state (G) adiabatic energy $V_a^G(s)$ at large absolute values of the reaction coordinate. But calculation of the MEP at large $|s|$ is often impractical, especially in the asymptotic regions near reactants or products, where a very small gradient step size is often required for good accuracy because of the small magnitude of the gradient. To alleviate this problem, the program contains an option to calculate $V_a^G(s)$ and $\mu_{\text{eff}}(s)$ at large $|s|$ by a two-point extrapolation method, controlled by the EXFIRST and EXSECOND keywords in the PATH Section of the fu5 input when the hooks are used and the NOSADDLE keyword is specified in the PATH section. This two-point extrapolation can only be used for reactions without a saddle point or with electronic structure input files fu30 and fu40. See the above discussion for more details. Here we discuss the method in terms of $V_a^G(s)$, but the functional form is the same for $\mu_{\text{eff}}(s)$. [A similar extrapolation is also made for $V_{\text{MEP}}(s)$ but this is for informational purposes only, e.g., it may be useful for plotting.]

The method used for bimolecular reactions when the calculation is based on a potential energy surface involves a simple exponential function which is fit to the values of $V_a^G(s)$ at the positive ($s = \text{SLP}$) or negative ($s = \text{SLM}$) limits of the reaction coordinate for

which $V_a^G(s)$ has been accurately determined and to the adiabatic ground-state energies of the products, $V_a^G(s = +\infty)$ or reactants, $V_a^G(s = -\infty)$. Thus

$$V_a^G(s) = V_a^G(s = +\infty) + \left[V_a^G(\text{SLP}) - V_a^G(s = +\infty) \right] e^{-\alpha(s - \text{SLP})}$$

for extrapolation towards products ($s > \text{SLP}$), and

$$V_a^G(s) = V_a^G(s = -\infty) + \left[V_a^G(\text{SLM}) - V_a^G(s = -\infty) \right] e^{\alpha(s - \text{SLM})}$$

for extrapolation towards reactants ($s < \text{SLM}$). The exponential range parameter α is supplied by the user, or the default (1.0 a_0^{-1}) can be used. Different values of α can be used in the same calculation for extrapolations towards products and reactants.

If the keyword EXFIRST and EXSECOND are not specified, or EXNSTP is zero, no extrapolation is performed. This is equivalent to using $\alpha = \infty$, i.e., this is equivalent to setting $V_a^G(s)$ equal to its value at $s = \text{SLP}$ for all $s > \text{SLP}$ and to its value at $s = \text{SLM}$ for all $s < \text{SLM}$.

As far as inputting a value of α , the user may use any method he or she likes to get α . One suggestion for guesstimating α is a 2-point fit of the exponential on to the end of the Hessian grid data: Let s_x be a point near SLP (s_x could be the second-to-last point, or slightly farther back if precision of $V_a^G(s)$ info is lacking). Then require exponential to fit $V_a^G(s = s_x)$ and $V_a^G(s = \text{SLP})$, which gives

$$\alpha = \frac{\ln \left[\frac{V_a^G(s = s_x) - V_a^G(s = +\infty)}{V_a^G(s = \text{SLP}) - V_a^G(s = +\infty)} \right]}{(\text{SLP} - s_x)}$$

However, if the value chosen for α affects the results, then this represents the basic uncertainty in the values; a wider range for $V_a^G(s)$ will be needed to obtain a better estimation of α .

10.B. Options for anharmonicity

In addition to the standard harmonic option, several anharmonic options are available using the keywords TOR, MORSE, MORSEQQ, MORMODEL, QQWKB, QQSEMI, and WKB. Anharmonicity may be included in reactants, products, or generalized transition states, but only in conjunction with the independent-normal-mode (INM) approximation in rectilinear coordinates. Two of these options, Morse I and Morse I quadratic-quartic (MORSE and MORSEQQ respectively with the MORMODEL variable keyword set to *morsei*), are described in the GTST book chapter (see pp. 87-92 of Ref. 5 in Sect. 21) and were first used in the calculations reported in Ref. 1 of Sect. 21. The options defined by the QQSEMI and QQWKB keywords are two WKB options that will be discussed in Subsection 10.B.1. The VARIABLE and VRANGE keyword options allow a different anharmonicity method to be used for each mode in each region of the MEP. When any of the above anharmonicity options are used, anharmonicity is computed for all vibrational levels of a given mode by the same method. An additional anharmonic option, specified by the WKB keyword, allows one to use a presumably more accurate, but more time consuming method (the WKB method based on the true potential along each generalized normal mode) for the ground-state energy level of each mode for modes treated thermally, or for the specified energy level for state-selected modes. When this option is chosen, the methods used for higher-energy states of thermal modes are still determined by an additional anharmonicity keyword. If another anharmonicity keyword is not included the higher-energy states will be determined using the harmonic approximation. Details of the implementation of the WKB keyword option will be discussed in Subsection 10.B.2. All WKB anharmonicity options are available only for calculations based on the hooks. Whenever the WKB approximation is used, it is employed in rectilinear coordinates, even if internal coordinates are used for the frequencies and for non-WKB energy levels in the partition functions.

10.B.1. WKB anharmonicity based on a quadratic-quartic fit

When either QQWKB or QQSEMI keywords is specified, the program uses the primitive or uniform WKB method, respectively, to compute the energy levels of a given generalized normal mode vibration from a quadratic-quartic potential fit to the true potential. The quadratic-quartic potential is determined by fitting to the potential computed at two distances, specified by the user in the mass-scaled coordinate system, along the

generalized normal mode. The energy levels of this potential can be computed by either a primitive WKB method [Ref. 21, Sect. 20] (QQWKB) or, for double well potentials, by a semiclassical method using parabolic connection formulas [Ref. 22, Sect 20] (QQSEMI). In these methods the phase integrals are calculated analytically. The calculations are described in Ref. 17 of Sect. 21.

10.B.2. WKB anharmonicity based on the true potential

When the WKB keyword is specified, the program uses the WKB method to compute the anharmonic ground- or first-excited-state energy levels of all generalized normal mode vibrations from the true potential along the modes. The method is described in more detail in Ref. 13 of Sect. 21. Note that WKB energy levels are not computed for reactant and product diatomic species, but instead either harmonic or anharmonic energy levels (as specified by the other anharmonicity options) are used.

Because the numerical solution of WKB eigenvalues requires repeated calculation of the potential and may therefore be quite time consuming, this approach is used only for the zero point ($n = 0$) eigenvalues for thermal modes or the specified energy level for state-selected modes. Higher energy vibrational levels, used in the calculation of the partition functions, are obtained by harmonic or anharmonic approximations discussed previously. In particular, when the WKB-with-the-true-potential option is applied to evaluate the ground-state energy levels, the vibrational partition function of a bound motion is calculated as follows:

$$Q_{\text{vib},m} = e^{-\varepsilon_{0,m}^{\text{WKB}}/kT} + \sum_{n=1}^{\infty} e^{-(n+1/2)\hbar\omega_m/kT}$$

$$Q_{\text{vib},m} = Q_{\text{vib},m}^{\text{HO}} - e^{-\hbar\omega_m/2kT} + e^{-\varepsilon_{0,m}^{\text{WKB}}/kT}$$

where $\varepsilon_{0,m}^{\text{WKB}}$ is the WKB ground-state energy of mode m , and $Q_{\text{vib},m}^{\text{HO}}$ is the vibrational partition function in the harmonic oscillator approximation. If the harmonic frequency is imaginary, the partition function is set to 10^{10} . Note that in the WKB method the turning points are determined during the calculation of the vibrational energies, and they can be used directly in the calculation of the effective reduced mass as a function of s .

This method is used in the test runs oh3tr3, ohcltr2, ohcltr3, and ohcltr4.

10.B.3. Hindered rotor treatment

All of the statistical thermochemical treatment of torsions treated as hindered internal rotations used in Polyrate 9.4.1 are explained in Ref. 58. This reference explains the methods in detail, and also gives the references to the original methods. Beginning with version 8.2 of POLYRATE, a torsional mode can be treated in the full generality of Ref. 52. Please note the errata that have been added to Ref. 52.

If mode m is treated as a hindered rotor (HR), its partition function tends to the limit of a harmonic oscillator (Q_m^{HO}) at low temperature and goes to the limit of a free rotor (Q_m^{FR}) at high temperature, and it remains reasonably accurate for the hindered-rotor regime between these limiting cases. In particular, a simple approximation has been used:

$$Q_m^{\text{HR}} = Q_m^{\text{HO}} \tanh\left(\frac{Q_m^{\text{FR}}}{Q_m^{\text{I}}}\right),$$

where the harmonic oscillator approximation for mode m is Q_m^{HO} , the free rotor approximation for mode m is Q_m^{FR} , and Q_m^{I} is an intermediate result. Another option is the torsional Pitzer–Gwinn method based on a reference potential:

$$Q_{\text{tor}}^{\text{RPG}} = \exp(-\beta\hbar\bar{\omega}/2) \left[\frac{\tilde{Q}_{\text{tor}}^{\text{HO}}(\bar{\omega})}{Q_{\text{tor}}^{\text{CHO}}(\bar{\omega})} \right] Q_{\text{FR}} \exp(-\beta W/2) I_0(\beta W/2)$$

Where $\bar{\omega}$ is the frequency at the minimum of the reference potential, W is the barrier height of internal rotation, and I_0 is a modified Bessel function. The reader is referred to Ref. 58 for all further details.

The symmetry number σ_{jm} for minimum j in the torsional potential for mode m should not be confused with the symmetry number σ for overall rotation.

The test runs cwmctr3 and cwmctr2 are examples of using the hindered rotor option.

10.C. Alternate method for the calculation of numerical second derivatives

In order to calculate the Hessian matrix, the second derivatives of the potential with respect to the Cartesian coordinates are computed numerically, using the analytic first derivatives supplied by the analytic potential energy surface subprogram. By default, the method of central differences is used to compute these second derivatives. The formula is given by equation (16) in the GTST book chapter (Ref. 5 of Sect. 21).

Two alternate options, selected by setting the NUMTYP variable keyword to *quadratic* or *fourth*, were added for use in cases where the default method may lead to numerical instability. In the most accurate option, NUMTYP = *fourth*, the second derivatives are computed by fitting the analytic first derivatives to a fourth order polynomial. For NUMTYP = *quadratic*, the second derivatives are computed from a quadratic fit of the potential.

The parameters NUMSTEP and DLX2, which are specified by the variable keywords NUMSTEP and DLX2 respectively, in the input file which is linked to the FORTRAN unit fu5, determine the step size for calculating second derivatives of the potential, and, if needed, for cubic and quartic force constants as well. The second derivatives are needed for two purposes: locating stationary points and computing vibrational potentials.

The DLX2 parameter is used for locating the stationary point, and the NUMSTEP parameter is used for calculating the force constants. When anharmonicity options are chosen that require cubic and quartic force constants, which are third and fourth derivatives of the potential, these are evaluated as numerical second and third derivatives, respectively, of the analytic gradient. The step size used for these numerical derivatives is always NUMSTEP.

10.D. Reaction-path following methods

Our recommendations, as described in Ref. 27 of Sect. 21, for the most appropriate methods for calculating the MEP and the reaction-path information required for variational transition state theory calculations with semiclassical transmission coefficients are as follows: (i) the Page-McIver algorithm, with the Hessian recalculated at every step, and with a cubic starting step, (ii) the ES1* algorithm (the ES1 method with DELTA2 =

SSTEP and DIFFD = 0.01) with a Hessian step size three times larger than the gradient step size, and with a quadratic starting step, and (iii) the Euler steepest-descents (ESD) algorithm with a Hessian step size nine times larger than the gradient step size, and with a quadratic starting step. The RPM keyword in the PATH section allows the user to choose between the above possibilities or to choose other methods for computing the MEP. A list of all methods available in this version is as follows:

The RPM keyword is set to *esd* for the ESD method. This method is discussed further in Ref. 18 of Sect. 21.

The RPM keyword is set *es1* for the ES1 method of Ref. 18 of Sect. 21. This is an improved version of the methods proposed by Ishida et al. [Ref. 19, Sect. 20] and by Schmidt, Gordon, and Dupuis [Ref. 20, Sect. 20]. The ES1 method allows for a user-selected criterion for whether or not a stabilization is carried out.

The RPM keyword is set to *pagem* for the Page-McIver method, which is called the local quadratic approximation (LQA) in the original reference [Ref. 17, Sect. 20], for calculating the MEP.

The RPM keyword is set to *vrpe* for the VRP method based on the Euler steepest-descents algorithm for the zero-order step, which is called VRP(ESD). This method is discussed further in Ref 49 of Sect. 21.

The ESD, ES1, and Page-McIver methods can be used with the RODS algorithm [Ref. 47, Sect. 21], to reorient the generalized-transition-state-theory dividing surface. This is done by adding the RODS keyword in the PATH section of file fu5. The RODS algorithm is not recommended for use with the VRP(ESD) algorithm since the generalized-transition-state-theory dividing surface has already been optimized by the VRP(ESD) algorithm.

Each of the above methods can be used with either a quadratic first step (i.e., a first step along the imaginary-frequency normal mode) or a cubic starting algorithm in which the first step off the saddle point is based on a cubic expansion of the potential energy surface using the imaginary-frequency normal mode eigenvector and the curvature vector. This is specified by the FIRSTSTEP variable keyword also in the PATH section. For a quadratic first step set FIRSTSTEP = *nmode* and for a cubic first step set the variable to *cubic*.

A discussion of the above methods and useful hints for choosing the most appropriate one, especially for direct dynamics calculations, is given in Ref. 27 of Sect. 21.

We use the following conventions for the step size of the save grid. The save grid is defined as the set of points where the frequencies are stored for straight VTST runs, these are also the points where the Hessian matrix is calculated and where generalized transition-state normal mode analyses are performed and saved. The INH keyword determines how many gradient steps are taken before a Hessian is calculated or read in. In other words, the step size of the save grid is required to be an integer multiple, INH, of the gradient grid step size. Hence, the save grid step size = INH \times SSTEP.

There is another option, which is specified by the SFIRST keyword, in which the first NFSTEP steps are taken with a different gradient step size FSIZE. Then the gradient step size, SSTEP, is used for the remaining steps.

10.E. Interpolation of small-curvature effective reduced mass

For the small-curvature tunneling calculations (SCT), the effective reduced mass is needed. The default method for computing the effective reduced mass from the values stored on the Hessian grid, which includes $s = 0$, is spline interpolation. An alternate method, selected using the LAGRANGE keyword within the SCTOPT list of the TUNNEL section of the input is Lagrangian interpolation of order n . This new method is recommended when using medium-to-large Hessian grid step sizes for which spline fits tend to oscillate. The step size used for either of the interpolation methods is INH \times SSTEP specified by the user with the SSTEP and INH keyword in the input data file. The alternate method is used in the sample runs described in Subsections 14.C.3, 14.C.9, and 14.C.29.

10.F. Reaction-path curvature calculation

See Section 6.H for a brief introduction to reaction-path curvature.

When the CURV variable keyword in the PATH section of the input is set to *oneside* or *dgrad*, the calculation of the component $B_{mF}(s)$ of the reaction-path curvature along the direction of mode m at a point s on the reaction path Hessian grid, is carried out by eq.

(98) of the GTST book chapter. (This is equivalent to the second form of eq. (47) of [Ref. 17, Sect. 20].) This method requires the computation of the derivative of the gradient of the potential with respect to the reaction coordinate s (see pages 96-98 in the GTST book chapter). In early versions of the polyatomic VTST code this derivative was computed using one-sided forward differences; see eq. (105) of the GTST book chapter. An option, `CURV = dgrad` has been added to use a more accurate method, namely, central differences (which is equivalent to a 3-point quadratic fit). When this new option was added the one-sided difference method was changed to one-sided backward differences to allow for a more consistent set of algorithms. It is recommended, however, that the method of backward differences, `CURV = oneside`, never be used. The step size used for these numerical derivatives is that of the gradient grid, `SSTEP`.

Another option, `CURV = dhess`, calculates the reaction-path curvature components using eq. (48) of Page and McIver [Ref. 17, Sect. 20]. This method does not require the derivative of the gradient. A sample run with this option is discussed in Subsection 14.C.4. Preliminary tests indicated somewhat higher accuracy with `dgrad`.

For any of these choices of `CURV`, the code does not compute the curvature at $s = 0$ directly. Instead, the curvature at $s = 0$ is computed linear interpolation as described by the `INTMU` keyword in the `PATH` section.

10.G. Solid-state and gas-solid interface reactions

A solid-state reaction, which includes gas-solid interface reactions as a special case, is a reaction in which N primary atoms are considered to move in the presence of a fixed solid. This is called the embedded cluster model; see Ref. 9 of Sect. 21. For solid-state reactions, the `SPECIES` should be `ssrp` or `ssts` as explained in section 6.B.

10.H. VTST-IOC calculation, including ICR, ICL, and ICA schemes

The original VTST-IOC method is the VTST-IOC method that is described in Ref. 36 of Sect. 21, and the modified version that was the default in POLYRATE from version 7.2 – 7.9.2 is described in Ref. 44 of Sect. 21. In this manual we use the phrase VTST-IC to refer to any scheme involving interpolated corrections from a higher level and VTST-

IOC to refer to any VTST-IC scheme in which the data for corrections involves a higher-level optimization. Thus VTST-IC and VTST-IOC can involve not only the original ICA scheme of Ref. 36 of Sect. 21 for interpolating corrections to frequencies but also to the ICR and ICL options described in Ref. 44 of Sect. 21 and also described toward the end of this section. To use any IOC method, the user has to prepare an input file (unit fu50; see Subsection 11.F) which contains the necessary higher-level information. The DL keyword must be set equal to *ioc* and/or the ICOPT keyword must also be given in the GENERAL section of the unit fu5 input file.

If the MEP is finite (unimolecular reaction with a single product) or semi-infinite (unimolecular reaction with two products or bimolecular reaction with a single product) or has wells (i.e., precursor and/or successor complexes) that the user wishes to use in the interpolation (i.e., if MEPTYPEP or MEPTYPER is (or both are) set to *well*), the user has to tell the program where the MEP ends and/or some properties of the complexes corresponding to the wells (see Subsection 11.F).

The VTST-IOC method demands an explicit matching of the frequencies, i.e., the user must indicate which mode at the saddle point corresponds to each mode at the other two stationary points involved in the interpolation (or else accept the default as explained later in this paragraph). For a bimolecular reaction with two products (MEPTYPER and MEPTYPEP are both set to *two*), the recommended method is to let the program match the frequencies in descending order (which is the default in the current version), i.e., the highest frequency at the saddle point is matched to the highest frequency of either of the two asymptotic species, and so on. If the user knows (by symmetry and/or by inspecting the uncorrected MEP output) that the matching should not be done in descending order, then the user can use the FREQMAT keyword to input the matching order manually in unit fu50. When MEPTYPER and MEPTYPEP are not both set to *two*, one of the normal modes at the reactants and/or products turns into the reaction coordinate motion, and the user has to specify the frequency matching explicitly, so the FREQMAT keyword must be used.

If a well is used for interpolation on the product side of the MEP, and the user is not interested in the reverse rate constant, then the accurate product information is not required. The user can use the NOPRODFREQ keyword to tell the program that no correction for the product frequencies is needed. Otherwise the user has to use the PRODFREQ keyword to input the corrected frequencies for the product in unit fu50.

The original IOC method involved the ICA option, which corrects all the frequencies, and sometimes it will generate a negative correction that changes a small real frequency into a negative frequency (not an imaginary frequency). In order to prevent this unphysical result, one may use IVTST-0 for one or more low frequencies, or, preferably, one may use one of two new interpolated correction options, namely the ICL (Interpolated Correction based on Logarithm of ratios) and the ICR (Interpolated Correction based on Ratios) options.

We note that all three IOC interpolation options for frequencies correct the lower-level frequencies only when they are real. If the lower-level frequencies are imaginary, the program does nothing, so the frequencies remain imaginary. (The recommended way to handle this situation is to use the LOWFREQ option in the fu50 file. This option treats selected low-frequency vibrations by IVTST-0, while carrying out the rest of the calculation by the VTST-IOC method.)

The differences between the original ICA option and the ICL and ICR options are best illustrated by equations. Let $\omega_m^{\text{LL}}(s)$, $m = 1, 2, \dots, F-1$ (where $F = 3N - 6$ for an N -atom nonlinear generalized transition state) denote the lower-level frequencies of the generalized transition state at s . In the original IC method, which is called “interpolated corrections–additive” (ICA), the dual-level (DL) frequencies are given by

$$\omega_m^{\text{DL}}(s) = \omega_m^{\text{LL}}(s) + f^{\text{ICA}}(s)$$

where

$$f^{\text{ICA}}(s) = \text{interpolant} \left\{ \omega_m^{\text{HL}}(s) - \omega_m^{\text{LL}}(s) \right\}_{s=-\infty, 0, \infty}.$$

The “interpolated corrections based on the ratio” (ICR) and “interpolated corrections based on the logarithm” (ICL) methods are defined by

$$\omega_m^{\text{DL}}(s) = \omega_m^{\text{LL}}(s) f^{\text{ICR}}(s)$$

where

$$f^{\text{ICR}}(s) = \text{interpolant} \left\{ \omega_m^{\text{HL}}(s) / \omega_m^{\text{LL}}(s) \right\}_{s=-\infty, 0, \infty}$$

and

$$\omega_m^{\text{DL}}(s) = \omega_m^{\text{LL}}(s) \exp \left[f^{\text{ICL}}(s) \right]$$

where

$$f^{\text{ICL}}(s) = \text{interpolant} \left\{ \ln \left[\omega_m^{\text{HL}}(s) / \omega_m^{\text{LL}}(s) \right] \right\}_{s=-\infty, 0, \infty}.$$

There are also options for interpolating $V_{\text{MEP}}(s)$. The original option, called SECKART, is described in Ref. 36 of Sect. 21. In the DECKART option, the dual-level $V_{\text{MEP}}(s)$ is defined as:

$$V_{\text{MEP}}^{\text{DL}}(s) = V_{\text{MEP}}^{\text{LL}}(s) + \left[V_{\text{Eck}}^{\text{HL}}(s) - V_{\text{Eck}}^{\text{LL}}(s) \right]$$

where

$$V_{\text{Eck}}(s) = \frac{AY}{1+Y} + \frac{BY}{(1+Y)^2} + C$$

$$Y = \exp\left(\frac{s - S_0}{L}\right)$$

$$A = V_{\text{MEP}}(s = +\infty) - V_{\text{MEP}}(s = -\infty)$$

$$C = V_{\text{MEP}}(s = -\infty)$$

$$B = (2V^\ddagger - A - 2C) \pm \sqrt{(V^\ddagger - C)(V^\ddagger - A - C)};$$

we choose

$$S_0 = -L \ln\left(\frac{A+B}{B-A}\right)$$

so that the maximum of $V_{\text{Eck}}(s)$ occurs at $s = 0$. V^\ddagger is the classical barrier height, and L is the range parameter. In $V_{\text{Eck}}^{\text{HL}}(s)$, L is determined from the imaginary frequency of the higher-level calculation by the equation

$$L^{\text{HL}} = \sqrt{\frac{2V^\ddagger(V^\ddagger - A)}{\mu(\omega^\ddagger)^2 B}}$$

where μ is the scaling mass (normally set to 1.0 amu), and ω^\ddagger is the imaginary frequency at the higher level. In $V_{\text{Eck}}^{\text{LL}}(s)$, L is determined from a four-point fit to $V_{\text{MEP}}(s)$ at reactants, products, saddle point, and one extra point as discussed in Refs. 36 and 44 in Sect. 21.

Versions 7.2 – 7.9.2 used the so-called improved scheme (i.e. *ic1* for vibrations and *deckart* for $V_{\text{MEP}}(s)$) as the default VTST-IOC method. Beginning with version 8.0 the defaults are *ic1* and *seckart*. To perform the calculation by the original IOC method, a combination of *ica* and *seckart* was required. The Eckart function fit to 3 stationary points and ω^\ddagger at the higher level can only be used for bimolecular reactions.

The following table summarizes the IOC options for interpolating corrections to vibrations an to the V_{MEP} curve:

Option no.	Keywords
1	<i>ica, seckart</i>
2	<i>icl, deckart</i>
3	<i>icl, seckart</i>
4	<i>icr, seckart</i>
5	<i>ica, deckart</i>
6	<i>icr, deckart</i>

Option 1 was the original default, option 2 was the recommended choice and default in versions 7.3 – 7.9.2, and option 3 is the current recommended choice and default.

In order to perform VTST-IOC calculations using any of the *ica*, *icl*, or *icr* choices for interpolating vibrational information, we suggest the following procedure for a bimolecular reaction:

Step 1. Calculate the rate constants at the lower level. At the end of the reaction path section in the long output file (i.e. before the reaction rate calculation), the range parameter L is calculated automatically from a four-point fit. In the same section, an L value obtained from the imaginary frequency at the lower level is also calculated, but this is only for comparison.

Step 2. Input the range parameter L calculated from the four-point fit as the value for the RPL keyword in the fu50 input file. Or calculate the range parameter L for the RPL keyword in the fu50 input file with the findl.f program in /util. (See Sect. 15.D.) This utility program asks the user to input the V_{MEP} values at reactants, saddle point, and products (0, barrier height, endoergicity) and the (s , V_{MEP}) values of an extra point which lies somewhere between $s = -\infty$ and $s = 0$ (between the reactants and saddle point).

Step 3. Use the RESTART keyword to calculate the rate constants from the interpolated corrections with the higher-level information given in fu50. (Note: The RESTART option is not working properly in this version, you must do the lower-level calculations again if you are dealing with an analytic surface.)

For a unimolecular reaction, the B parameter can be calculated from the findb.f program in /util.

10.I. VTST-ISPE calculation

The VTST-ISPE method for calculating reaction rates is a dual-level approach. ISPE denotes that the higher-level information consists of interpolated single-point energies. The fully converged reaction path is calculated at a lower level of theory Y, e.g. MP2/6-31G, and single-point energies are calculated at a higher level of theory X, e.g. CCSD(T)/cc-pVDZ, where the geometry for the single-point calculation is taken from level Y, denoted X//Y. In order to circumvent the limitation of the VTST-IOC formulas that are designed for calculations in which only corrections at stationary points are used, the VTST-ISPE calculations use the mapped interpolation formulas rather than the VTST-IOC formulas. However, whereas the IVTST-M method requires at least one nonstationary point on both sides of the saddle point, VTST-ISPE can handle the case where corrected energies are provided only at one nonstationary point on either side of the saddle point.

To calculate the reaction rates with the VTST-ISPE algorithm, one should calculate a converged reaction path at the lower level Y. Then a spline under tension is used to interpolate the energy difference of a few single-point level-X energies along the level-Y MEP and the energy from the lower level Y (*i.e.*, $V^{X//Y} - V^Y$). As in our previous interpolated variational transition state theory with mapping (IVTST-M) method, a mapping function is used to provide a systematic interpolating procedure for both bimolecular and the unimolecular reaction. Like IVTST-M, which does not support LCT calculations at this time, VTST-ISPE supports only ZCT and SCT tunneling options at this time.

To map the reaction coordinate s into the z space, the following expression is used:

$$z = \frac{2}{\pi} \arctan\left(\frac{s - s_0}{L}\right) \quad (1)$$

Information from the lower level is used to obtain the parameters s_0 and L for the mapping. First we calculate

$$s_A^0 = -\sqrt{\frac{V_{\text{MEP}}(s=0) - V_{\text{MEP}}(s_X)}{|\omega^\ddagger|^2 \mu}} \quad ; X = \text{R, RW} \quad (2)$$

and

$$s_B^0 = \sqrt{\frac{V_{\text{MEP}}(s=0) - V_{\text{MEP}}(s_X)}{|\omega^\ddagger|^2 \mu}} \quad ; X = \text{P, PW} \quad (3)$$

where μ is the scaling mass, ω^\ddagger is the imaginary frequency of the saddle point at the lower level, and R, RW, P, and PW denote the reactants, the reactant well, the products, and the product well at the lower level. Then we calculate

$$s_A = -\min(|s_A^0|, 2s_B^0) \quad (4)$$

and

$$s_B = -\min(2|s_A^0|, s_B^0) \quad (5)$$

Finally, the parameters s_0 and L are obtained as:

$$s_0 = \frac{s_A + s_B}{2} \quad (6)$$

and

$$L = \frac{-|s_A| + s_B}{2} \quad (7)$$

The parameters s_0 and L obtained from equations (5) and (6) are used in equation (1) to map the reaction coordinate s into z . For a unimolecular reaction with a unimolecular product ($A \rightarrow B$), where the reaction coordinate s occupies a finite interval, this maps a finite interval in s into a finite interval in z . For a bimolecular reaction with bimolecular products ($A + B \rightarrow C + D$), where the s variable occupies interval $[-\infty, \infty]$, the mapped z values are located in the finite interval $[-1, 1]$. For $A \rightarrow B + C$ or $A + B \rightarrow C$, equation (1) maps a semi-infinite interval of s to a finite interval of z . In all cases, a spline fit to the differences of the classical energies is applied with this finite range of z . Therefore, we can write the interpolated energy along the reaction path at the dual level as:

$$V_{DL} = V_{LL} + \text{spline}(\Delta V, z) \quad (8)$$

where DL means dual level (VTST-ISPE), and LL means the lower level. ΔV is the difference of the classical energies between higher-level and lower-level at a particular s value.

In the current version of POLYRATE, ISPE calculations may be used with the ZCT and SCT tunneling methods but not with LCT or μ OMT.

To carry out a VTST-ISPE calculation, an external file fu51 is used to input the values of the higher-level energies. Please refer to Section 11.G for a description of fu51. The VTST-ISPE method can be applied with higher-level calculations at only the reactants, products, and saddle point, or it may be applied with more points. In the latter case, extra points are used to correct the energies along the MEP, the POINT sections are required for the extra points.

The interpolation can be carried with or without the wells along the MEP. It is recommended to use the well option if there is a complex along the MEP in order to obtain the best interpolation. For bimolecular reactions, the energies of the saddle point, wells, products, and extra points that are input via fu50 should be with respect to the sum of the energies of the reactants at the higher-level (which is set to zero). There are two test runs using the VTST-ISPE method, the ch5fu51tr1 test run which is a bimolecular reaction with the J1 surface, and the cmctr3 test run, which is a unimolecular reaction with two wells. For both reactions, we use the two turning points at the representative energy of the SCT calculation at 300 K as the extra points, and we evaluate the single-point energies with the MP2/6-31G** method.

The VTST-ISPE method can be applied for a minimum of three high-level points, namely the reactants, saddle point, and products. The calculations are performed using unit fu51 with the stationary points being required in the ISPEGEN section of unit fu51, and the POINT section being included if and only if additional points along the reaction path are added. A POINT section is needed for each additional high-level point along the reaction path.

See Ref. 51 of Sect. 21 for more discussion.

10.J. Alternative methods for mapped interpolation

The mapped interpolation (MI) methods for IVTST-M calculations and for extrapolation beyond SLM and SLP are described in Ref. 48 of Sec. 21. In addition to the standard methods described in that paper, POLYRATE contains a few variants that may be used to test sensitivity to the approximations made in interpolation.

The first variant concerns the interpolation of $V_{\text{MEP}}(s)$ when there is only one reactant or product or when interpolation is based on a well. The default is to use a cubic polynomial in s to fill in 10 points between the end of the grid and the finite-distance stationary point. Variant 1, with the *nopolys* option for the IVTSTMOPT keyword in the PATH section of the fu5 input file, replaces the 10 points that in the standard algorithm are calculated from a local cubic polynomial by a single point located a small distance (δs) from the reactant (or product or well), whose energy is approximated using the lowest

frequency at the stationary point. Thus the spline under tension is fit to $14 + m$ or $5 + m$ points rather than $23 + m$ points. (See Ref. 48 for the definitions of m and other details.)

The second variant allowed in MI calculations is an alternative way to calculate the effective reduced mass for CD-SCSAG calculations. The standard way involves interpolating each curvature component. The alternative method, keyword INTMUEFF in the GENERAL section of the fu31 input file, involves interpolating the argument of the exponential function in eq. (16) of Ref. 48 instead of interpolating curvature components. When the INTMUEFF keyword is used in conjunction with a VTST-IOC calculation, no higher-level correction to $\mu^{\text{SC}}(s)$ will be made. The INTMUEFF keyword is only available when using unit fu31.

10.K. Quantized-reactant-state tunneling calculation

For unimolecular reactions at low temperatures, most of the reactant molecules have their reaction-coordinate generalized vibrational modes in the ground state. Thus the semiclassical treatment, which assumes continuous tunneling energies, is inappropriate. Starting in version 6.2, we have included a new method that performs the tunneling calculations only at the energies corresponding to the eigenstates of the reaction-coordinate mode of the reactant for the case of a unimolecular reaction. (Note that all other modes are still assumed to be in their ground states.) The reaction-coordinate eigenstates can be determined using the harmonic or WKB approximation. Details of the theoretical background and an application to the diffusion of hydrogen atoms on a Ni surface can be found in Ref. 37 in Sect. 21. The keywords used to perform this type of tunneling calculation can be specified in the TUNNEL section in the unit fu5 input file. See Section 11.A.8 under the keyword QRST for details.

10.L. Curvilinear generalized normal mode analysis

The generalized normal mode analysis that is performed at each Hessian grid point (see Section 9.C) on the MEP may be carried out either in Cartesian coordinates or in internal coordinates. The COORD variable keyword in the PATH section makes it possible to select the type of coordinates to be used. The default option for the COORD variable keyword is *cart* which specifies Cartesian coordinates. Two options are available to select a generalized normal mode analysis in nonredundant curvilinear coordinates:

curv1 and *curv2*. If either of the curvilinear internal coordinate options is selected for COORD, the Cartesian gradient vector \mathbf{G} and force constant matrix \mathbf{F} at the save point are first transformed to the curvilinear internal gradient vector \mathbf{g} and curvilinear internal force constant matrix \mathbf{f} using equations 30 and 31 in Ref. 37 of Sect. 20 or using equations 10 and 11 in Ref. 46 of Sect. 21.

When the option *curv1* is selected, the method of Ref. 41 of Sect. 21 is used for the generalized normal mode analysis. The option *curv1* is only available for nonredundant internal coordinates for reactions involving a total of three atoms, and it may be used whether or not the geometry along the reaction path is linear. When the *curv1* option is chosen, the curvilinear internal coordinates—two stretches and a nondegenerate bend for nonlinear triatomics or two stretches and a degenerate bend for linear triatomics—are automatically defined by the program. In this method, the \mathbf{B} matrix and \mathbf{C} tensor are constructed using the difference Cartesian coordinate formulas from Pariseau et al. [Ref. 38, Sect. 20] for the bond stretch and nonlinear valence bend coordinates and from Hoy et al. [Ref. 39, Sect. 20] for the degenerate linear bend coordinate. The difference Cartesian \mathbf{B} matrix and \mathbf{C} tensor are then transformed to Cartesian coordinates.

When the option *curv2* or *curv3* is selected, the method of Ref. 43 or 46 of Sect. 21 is used for generalized normal mode analysis. The options *curv2* and *curv3* can be used for any number of atoms, but *curv2* is used for nonredundant internal coordinates, and *curv3* is used for redundant ones. Note that stretch, nonlinear or linear bend, and dihedral curvilinear internal coordinates are allowed for this method. (The extension to linear bends was presented in Ref. 45 of Sect. 21.) When the *curv2* option is chosen, the user is required to define the curvilinear internal coordinates used for the curvilinear generalized normal mode analysis via the INTDEF keyword. In this method the Cartesian \mathbf{B} matrix and \mathbf{C} tensor are constructed by using explicit FORTRAN statements [Ref. 40, Sect. 20] for the \mathbf{B} matrix and \mathbf{C} tensor derived by Challacombe and Cioslowski [Ref. 41, Sect. 20], except for the linear bend case where the derivation is presented in Ref. 45 of Sect. 21. If the curvilinear internal coordinate is a doubly degenerate linear bend, a symbol “=” is used to indicate the connections between atoms within the INTDEF keyword.

Next, at a nonstationary point along the MEP, the curvilinear internal force constant matrix \mathbf{f} is projected as

$$\mathbf{f}^p = \left[1 - \mathbf{p}(\mathbf{B}\mathbf{u}\mathbf{B}^T) \right] \left[1 - (\mathbf{B}\mathbf{u}\mathbf{B}^T)\mathbf{p} \right],$$

where

$$\mathbf{p} = \frac{\mathbf{g}\mathbf{g}^T}{\mathbf{g}^T(\mathbf{B}\mathbf{u}\mathbf{B}^T)\mathbf{g}},$$

before applying the Wilson **GF** method [Ref. 36, Sect. 20] of vibrational analysis to obtain frequencies and generalized normal modes. The matrix **u** is a $(3N \times 3N)$ matrix with the reciprocal of each inverse mass appearing three times for each atom on the diagonal, and the matrix **B** is the Wilson B matrix.

If `POTENTIAL = unit30` or `unit40` (see Section 11.A.3), normal mode analyses at stationary points are always carried out in Cartesian coordinates. If `POTENTIAL = hooks`, normal mode analyses at minima are carried out in Cartesian coordinates, and the normal mode analysis at the transition state is carried out in curvilinear coordinates. Note that options for anharmonicity (e.g., `TOR` and `MORSE` keywords in Section 11.A.6) have not been implemented for curvilinear internal coordinates. Therefore, keyword options `curv1`, `curv2` and `curv3` should not be used with anharmonicity options.

At stationary points, one should obtain the same frequencies by using either curvilinear or rectilinear coordinates, except for small differences due to round off. However at non-stationary points the generalized normal mode frequencies do depend on the coordinate system — see Refs. 25, 41, 43, 45, and 46 in Section 21.

In the current version, both `curv1` and `curv2` are restricted to use a nonredundant set of internal coordinates. Redundant internal coordinates are always chosen by the option `curv3`. In `POLYRATE`, the number of curvilinear internal coordinates cannot exceed $3N-6$ for the `curv1` and `curv2` options. The option `curv3` allows the number of internal coordinates to be as large as $3N+6$.

10.M. IVTSTOFREQ option

In some cases, we found that the directly calculated frequencies of the lowest real-frequency modes are not stable or not accurate along the reaction path, and this can cause problems in evaluating the zero point energies, partition functions, and tunneling

probabilities. To ameliorate this problem, we have added a new scheme for interpolating frequencies. This scheme is called IVTST-0-for-frequencies; it is a single-level analog of the method used for interpolating differences of frequencies in the VTST-IC method [Ref. 36, Sect. 21]. In the IVTST-0-for-frequencies scheme, the interpolation procedure has one of four possible forms, depending on the molecularity of the end points of the interpolation. Unimolecular reactions always have a molecularity of one at the negative s terminus. Interpolations for bimolecular reactions may have molecularity two at the negative s terminus, or – instead of extending all the way to reactants, the interpolation may extend only to a reactant-side well where the molecularity is one. Similar considerations apply on the product side.

MEP type 1. Bimolecular/bimolecular interpolation

A bimolecular/bimolecular interpolation is for a reaction that has two reactants and two products. Either there are no wells present on either side of the reaction path or (more likely) wells are present but are not used in the interpolation, i.e., they are not recognized. There are two functional forms used for this case,

a. If $(\omega^\ddagger > \omega_R)$ and $(\omega^\ddagger > \omega_P)$ or if $(\omega^\ddagger < \omega_R)$ and $(\omega^\ddagger < \omega_P)$, where $\omega_m^\ddagger = \omega_m(s=0)$, $\omega_R = \omega_m(s=-\infty)$, and $\omega_P = \omega_m(s=+\infty)$, an Eckart function is used to approximate the frequencies along the reaction path:

$$\omega_m(s) = \frac{A_m Y}{1+Y} + \frac{B_m Y}{(1+Y)^2} + C_m \quad (1)$$

where

$$A_m = \omega_m(s=+\infty) - \omega_m(s=-\infty) \quad (2)$$

$$B_m = (2\omega_m^\ddagger - A_m - C_m) \pm \sqrt{2[(\omega_m^\ddagger - C_m)(\omega_m^\ddagger - A_m - C_m)]} \quad (3)$$

$$C_m = \omega_m(s=-\infty) \quad (4)$$

$$Y = \exp\left(\frac{s - S_{0,m}}{L}\right) \quad (5)$$

$$S_{0,m} = -L \ln\left(\frac{A_m + B_m}{B_m - A_m}\right). \quad (6)$$

Unlike the original VTST-IC algorithm, we use a single-level analytical expression for estimating the value of L instead of using a fit involving a lower level. The analytical expression is [Ref. 26, Sect. 21]

$$L^2 = -\frac{2V^\ddagger(V^\ddagger - A)}{\mu(\omega^\ddagger)^2 B} \quad (7)$$

$$A = V_{\text{MEP}}(s=+\infty) - V_{\text{MEP}}(s=-\infty) \quad (8)$$

$$C = V_{\text{MEP}}(s = -\infty) \quad (9)$$

$$B = (2V^\ddagger - A - C) \pm \sqrt{2[V^\ddagger - C](V^\ddagger - A - C)} \quad (10)$$

$$V^\ddagger = V_{\text{MEP}}(s = 0) \quad (11)$$

where V_{MEP} is the potential energy along the reaction path, and ω^\ddagger is the imaginary frequency at the saddle point.

b. If $(\omega_{\text{R}} > \omega^\ddagger > \omega_{\text{P}})$ or if $(\omega_{\text{R}} < \omega^\ddagger < \omega_{\text{P}})$, a hyperbolic tangent function is used to approximate the frequencies along the reaction path:

$$\omega_m(s) = \frac{1}{2} A_m \tanh\left(\frac{s - T_{0,m}}{L}\right) + E_m \quad (12)$$

where

$$E_m = \frac{\omega_m(s = +\infty) + \omega_m(s = -\infty)}{2} \quad (13)$$

$$T_{0,m} = -\frac{L}{2} \ln\left(\frac{1 + x_m}{1 - x_m}\right) \quad (14)$$

$$x_m = 2 \frac{\omega_m^\ddagger - E_m}{A_m} \quad (15)$$

c. For cases where two consecutive of the three frequencies are the same (i.e., $\omega^\ddagger = \omega_{\text{R}}$ or $\omega^\ddagger = \omega_{\text{P}}$), a constant value or used at one side of the reaction path and a symmetric Eckart function fit is used on the other. For example, if $\omega_{\text{R}} = \omega^\ddagger \neq \omega_{\text{P}}$, then

$$\omega_m(s) = \begin{cases} \omega_{\text{R}} & s \leq 0 \\ \omega_{\text{P}} + B_m Y / (1 + Y)^2 & s > 0 \end{cases} \quad (16)$$

where

$$B_m = (2\omega_m^\ddagger - \omega_{\text{P}}) \pm \sqrt{2}(\omega_m^\ddagger - \omega_{\text{P}}) \quad (17)$$

$$Y = \exp\left(\frac{s}{L}\right) \quad (18)$$

and L^2 is given by eqs. (7)–(11).

MEP type 2. Two reactants to one product (or well)

We introduce the notation P–W which denotes the product *or* the well, whichever is being used as the right-side edge of the interpolation region. In this type of reaction the function used to fit the frequencies depends on the s value, which is in the range of $-\infty < s < s_{\text{P–W}}$.

a. If $s < 0$, a hyperbolic tangent function is used to approximate the frequencies as described in the previous section for base 1b.

b. If $0 \leq s \leq s_{P-W}$, a cut-off hyperbolic tangent is used to approximate the frequencies:

$$\omega_m(s) = \frac{1}{2} A_m \tanh\left(\frac{s - T_{0,m}}{L} \left(\frac{s_{P-W}}{s_{P-W} - s}\right)\right) + E_m \quad (19)$$

where A_m , $T_{0,m}$, and E_m are the same as previous section. L is evaluated using eqs. (7)–(11). This makes $\omega_m(s)$ continuous with one continuous derivative.

If $\omega_{m,R} = \omega_m^\ddagger$ or $\omega_m^\ddagger = \omega_{m,P-W}$, then the value of $\omega_{m,R}$ or $\omega_{m,P-W}$ is modified by $\pm 2 \text{ cm}^{-1}$ to make the function monotonic.

MEP type 3. One reactant (or well) to two products

This type of reaction is similar to type II but the functional forms are reflected through $s = 0$.

MEP type 4. One reactant (or well) to one product (or well)

We introduce the notation R–W which denotes the reactant or the reactant-side well, whichever is being used as the left-side edge of the interpolation region.

a. If $(\omega^\ddagger > \omega_{R-W})$ and $(\omega^\ddagger > \omega_{P-W})$ or if $(\omega^\ddagger < \omega_{R-W})$ and $(\omega^\ddagger < \omega_{P-W})$, two cut-off Gaussian functions are used to approximate the frequencies along the reaction path:

$$\omega_m(s) = A_{m,x} \exp\left(-\frac{B_{m,x}}{1 - \left(\frac{s}{s_x}\right)^2}\right) + C_{m,x}; \quad x = R-W, P-W \quad (20)$$

$$A_{m,x} = (\omega_m^\ddagger - \omega_x) \exp(B_{m,x}); \quad x = R-W, P-W \quad (21)$$

$$C_{m,x} = \omega_x; \quad x = R-W, P-W. \quad (22)$$

The parameter $B_{m,x}$ is evaluated using a single-level analytical expression rather than a fit to a lower-level surface. In particular we use

$$B_{m,x} = \frac{s_x^2 \mu(\omega^\ddagger)}{2(V^\ddagger - V_x)}; \quad (23)$$

b. If $\omega_{m,R-W} = \omega_m^\ddagger$ or $\omega_{m,P-W} = \omega_m^\ddagger$ then the frequency is set equal to a constant on one side of the saddle and approximated by a symmetrical cut-off Gaussian is used for

the other side on the reaction path. For example, if $\omega_{m,R-W} = \omega_m^\ddagger \neq \omega_{m,P-W}$, then the frequency is set to ω_{R-W} when $s \leq 0$ and evaluated using eqns. (20–23) when $s > 0$.

c. If $(\omega_{R-W} > \omega^\ddagger > \omega_{P-W})$ or if $(\omega_{R-W} < \omega^\ddagger < \omega_{P-W})$, a cut-off hyperbolic tangent function is used to approximate the frequencies along the reaction path.

$$\omega_m(s) = \frac{1}{2} A_m \tanh \left[\frac{s - T_{0,m}}{L} \left(\frac{s_{P-W}}{s_{P-W} - s} \right) \left(\frac{|s_{R-W}|}{s - s_{R-W}} \right) \right] + E_m \quad (24)$$

where A_m , $T_{0,m}$, and E_m are determined by eqns. (2,13–16), and L is evaluated using eq. (7).

The keyword option IVTST0FREQ is used to apply the IVTST0FREQ method for interpolating the vibrational frequencies along the path. If either side of the reaction path is unimolecular or if one recognizes an intermediate structure (indicated by *WELLR or *WELLP sections), then an arc distance from saddle point has to be provided for the interpolation (by using the SR and SP keyword options).

The equations in this section are not currently published; the only available reference is this manual.

10.N. LCG4 option

Beginning with version 8.6, the options LCT3 and LCT4 are both available, but one cannot use both of them in the same run. These two options replace the previous LCT option.

The LCTOPT keyword includes four new options.

The option *vadavg* can be used with LCT3 or LCT4, and it averages the vibrationally adiabatic potential in a doubly vibrationally adiabatic region. Specifically,

$$V_{\text{eff}}(\xi) = V_{\text{eff}}^{\text{III}}(\tilde{s}_0, \xi_{\text{I}}) + \frac{\xi - \xi_{\text{III}}}{\xi_{\text{I}} - \xi_{\text{III}}} \left[V_{\text{eff}}^{\text{I}}(\tilde{s}_0, \xi_{\text{I}}) - V_{\text{eff}}^{\text{III}}(\tilde{s}_0, \xi_{\text{III}}) \right]$$

where ξ_{III} and ξ_{I} indicate the beginning and end of the region that overlaps. The default is *novadavg* and the vibrational adiabatic potential in this region is given by

$$V_{\text{eff}}(\xi) = \min \left[V_{\text{eff}}^{\text{III}}(\tilde{s}_0, \xi_{\text{III}}), V_{\text{eff}}^{\text{I}}(\tilde{s}_0, \xi_{\text{I}}) \right]$$

The option *vefsrch numb* only can be used with the LCT4 option, and it switches on the search for nonadiabatic regions that are completely adiabatic in the LCT3 method. In

particular, it becomes operative at the tunneling energy after (higher than) that for which the last nonadiabatic region was found according to the LCG3 algorithm, and the search is carried out for a total of *numb* energies beginning at that point. The search locates the maximum of the vibrationally adiabatic potential at a given tunneling energy and compares the value with the actual potential. If the actual potential is smaller than the adiabatic potential, then the procedure ends, otherwise the nonadiabatic region is extended until the vibrationally adiabatic potential is smaller than the actual potential. The default is *vefsrch* 999 because this search is required for correct LCG4 calculations. There is also an option called *novefsrch* that is equivalent to *vefsrch* 0.

For example, if 80 linear paths are calculated, and the 40 lowest-energy paths have nonadiabatic regions in LCG3, setting *numb* equal to 20 will cause the program to search for nonadiabatic regions along paths 41–60. This avoids the search for nonadiabatic regions along the 20 highest-energy paths where it is likely that nonadiabatic regions may exist only due to round-off; it allows one to avoid extra calculations at these energies very close to the effective barrier top because they have no significant effect on the results.

A restart option has been added, and it is described by two new keywords in LCTOPT. In particular, *lctstr* stores the information in unit fu49 (this is the default), and *lctsrst* restarts a calculation from unit poly.fu48. An old calculation stored in fu49 can be restored directly if fu49 is renamed as fu48. The default is *noctrst*. If *lctrst* is specified, but the unit poly.fu48 does not exist, the program stops automatically. With *lctrst* the LCTOPT options have to be the same as in previous calculations, that is, *ngamp* and *ngtheta* cannot be changed.

To incorporate the restart option in POLYRATE the files param.inc and common.inc were modified. The old LCG3 subroutine (renamed in version 8.5 as LCG) reads from unit fu48 and stores the information in the common block LCRST1 which is linked to LCVEF.

This restart option will work in POLYRATE, MORATE, and GAUSSRATE. When the restart option is used with GAUSSRATE, the restarting data for the LCT calculation will be read from fu48 instead of esp.fu83.

This restarting procedure is simple and flexible, and it allows the use of an old LCG3 fu48 file to restart an LCG3 calculation or to start a new LCG4 calculation, or vice versa. All the combinations are possible. The restart option is specially useful when we want to calculate tunneling into vibrational accessible excited states. In a first run we can do the calculation of the ground state large curvature transmission factor. This information is saved in unit FU49. We can included tunneling into vibrationally excited final states in a second run if is still worth it after looking at the output.

Further details of the LCG4 algorithm are provided in Ref. 54 in Sect. 21.

11. DESCRIPTION OF INPUT

In the following subsections, a record-by-record description of the data for each input file is given:

Subsection 11.A: the input data needed from unit fu5;

Subsection 11.B: the input data needed from unit fu29;

Subsection 11.C: the input data needed from unit fu30;

Subsection 11.D: the input data needed from unit fu31;

Subsection 11.E: the input data needed from unit fu40;

Subsection 11.F: the input data needed from unit fu50;

Subsection 11.G: the input data needed from unit fu51.

11.A. General description of file fu5 input

The input for FORTRAN unit fu5 is of the free format keyword style. The fu5 input consists of up to twelve sections. These are defined in sub-section 11.A.1. The input file consists of section headers (section headers are always preceded by a star, *) and associated keywords.

The keywords are of three types: switch, variable, and list. A switch keyword acts as a toggle for an option. A switch keyword, for example SWITCHKEY, usually has an associated form, for example NOSWITCHKEY, that causes the opposite action. A variable keyword always requires an argument. These are usually used for parameters that have to have a value, such as step sizes, convergence criteria, etc. The argument must appear on the same line as the variable keyword. The third type is the list keyword. A list keyword has one or more level-2 keywords associated with it; these keywords may be switch or variable keywords, and the latter kind have arguments. The use of a list keyword takes the form:

```
LEVEL1KEYWORD
  LEVEL2KEYWORDA
  LEVEL2KEYWORDB
  . . .
END
```

where the END line is mandatory.

All switch and variable keywords have default values, and these types of keywords need be given only if the user wishes to override the default. Some, but not all, list keywords have defaults as well. (Additional keywords, called level-2 keywords or subkeywords, are associated with some list keywords — see above. Sometimes, but not always, level-2 keywords have defaults.) List keywords start a subsection that is always terminated with an END line.

Except for the input of the TITLE keyword in the GENERAL section, all input is case insensitive, and anything after a pound sign (#) on a line is assumed to be a comment. Blank lines are ignored. It is highly recommended that the novice user print out one of the test run input files and compare it with the descriptions in the following subsections. A suggested file for this purpose is ch5j2tr3.dat. The rest of subsection 11.A details the keywords for each of the twelve sections. Please note that every keyword must be used in its proper section, i.e., between the occurrence of its starred header and the occurrence of the next starred header (or, for the last section, between the proper starred header and the end of the file).

11.A.1. Section descriptions

The following input sections correspond to the major parts of a VTST rate calculation. Each section begins with a line giving the section name, preceded by an “ * ”. An input file can contain all or some of these sections in any order except for a restart calculation. For a restart calculation, the first section in the file must be the GENERAL section and the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, START, and PATH sections cannot be used. Keywords for each section are discussed further in the rest of this subsection. For runs using the fu30, fu31, or fu40 electronic structure file input options, one should not include sections SECOND or OPTIMIZATION.

<u>Section Name</u>	<u>Description</u>
*GENERAL	general data such as the title and defining the atoms involved
*ENERGETICS	options for computing potential energy and first derivatives
*SECOND	options for computing second derivatives
*OPTIMIZATION	methods for performing geometry optimizations
*REACT1	definition and properties of the first reactant
*REACT2	definition and properties of the second reactant
*PROD1	definition and properties of the first product
*PROD2	definition and properties of the second product
*WELLR	definition and properties of the reactant side well
*WELLP	definition and properties of the product side well
*START	definition and properties of the saddle pt. or starting geom.
*PATH	options for computing the MEP
*TUNNEL	options for computing the tunneling corrections
*RATE	options for computing the reaction rate

11.A.2. GENERAL section

The GENERAL section is used to give a title that can be used to identify the run, define all the atoms in the reacting system, specify the potential energy surface or electronic structure input option to be used, set up restart criteria (if applicable), and specify whether IVTST-0, IVTST-1, VTST-IOC, or VTST-ISPE is to be used. (IVTST-0 can also be specified on file fu50.) If this is a restart run then the first keyword given in this section must be TITLE. It is good practice to always make this the first keyword so that input files can be easily identified. The table below lists all the valid keywords for the GENERAL section. Indented keywords are level-2 keywords, and the others are level-1 keywords. Level-2 keywords are always associated with a level-1 list keyword and must come after that keyword and before any other level-1 keyword or starred header. In the alphabetical glossary that follows a more complete explanation for each keyword is given. In a standard run this section would have the TITLE and ATOMS keywords. Note that ATOMS is a required keyword.

When using electronic structure input files, IVTST-M is on if and only if one uses file fu31 or if one sets LOPT(2) = -1 in file fu30 or if one sets MAXLPTS = -1 in file fu40. Otherwise IVTST-M is off, which means that any interpolation POLYRATE needs to perform to fill the save grid storage arrays is calculated by Lagrangian interpolation.

Basic:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
TITLE	list	blank	title with 5 lines maximum
ATOMS	list	required	atoms in system
CHECK	switch	off	check the input data
CLASSVIB	switch	off	classical treatment of vibrations for vibrational partition function
DL	variable	<i>none</i>	interpolated corrections
ICOPT	list	not used	interpolated correction options
BOTH	switch	off	based on corrected and uncorrected data
CORRECT	switch	on	based on corrected data
ENERGY	variable	<i>seckart</i>	Eckart function fit to $V_{MEP}(s)$
FREQ	variable	<i>icl</i>	interpolated correction of frequency
IVTST0	switch	off	zero-order IVTST
IVTST1	switch	off	first-order IVTST
GEOMUNIT	variable	<i>au</i>	specifies the units for the geometries given in GEOM keyword
MDMOVIE	switch	off	write MDMOVIE input file

Basic:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
RESTART	list	not used	restart options
WRITEFU1	switch	off	write restart file on unit 1
READFU1	switch	off	read restart file from unit 1
READFU1&2	switch	off	read restart files from units 1&2
WRITEFU3	switch	off	read from 1&2, write to 3
MERGE	variable	0	an option for merging MEPs
WRITE62	variable	none	write information in the fu62 file
WRITEFU30	switch	off	write an fu30 file
WRITEFU31	switch	off	write an fu31 file
SUPERMOL	switch	on	supermolecule mode for reactants and products

Glossary of GENERAL keywords

ATOMS

ATOMS is a required list keyword that is used to specify all the atoms in the system. For each atom the user must specify n , a sequential number unique to each atom in the molecule, and a label, either the atomic number or symbol of atom n . The mass of the atom in atomic mass units (amu) can also be specified following the label. If it is not included then the mass of the most prominent isotope is used. This program uses the universal atomic mass scale, i.e., the mass of ^{12}C is 12.000000. (For this scale, the correct abbreviation is u, although most people use the old-fashioned abbreviation amu.) The mass is supplied from an internally stored table using the atomic number or symbol as given by the user. Note that the unique number n will be used in other sections to specify the atom. The example below corresponds to monodeuterated water.

Example:

```
ATOMS
  1    O
  2    H    1.0078
  3    H    2.0140
END
```

CHECK

CHECK is a switch keyword that tells the program to stop immediately after checking the input data files. It is especially useful to use this keyword before a long or complicated calculation to make sure that all the data are input correctly. The default is off.

CLASSVIB (NOCLASSVIB)

CLASSVIB is a switch keyword that determines if the vibrational partition functions are to be calculated assuming a classical treatment of the vibrations. The default is that vibrations are treated as quantized. If CLASSVIB is used, all vibrational partition functions are calculated using the classical expression for harmonic oscillators; therefore, no anharmonicity should be specified for any modes.

GEOMUNIT

GEOMUNIT is a variable keyword that specifies the units for the geometries given in the GEOM keyword from the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections of the fu5 input file. This keyword only affects to the geometries read when the INITGEO keyword is set to *geom* and if other input files are also used (units fu29, fu30, fu31, fu40, or fu50), this keyword will have no effect on selecting the units for the geometries in those files. The default is *au*.

<u>Option</u>	<u>Description</u>
<i>au</i>	atomic units
<i>ang</i>	angstroms

Example:

```
GEOMUNIT    ang
```

DL

DL is a variable keyword that determines if the dual-level interpolated corrections are done. (See Subsection 10.H.) The default is *none*. See Section 11.F. This keyword can also be used to perform IVTST-0 calculations with unit fu50; see Section 6.N. The valid options are

<u>Option</u>	<u>Description</u>
<i>ioc</i>	dual-level using fu50 (VTST-IOC)
<i>ispe</i>	dual-level using fu51 (VTST-ISPE)
<i>none</i>	dual-level calculation is not performed (default)

Example:

```
DL    ioc
```

ICOPT

ICOPT is a list keyword that sets options for interpolated optimized corrections. If this keyword is used, then it is assumed that VTST-IOC is desired, and the DL keyword is set to *ioc*. Note that if neither *icr* nor *ica* is specified, the correction defaults to the ICL method, and if *energy deckart* is not specified then the single Eckart function is used. (This differs from versions 7.2 – 7.9.2 and earlier) The ICA, ICL, and ICR schemes, and the two ways to use Eckart functions are discussed in section 10.H. The valid options are given below.

<u>Option</u>	<u>Description</u>	<u>Default</u>
<i>both</i>	separate calculations will be based on the corrected and uncorrected data	off
<i>correct energy</i>	calculation will be based only on the corrected data interpolated optimized corrections for energies: <i>deckart</i> - difference of two Eckart functions fit to the $V_{MEP}(s)$ curve at higher level and Eckart function fit to lower level <i>seckart</i> - do not use Eckart function fit to the $V_{MEP}(s)$ curve at higher level but rather use a single Eckart fit to the difference	on <i>seckart</i>
<i>freq</i>	interpolated optimized corrections for frequencies: <i>ica</i> - interpolated correction based on arithmetic differences <i>icl</i> - interpolated correction based on logarithm of ratios <i>icr</i> - interpolated correction based on ratios <i>none</i> - no corrections	<i>icl</i>

Example:

```
ICOPT
    ENERGY DECKART
    FREQ ICA
    BOTH
END
```

IVTST0 IVTST1

IVTST0 and IVTST1 are switch keywords to turn on the zero- and first-order Interpolated VTST (IVTST-0 and IVTST-1) calculations, respectively (See Subsection 6.N.). The default is off.

MDMOVIE

MDMOVIE is a switch keyword that determines if an input file for MDMOVIE should be written out. MDMOVIE is a visualization program that allows animation of motion of the system along the MEP. The default is off.

RESTART

RESTART is a list keyword that gives the user control over the restart capability of the run or designates the current run as a restart calculation. See Section 7.C. for general comments on this option. If this keyword is not used, a future calculation at a different temperature must start from scratch. Alternatively a user can use the *writefu1* option within RESTART and the reaction path will be written to unit fu1. In future runs, the user would just need to specify the *readfu1* option within RESTART for the same reaction path to be read from unit fu1. All the valid options are given below. Note that *merge* requires an argument specifying the increment value. The default is that this keyword is not included, i.e. this is not a restart run and no restart file should be written.

<u>Option</u>	<u>Definition</u>
<i>writefu1</i>	write restart information to unit fu1 for future use
<i>readfu1</i>	restart calculation using information on unit fu1
<i>readfu1&2</i>	restart calculation using merged MEP, MEP #1 on unit fu1 is merged with MEP #2 on unit fu2 and the merged file is not saved.
<i>writefu3</i>	the merged MEP from <i>readfu1&2</i> is written to unit fu3
<i>merge x</i>	increment in <i>s</i> (in bohr) to be added to the <i>s</i> values of MEP #2 when it is merged with MEP #1. This is required if the READFU1&2 option is used.
<i>writefu30</i>	write an fu30 file
<i>writefu31</i>	write an fu31 file

Example:

```
RESTART
      READFU1
END
```

NOTE: When writing an fu30 file (*writefu30* option) the user needs to supply an fu30 file with the necessary LOPT(1-40) options.

SUPERMOL (NOSUPERMOL)

SUPERMOL is a switch keyword which indicates that the supermolecule mode is used for reactants and products. The default for SUPERMOL in POLYRATE is on. This keyword is used only when the keyword POTENTIAL is set equal to *hooks*, and the choice has no effect if there is only one reactant and one product. The use of supermolecule mode is most easily explained by an example. Consider a bimolecular reaction with reactants A and B. For an analytical potential energy function or a non-size-extensive electronic structure method, one would usually calculate the energy of reactants by placing A at a distance of 20–100 a_0 from B and calculating the energy of the AB supermolecule. However, for size-extensive methods like Hartree-Fock and MP2, one would usually prefer to calculate the energies of A and B separately and add them. The latter requires NOSUPERMOL.

For a bimolecular system, NOSUPERMOL causes POLYRATE to calculate the zero of energy as the sum of the potential energies of the two reactants using two separate calculations in which each reactant is treated separately. This is not very useful with an analytical potential energy surface. When NOSUPERMOL is specified, the geometries in REACT1 and REACT2 each contain less than the full number of atoms. In the case of SUPERMOL, the first reactant is optimized and its energy is calculated by having the second reactant frozen at a long distance (which is specified through the definition of the geometries in section REACT2 in file fu5). Then the second reactant is optimized with the first one already at its equilibrium geometry, but far away, and the sum of their energies is obtained as the energy of the supermolecule calculation, i.e., when SUPERMOL is on one does not add the energies from REACT1 and REACT2 at the end of the calculation. The same considerations apply to PROD1 and PROD2 if there are two products.

TITLE

TITLE is a list keyword that allows the user to give a title to the run. The user can specify up to five lines of 80 characters each. Like all list variables, an end must be given at the end of the TITLE subsection. This must be the first keyword list in the GENERAL section for a restart calculation.

Example:

```
TITLE
      CH3 + H2 -> CH4 + H TEST RUN
      ESD/HARMONIC
END
```

WRITE62

WRITE62 is a variable keyword that determines if a subset of the stationary point information from file fu61 should be printed in fu62 file. The default is *none*. The valid options are

<u>Option</u>	<u>Description</u>
<i>none</i>	no information is printed in fu62
<i>viball</i>	frequencies (in cm^{-1}) are printed for all stationary points
<i>vibsp</i>	frequencies (in cm^{-1}) are printed only for the saddle point

Example:

```
write62 vibsp
```

WRITEFU30

WRITEFU30 is a switch keyword that determines if an output fu30 file should be written out. Notice that the points are not in the required ascending order. Therefore the points in the created fu30 file need to be reordered by the user before using the file. The default is off.

WRITEFU31

WRITEFU31 is a switch keyword that determines if an output fu31 file should be written out. The written file is ready to use with the *unit31* option for the POTENTIAL keyword in the ENERGETICS section of the fu5 input file. The options for writing the fu31 file cannot be selected. It will always be written in atomic units, and the program will always write unscaled gradients and packed Hessians in the fu31 file.

11.A.3. ENERGETICS section

The ENERGETICS section is used to define the method for computing the potential energy of the reacting system and the corresponding first derivatives. The table below lists all the valid keywords for the ENERGETICS section. In the alphabetical glossary that follows a more complete explanation for each keyword is given.

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
EZERO	variable	<i>calculate</i>	method for determining zero of E
EZUNIT	variable	<i>a.u.</i>	units of zero of energy
POTENTIAL	variable	<i>hooks</i>	type of PES to be used

Glossary of ENERGETICS keywords

EZERO

EZERO is a variable keyword specifying how the zero of energy should be determined. This keyword is discussed briefly in detail in Section 8.B. POLYRATE always sets the zero of energy equal to the energy of the reactants, which is defined to be the sum of the energies of both reactants for a bimolecular reaction or to be the energy of the single reactant for a unimolecular reaction. However, the other option is to read in the zero of energy with option *read #*. The default is *calculate*.

<u>Option</u>	<u>Definition</u>
<i>calculate</i>	If <i>calculate</i> is chosen the program will set EZERO to the value that it calculates for the energy of the reactants.
<i>read #</i>	The <i>read #</i> option, where # is the value of the energy of the reactants in units specified by the EZUNIT keyword, causes the program to set EZERO to #.

Example:

```
EZERO read 0.1354
```

Notes:

- (1) If the *read #* option is chosen, the user must ensure that # is the value of the energy of the reactants in units specified by the EZUNIT keyword.
- (2) If the WKB method is used for any of the vibrational modes of reactant 1, then the *read #* option must be used. Otherwise the user can choose *calculate*.
- (3) Another possible use of *read #* would be to allow efficient calculations for cases where the reactant energy is already known and would be expensive to recalculate. However POLYRATE always calls at least one of the hook routines for the reactants, so this efficiency must be implemented by the user if desired.

EZUNIT

EZUNIT is a variable keyword specifying the units used for the zero of energy. The default is *a.u.* The other valid option is *kcal*. Note 1 a.u. = 627.5095 kcal/mol. This keyword must be read *before* reading the value of the zero of energy with the *read #* option for the keyword EZERO, if that option is used.

<u>Option</u>	<u>Definition</u>
<i>a.u.</i>	atomic units (hartrees)
<i>kcal</i>	kcal/mol

Example:

```
EZUNIT kcal  
EZERO read 100.1354
```


POTENTIAL

POTENTIAL is a variable keyword that allows the user to specify the source of the potential energy information for a single-level calculation or for the lower level of a dual-level calculation. In the current version of POLYRATE there are five valid options: *hooks*, *unit29*, *unit30*, *unit31*, or *unit40*. The option *hooks* means that the user is providing potential information through the hooks. When POLYRATE is used as a stand-alone package, the hooks call SURF and SETUP; this is how one supplies an analytic potential energy surface (as discussed in Section 8 of this manual). If *unit29* is used the information is read from file fu29 for an IVTST-0 or IVTST-1 calculation. Specification of one of the keywords *unit30*, *unit31*, or *unit40* means that the MEP data is to be read from file fu30, fu31, or fu40, respectively. (as discussed in Sections 9 and 11.C, 11.D, and 11.E of this manual) The fu30 and fu40 options (*unit30* and *unit40*, respectively) are called the electronic structure input file options in older versions of POLYRATE.

The default is *hooks*. Derivative versions of POLYRATE will also have various options for interfacing with electronic structure packages like ACES II, GAUSSIAN, MOPAC, AMSOL, etc. This may occur through the hooks (modular mode) or in a less general fashion (non-modular mode), but now electronic structure input can be used with any of the five choices of POTENTIAL.

<u>Option</u>	<u>Definition</u>
<i>hooks</i>	use the hooks
<i>unit29</i>	read in from electronic structure file fu29
<i>unit30</i>	read in from electronic structure file fu30
<i>unit31</i>	read in from electronic structure file fu31
<i>unit40</i>	read in from electronic structure file fu40

Example:

```
POTENTIAL unit30
```

11.A.4. SECOND section

The SECOND section is used to define the method for computing the Hessian of the reacting system when the POTENTIAL keyword is set to the *hooks* option; this section should not be included if the option selected for POTENTIAL is *unit30*, *unit31*, or *unit40*. The table below lists all the valid keywords for the SECOND section. In the alphabetical glossary that follows a more complete explanation for each keyword is given.

Basic:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
HESSCAL	variable	<i>ghook</i>	how to calculate Hessians

Second derivative POLYRATE methods:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<u>Other:</u>			
FPRINT	switch	off	print second derivative information
NUMSTEP	variable	1×10^{-4}	step size for 2nd derivatives of <i>V</i>
NUMTYPE	variable	<i>central</i>	method for computing 2nd deriv.

Glossary of SECOND keywords

FPRINT (NOFPRINT)

FPRINT is a switch keyword that is used if the normal mode energies, eigenvectors and eigenvalues are to be printed following the normal mode analysis. Default is NOFPRINT.

HESSCAL

HESSCAL is a variable keyword specifying how to calculate Hessians. The default is *ghook* which means POLYRATE will calculate the Hessians by finite differences from the gradients as specified by NUMTYP. Another option is *hhook* which means the Hessians are evaluated by calls to the HHOOK subprogram. See Section 8.C.

<u>Option</u>	<u>Definition</u>
<i>ghook</i>	call GHOOK to get first derivatives, and use central differences to compute the second derivatives of the potential. The step size used is given by the NUMSTEP keyword.
<i>hhook</i>	call HHOOK to get second derivatives.

Example:

```
HESSCAL  ghook
```

NUMSTEP

NUMSTEP is a variable keyword giving the step size (in unscaled bohrs) for the numerical second derivative of potential energy function with respect to the unscaled atomic coordinates for frequency calculations. The default value is 1.0×10^{-4} .

Example:

```
NUMSTEP  0.00005
```

NUMTYP

NUMTYP is a variable keyword that is used to specify the method for computing the numerical second derivatives of the potential energy with respect to unscaled atomic Cartesians if HESSCAL is *ghook*. See Section 10.C. The default is *central*. The allowed options are:

<u>Option</u>	<u>Definition</u>
<i>central</i>	use central differences of analytic first derivatives to compute the second derivatives of the potential. The step size used is given by the NUMSTEP keyword.
<i>quadratic</i>	use a quadratic fit of the potential to compute the second derivatives of the potential. The step size used is given by the NUMSTEP keyword.
<i>fourth</i>	use a fourth order polynomial fit to the analytic first derivatives to compute the second derivatives of the potential. Step sizes used are NUMSTEP and 2*NUMSTEP.

Example:

NUMTYP *fourth*

11.A.5. OPTIMIZATION section

The OPTIMIZATION section is used to specify the method for carrying out geometry optimizations when the POTENTIAL keyword selection is *hooks*. This section is not needed if POTENTIAL is set to *unit30*, *unit31*, or *unit40*. The table below lists all the valid keywords for the OPTIMIZATION section. In the alphabetical glossary that follows, a more complete explanation for each keyword is given.

Basic:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
OPTMIN	variable	<i>bfgs</i>	how to optimize minima
OPTTS	variable	<i>nr</i>	how to optimize transition state

Two optimization routines are available in POLYRATE. The first is quasi-Newton with BFGS Hessian updates. The alternative is Newton-Raphson with Brent line minimization, which is called the NR method. For minima (equilibrium structures of reactants and products), the default optimizer is BFGS. For saddle points (transition states), the default optimizer is the NR method. See Section 6.E for further details.

POLYRATE optimization methods:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<i><u>Common:</u></i>			
PRINT	switch	off	print geometrical parameters and derivatives after each opt step
<i><u>Other:</u></i>			
DLX2	variable	1×10^{-5}	step size for first derivative of grad in geometry optimization
EFOPT	list	off	indicate the option for EF
GCOMP	variable	1×10^{-5}	maximum allowed component of the gradient for convergence at the minima
TSGCOMP	variable	1×10^{-5}	maximum allowed component of the gradient for convergence at the saddle point
HREC	variable	10	recalculate Hessian every HREC step for convergence at minima
TSHREC	variable	5	recalculate Hessian every TSHREC step for convergence at saddle point
NITER	variable	50	max number of iterations for opt
RETRY	switch	off	switch to full Newton method if BFGS optimization fails
SCALE	variable	0.2	step size scaling factor in Newton search
SDSTART	switch	on (minima opt) off (TS opt)	the first optimization step is steepest-descent
STPTOL	variable	1×10^{-5}	“optimization failure” parameter

Glossary of OPTIMIZATION keywords

DLX2

DLX2 is a variable keyword used to specify the step size (in unscaled bohrs) for calculating the first derivatives of the gradients of the potential energy function in geometry optimization. The default value is $1 \times 10^{-5} a_0$.

Example:

```
DLX2                0.000012
```

EFOPT

EFOPT is a list keyword used to specify information for the Eigenvector Following (EF) algorithm. There are five possible variables that may be specified, namely, RMAX, RMIN, OMIN, DDMAX, and DDMAXTS. The RMAX and RMIN variables bracket the ratio of the calculated energy to the predicted energy; the defaults are RMIN = 0 and RMAX = 4. The OMIN variable is used to indicate the minimum of the overlap in the direction which is determined by the dot product between the eigenvector and the previous followed direction; the default is 0.0 for reactants and products and 0.8 for a saddle point. The DDMAX and DDMAXTS variables are used to set the maximum of the trust radius (in angstroms); the DDMAX variable is used for the reactants and products, and the DDMAXTS variable is used for the saddle point. The defaults are DDMAX = 0.5 and DDMAXTS = 0.3.

Example:

```
EFOPT
  RMAX    2.0
  RMIN    1.2
  OMIN    0.2
END
```

GCOMP

TSGCOMP

GCOMP and TSGCOMP are variable keywords used to specify the convergence criteria for optimization of minima and transition states, respectively. The optimization is considered converged if the maximum component of the gradient in atomic units is smaller than the value specified. The default value is $1 \times 10^{-5} E_h/a_0$ for both these keywords. Sometimes, if the saddle point region is very flat, 1×10^{-5} is not accurate enough to produce a good reaction path; in such cases we recommend 1×10^{-7} .

Example:

```
GCOMP                0.000001
```

HREC TSHREC

HREC and TSHREC are variable keywords used to specify how often an accurate Hessian is to be computed in a BFGS or Newton-Raphson calculation. See Section 6.E for more details. Notice that if the number specified is equal to 1 in a Newton-Raphson calculation (*nr* option for keywords OPTMIN or OPTTS), a full Newton method is used. At a step when the Hessian is not recalculated accurately, the Hessian is either estimated using the BFGS formula (*bfgs* option for keywords OPTMIN and OPTTS) or kept frozen (*nr* option for keywords OPTMIN and OPTTS). The defaults are HREC equal to 10 and TSHREC equal to 5.

Example:
HREC 1

NITER

NITER is a variable keyword used to specify the maximum number of iterations to be used in optimizing the geometry. The default value is 50.

Example:
NITER 75

OPTMIN

OPTMIN is a variable keyword that determines how geometries of minima are optimized. The default is *bfgs*.

<u>Option</u>	<u>Definition</u>
<i>bfgs</i>	specifies to use the Broyden-Fletcher-Goldfarb-Shanno Hessian update method. The Hessian is calculated using the option specified in HESSCAL in section SECOND at the initial geometry of the search; at subsequent geometries, the Hessian is estimated using information about the Hessian at the previously geometry and the gradient at the previous and current geometry. (When the search is complete, the Hessian is again calculated by the method specified in HESSCAL, and this Hessian is used for frequencies.)
<i>ef</i>	specifies to use the Eigenvector Following algorithm.
<i>nr</i>	specifies to use the Newton-Raphson algorithm with Brent line minimization.
<i>ohook</i>	optimizes the geometry by calling hooks subprogram OHOOK using options specified in files fu71 through fu74. See Section 8.C.

Example:
OPTMIN *bfgs*

OPTTS

OPTTS is a variable keyword that determines how the geometry of the transition state is optimized. The default is *nr*.

<u>Option</u>	<u>Definition</u>
<i>bfgs</i>	specifies to use the Broyden-Fletcher-Goldfarb-Shanno Hessian update method. The Hessian is calculated using the option specified in HESSCAL in section SECOND at the initial geometry of the search; at subsequent geometries, the Hessian is estimated using information about the Hessian at the previous geometry and the gradient at the previous and current geometry. (When the search is complete, the Hessian is again calculated by the method specified in HESSCAL, and this Hessian is used for frequencies.)
<i>ef</i>	specifies to use the Eigenvector Following algorithm.
<i>nr</i>	specifies to use the Newton-Raphson algorithm with Brent line minimization..
<i>ohook</i>	optimizes TS geometry by calling hook subprogram OHOOK using options specified in file fu75. See Section 8.C.

Example:

```
OPTTS      nr
```

PRINT (NOPRINT)

PRINT is a switch keyword for printing the geometry and the derivatives after each step in the geometry optimization. Default is off, i.e., NOPRINT.

RETRY (NORETRY)

RETRY is a switch keyword specifying to switch to full Newton search (HREC = 1, no linear search) if the optimization stalls (see STPTOL keyword), instead of quitting. Since this could be quite time consuming, this option has to be used with care. The default is NORETRY.

Note: the difference between full Newton and quasi-Newton is that full Newton calculates a full new Hessian at every step whereas quasi-Newton uses an approximate updated Hessian. In many cases, if the standard search fails it is better to restart the optimization with a different geometry than to use full Newton.

SCALE

SCALE is a variable keyword giving the unitless step size scaling factor in the Newton search for a stationary point. The default value is 0.2.

Example:

```
SCALE      1.1
```

SDSTART (NOSDSTART)

SDSTART is a switch keyword that specifies to start the optimization with a steepest-descent step. This is equivalent to defining the Hessian at the starting point as a unit matrix. This keyword is meaningful only in minima optimizations where BFGS is specified. This option proves useful in speeding up convergence of the optimization when the starting point happens to be outside the quadratic region of the surface around the minimum (the initial guess geometry is bad). The default is SDSTART.

STPTOL

STPTOL is a variable keyword that specify the conditions for optimization failure. If the maximum component of the Newton optimization step is smaller than the value specified by STPTOL, the optimization is considered stalled since the geometry is not changing anymore, and the program quits. This could happen if the (estimated) Hessian used is bad; that in turn probably means that either HREC or SCALE are too large. The default value is $1 \times 10^{-5} a_0$.

Example:

```
STPTOL 0.0001
```

11.A.6. REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, AND START sections

The REACT1, REACT2, PROD1, PROD2, WELLR, WELLP and START sections all have the same set of keywords. A separate section is needed for each reactant and product in the system and for the saddle point or nonstationary starting point. Thus for a system with one reactant, one product, and a saddle point the user needs to include the REACT1, PROD1, and START sections in the input file. For simplicity the keywords have been broken into two categories: basic and anharmonicity. This is only for clarity in the manual. The keywords in these categories can be used in any order, except for the STATUS keyword which must be given as the first option line of a section. Also for simplicity the keywords in each category have been labeled as either required, common, or other. The required keyword (GEOM) must be included. The keywords labeled common are those that might be expected to be found reasonably frequently in input files for typical POLYRATE runs. This labeling is only intended as an aid to the novice user. Again an alphabetical glossary of all the keywords follows the two summary tables.

Notes that START refers to the saddle point if there is one and to the starting point if there is no saddle point. Also note that the anharmonicity keywords in the START section control the anharmonicity all along the reaction path since it must be consistent with the START point.

The first restart option (See Section 7.C) does not use any of these sections, instead it is controlled by the RESTART keyword in the GENERAL section. However, the second restart option (See Section 7.C) is controlled by the STATUS keyword of this section. See the glossary for the definition and proper usage of the STATUS keyword.

Basic:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<i><u>Required:</u></i>			
GEOM	list	required	Cartesian coordinates of atoms
<i><u>Common:</u></i>			
CONSTANT	list	not used	coordinates held constant in opt.
ELEC	list	all degen. = 1	electronic degeneracies
ENERGY	variable	none	energy of the stationary points
EIGENVECTOR	list	none	eigenvector of the saddle point
FREQ	switch	on	perform normal-mode analysis
FREQUNIT	variable	waven	units of the frequencies
HESSIAN	list	none	Hessian of the saddle point
INITGEO	variable	geom	location of input data that specifies initial geometries
STATUS	variable	0	status of the stationary point work
SPECIES	variable	nonlinrp	species type (linearity and phase)
VIB	list	none	frequencies of the stationary points
<i><u>Other:</u></i>			
DIATOM	list	do not use Morse function	treat diatomic as a Morse function
RE	variable	none	Morse equilibrium bond length
BETA	variable	none	Morse range parameter
DE	variable	none	Morse dissociation energy
LINAXIS	variable	z-axis	orientation of a linear molecule
PROJECT	switch	on	project out overall trans. and rot.

Anharmonicity:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<u>Common:</u>			
ANTLR	variable	1×10^{-8}	criterion on Morse x_e
DEMIN	variable	0.159	lowest dissociation energy (D)
DQQP	variable	0.0001, 0.001	two distances for quad.- quar. fit
HARMONIC	switch	on	all modes harmonic
MORSE	switch	off	all modes Morse
MORSEQQ	switch	off	all modes Morse/quadratic-quartic
MORMODEL	variable	<i>morsei</i>	Morse options
PRMODE	switch	off	print XMOL movie clips
QQWKB	switch	off	WKB with a quadratic-quartic fit
QQSEMI	switch	off	semi-classical WKB with a quadratic-quartic fit to potential
TOR	list	not used	information for modes treated as torsions
TOROPT	list	not used	more information for modes treated as torsions
VANHAR	list	not used	each mode is treated individually
VRANGE	list	not used	each mode is treated individually for a given range of s
WKB	list	not used	WKB with true potential
WKBTOL	variable	1×10^{-8}	tolerance for finding eigenvalues
KBQUAD	variable	40	no. of quad. pts in phase integrals
KBPRINT	switch	off	print turning points

Glossary of REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START keywords

ANTLR

ANTLR is a variable keyword whose argument is the criterion on the unitless Morse anharmonicity parameter, x_e , such that the quadratic-quartic model is used if $x_e < \text{ANTLR}$. The *antlr* keyword is only used with the *morseqq* method. The default value is 1×10^{-8} .

Example:

```
ANTLR      0.000013
```

CONSTANT

CONSTANT is a list keyword used to specify which of the Cartesian coordinates should be held constant during the Newton search to find the species' equilibrium geometry. In this list the unique atom number is used along with *x*, *y*, and/or *z* for those coordinates to be held constant. The default is that all coordinates are optimized. In most cases the user needs to specify at least six Cartesian coordinates to be constant for a nonlinear species and five for a linear species. Note that if this keyword is specified, the default is erased, and the user should input all the coordinates which are to be held constant.

Example:

```
CONSTANT
  1      x      y      z
  2      y      z
  3      z
END
```

DEMIN

DEMIN is a variable keyword used to input the lowest reactant or product dissociation energy in a.u., measured from the reactant limit. This is called *D* in published papers. It is needed for any of the Morse anharmonicity options and for μVT calculations. There is a default value of $0.159 E_h$ ($= 100 \text{ kcal/mol}$) for this variable but it is dangerous to use the default. The user is strongly encouraged to understand *D* if Morse anharmonicity is used.

Example:

```
DEMIN      0.1733
```

DIATOM

DIATOM is a list keyword that is used if the particular reactant or product is a diatomic molecule and is to be treated as a Morse oscillator. The default is to treat diatomics by the same methods as any other kind of molecule; thus this keyword would not be used. Note that if the DIATOM option is chosen, then the user must specify all three options, R_e , β , and D_e , which are the Morse parameters for equilibrium geometry, dissociation energy, and range parameter, respectively, all in atomic units. There are no defaults for these options. The example below corresponds to the O_2 molecule.

Example:

```
DIATOM
      RE          2.3158
      BETA        1.4694
      DE          0.1779
END
```

DQQP

DQQP is a variable keyword with two arguments, which are respectively the first and second step sizes in bohr used when fitting the quadratic-quartic potential to two points along the normal coordinate. Note that the first argument (the first step size) must be less than the second. DQQP is to be used only if one of the keywords QQWKB or QQSEMI is specified. Default is $1 \times 10^{-4} a_0$ and $1 \times 10^{-3} a_0$.

Example:

```
DQQP 0.03 1.2
```

EIGENVECTOR

This keyword should be used only in a START section and it should be used if and only if the STATUS option is equal to 6. EIGENVECTOR is a list keyword that is used to specify the eigenvectors of the saddle point in an unscaled Cartesian coordinate system. The eigenvectors of the stationary point should be given in full matrix form with the columns being a particular mode. In addition, the axes must be consistent with the geometry in GEOM.

Example:

```
EIGENVECTOR
  L11  L21  L31  L41  L51  L61
  L12  L22  L32  L42  L52  L62
  L13  L23  L33  L43  L53  L63
  L14  L24  L34  L44  L54  L64
  L15  L25  L35  L45  L55  L65
  L16  L26  L36  L46  L56  L66
END
```

Explanation of example:

L_{ij} is the j th component of the i th eigenvector. Note that $j = 1$ is for the x component of atom 1, $j = 2$ is for the y component of atom 1, $j = 3$ is for the z component of atom 1, $j = 4$ is for the x component of atom 2, etc. Note also that the eigenvalues must be in reverse canonical order (this order is described in Sect. 6.E). Thus, I=1 is for the imaginary-frequency normal mode, then come the eigenvectors corresponding to translations and rotations, and finally come the true vibrations, in order of increasing frequency.

ELEC

ELEC is a list keyword used to specify the degeneracy and energy in atomic units of each electronic state of this species. Up to three states are allowed. The ground state should be given first. The default is for all three states to be singly degenerate. The example given below is for the 3P multiplet of an oxygen atom.

Example:

```
ELEC
  5    0.00000000
  3    0.00072218
  1    0.00103201
END
```


ENERGY

This keyword should be used if and only if STATUS \geq 4. ENERGY is a variable keyword that is used to specify the energies (a.u.) of a reactant, product, or saddle point.

Example:

```
ENERGY    -2.4221970828E-02
```

FREQ (NOFREQ)

The FREQ keyword is used to specify whether normal mode analysis is carried out. The default is FREQ.

FREQUNIT

The FREQUNIT keyword is used to specify the units of the frequencies read by the FREQ keyword. The default is *au*.

<u>Option</u>	<u>Definition</u>
<i>waven</i>	Frequencies are given in cm^{-1} .
<i>au</i>	Frequencies are given in atomic units.

GEOM

GEOM is a list keyword that is required for all calculations except those with the readfu1 or readfu1&2 option selected for RESTART in the GENERAL section. In this list the Cartesian coordinates of all atoms in this particular reactant, product or reaction path species must be given along with each unique atom number. If STATUS (see below) is 0, the user should choose a geometry (in bohrs, not mass-scaled) that is a guess to the species' stationary-point geometry. If status is \geq 2, then the geometry should be the correct geometry, not just a guess. (See the ATOMS keyword in the GENERAL section for the definition of the unique atom number. This number must be consistent for each atom in ATOMS and GEOM.) For a linear species, the user must specify a linear geometry. The example below is for the same deuterated water molecule used in the ATOMS example.

Example:

```
GEOM
  1    0.000000    0.000000  0.000000
  2   -0.456364   -1.749560  0.000000
  3    1.808100    0.000000  0.000000
END
```

HESSIAN

If the option of the STATUS keyword is 4, the Hessian of the saddle point should be read in as a triangular form in unscaled Cartesians. HESSIAN is a list keyword that is used to indicate the Hessian (hartrees per square bohr) of the saddle point. This keyword is only valid for the START section, and can not be used in the REACT1, REACT2, PROD1, and PROD2 sections. In addition, the axes must be consistent with the geometry given in GEOM. The format is free format, with any number of entries per line and entries on the same line separated by one or more spaces. Since the Hessian should be a symmetric matrix we read only the lower triangular part.

Example:

```
HESSIAN
      F11  F21  F22  F31  F32  F33
      F41  F42  F43  F44  F51  F52
      F53  F54  F55  F61  F62  F63
      F64  F65  F66

END
```

Explanation of the example:

The element F_{jj} of the Hessian matrix \mathbf{F} corresponds to the same order of components as explained for j under the EIGENVECTOR keyword.

HARMONIC

HARMONIC is a switch keyword that is used when one wishes to treat all vibrational modes harmonically. It is the default; i.e., if no keyword is given for anharmonicity then the harmonic approximation is used. The default is that harmonic is on, so when it is specified, the purpose is just for clarity (i.e., to make the default explicit).

INITGEO

INITGEO is a variable keyword that is used to indicate the location of the input data that specifies the initial reactant 1, reactant 2, product 1, product 2, and starting geometries. The default is *geom*. The valid options are:

<u>Option</u>	<u>Definition</u>
<i>geom</i>	the initial geometries are taken from the lists following the GEOM keywords in the react1, react2, prod1, prod2, and start sections of the fu5 input file.
<i>hooks</i>	the initial geometries are taken from the external input files fu71-fu78.

Example:
INITGEO geom

LINAXIS

LINAXIS is a variable keyword that is used to indicate the orientation of a linear reactant, product, or starting molecule. The default is that LINAXIS is off if the initial geometry is nonlinear and that it is on if the initial geometry is linear. If LINAXIS is on, but no option is specified, the default value is *z-axis*. The valid options are:

<u>Option</u>	<u>Definition</u>
<i>x-axis</i>	the linear molecule is aligned with the x-axis
<i>y-axis</i>	the linear molecule is aligned with the y-axis
<i>z-axis</i>	the linear molecule is aligned with the z-axis

MORMODEL

MORMODEL is a variable keyword that is used to specify which Morse model is to be used. This keyword is used only if the MORSE or MORSEQQ switch is on. The default is *morsei*. Note that the value of DEMIN is specified by the user using the keyword DEMIN. All the valid options are given below (see Ref. 1 in Sec. 21 for explanations of these models):

<u>Option</u>	<u>Definition</u>
<i>morsei</i>	Morse I: $V_{\text{MEP}} + \text{local D} = \text{DEMIN}$
<i>morseii</i>	Morse II: local D from V'''
<i>morseiii</i>	Morse III: Morse II if $(V_{\text{MEP}} + \text{local D}) > \text{DEMIN}$; otherwise Morse I
<i>morseia</i>	Morse Ia: like Morse I except use hyperradius to determine side of MEP on which concave-side turning point is to be found if SCT or LCT tunneling correction is done.
<i>morseiia</i>	Morse IIIa: like Morse III except replace Morse I by Morse Ia, i.e., find direction of concave side from hyperradius when using Morse I and find it using 3rd derivative when using Morse II

Example:
MORMODEL morseiii

MORSE

MORSE is a switch keyword that is used if all vibrational modes are to be treated using the Morse approximation with the same value of MORMODEL. If this option is chosen, then the Morse model is specified by the keyword MORMODEL. Default is off.

MORSEQQ

MORSEQQ is a switch keyword that is used if all vibrational modes are to be treated using Morse/quadratic-quartic approximation with the same value of MORMODEL. This means that the Morse model is used for modes with Morse anharmonicity parameter $x_e \geq$ ANTLR, and the quadratic-quartic option based on second and fourth derivatives of the potential along the generalized normal mode is used for modes with $x_e <$ ANTLR. If this option is chosen, then the Morse model is specified by the keyword MORMODEL. Default is off.

PRMODE

PRMODE is a switch keyword used to print movie clips of the vibrational modes for the XMOL program in XYZ format. The file name of a movie clip consists of two parts. The first part has 4 letters which is used to indicate the type of the species: R1MD for reactant 1, R2MD for reactant 2, P1MD for product 1, P2MD for product 2, RWMD for reactant well, PWMD for product well, or TSMD for the saddle point. And the second part has a three-digit integer which is used to indicate the index of the vibrational mode in decreasing order. Therefore, the maximum number of vibration modes for which this keyword can be used is 999. Currently, each clip has 20 frames.

PROJECT (NOPROJECT)

PROJECT is a switch keyword that tells the program to use projection operators to remove overall translations and rotations from the second derivatives at the stationary points. This projection is not optional along the reaction path, but is optional at stationary points. Even at stationary points, one obtains more stable numerical results if projection is applied. The default is PROJECT.

QQSEMI

QQSEMI is a switch keyword that is used if all vibrational modes are to be treated by using a quadratic-quartic fit to the potential at two distances (specified by the DQQP keyword) along the generalized normal mode with the uniform semiclassical WKB method for computing energy levels. See Section 10.B.1. Default is off.

QQWKB

QQWKB is a switch keyword that is used if all vibrational modes are to be treated using the WKB method with a quadratic-quartic fit to the potential. The fit is done at two distances, specified by the DQQP keyword, along the generalized normal mode with the primitive WKB method for computing energy levels. See Section 10.B.1. Default is off.

SPECIES

SPECIES is a variable keyword that is used to specify the linearity and phase of the reactant, product, or starting species. See Section 6.B for discussion. The default is *nonlinrp*. The valid options are given below.

<u>Option</u>	<u>Definition</u>
<i>atomic</i>	species is a single atom
<i>linrp</i>	species is a linear reactant or product
<i>nonlinrp</i>	species is a nonlinear reactant or product
<i>ssrp</i>	species is a solid-state reactant or product
<i>lints</i>	species is a linear transition state or generalized transition state
<i>nonlints</i>	species is a nonlinear transition state or generalized transition state
<i>ssts</i>	species is a solid-state transition state or generalized transition state

Example:

```
SPECIES      linrp
```

STATUS

STATUS is a variable keyword that is used to indicate the status of the stationary points. When the STATUS keyword is used it must be given as the first option line of the *REACT1, *REACT2, *PROD1, *PROD2, *WELLR, *WELLP, or *START section in which the keyword is used. The options are:

option	*REACT1, *REACT2, *PROD1, *PROD2, *WELLR, or *WELLP	*START
0	calculate geometry, energy, and frequencies	calculate geometry, energy, frequencies, and eigenvectors
2	read in geometry; calculate energy and frequencies	read in geometry; calculate energy, frequencies, and eigenvectors
4	not available	read in geometry, energy and Hessian; calculate frequencies and eigenvectors
6	read in geometry, energy, and frequencies	read in geometry, energy, frequencies, and eigenvectors*

* Note: if *cubic* is chosen as the value of the FIRSTSTEP keyword in the PATH section, then option 6 is not valid.

Note that in the START section if STATUS is 4, then the ENERGY and HESSIAN lists must be given and if STATUS is 6, then the ENERGY, FREQ, and EIGENVECTOR lists must be given. The GEOM keyword is always required.

Example:

```
STATUS 2
TOR
```

TOR is a list keyword that is used to specify the information needed for torsional modes that will be treated using the hindered-internal-rotator approximation. The hindered internal rotator approximation is only used for a subset of the modes, and this keyword is used to give the parameters for each of those modes. The user must also use the VANHAR or VRANGE (for generalized transition states) keyword to specify which mode or modes are to be treated as hindered internal rotation. There should be one line in the TOR list for every mode treated as a hindered rotor. There are no defaults. If VANHAR or VRANGE specifies that a mode is to be treated as a hindered rotor, the line for that mode must be present in the TOR list. The input required for each hindered rotator mode is:

```
MODE P SIGMA(1) SCHEME LEVEL N
```

where

MODE	=	mode number m
P	=	number P of distinct minima along the torsional coordinate
SIGMA(1)	=	symmetry number σ_1 of lowest-energy minimum
SCHEME	=	one of five schemes defined in Ref. 52 of Section 21: OW Scheme ωW RO Scheme $R\omega$ CO Scheme $C\omega$ RW Scheme RW CW Scheme CW
LEVEL	=	one of the three levels defined in Ref. 52 of Sect. 21: FULL full SF single-frequency SC single-conformer
N	=	number N of atoms in first subgroup

In the five distinct schemes, R represents using the rectilinear model to calculate the moments of inertia of the torsional motion (see refs. 24, 29, and 30 in Sect. 21 for detailed description of the rectilinear model), and C represents using the curvilinear model to calculate the moments of inertia of the torsional motion (Ref. 59 in Sect. 20 gives a detailed description of the curvilinear model). Compared with the rectilinear model, for which one only needs to indicate which atoms are in each of the two fragments of the hindered rotation, the curvilinear model requires this information and also a choice of internal rotation axis (by using the NBOND keyword) and an assignment (by using the ISB keyword) of the lighter fragment as the rotational top.

Note that ISB is not used if SCHEME = 1.

Example:

```
TOR
    10  1  3  RO  SC  4
END
```

TOROPT

TOROPT is a list keyword that must be given for modes treated as hindered rotors if the default values of certain parameters are not accepted. Each record of TOROPT is a single line of the form:

MODE DATATYPE list of values

An END line signifies the end of all records in the list.

The quantities that may appear in the list if values depend on DATATYPE. The allowed values for DATATYPE and the associated lists are as follows:

<u>DATATYPE</u>	<u>list</u>
OMEGA	$\omega_2, \dots, \omega_P$ (the frequencies in cm^{-1})
U	U_2, U_3, \dots, U_P (excitation energies in cm^{-1})
I	I_2, I_3, \dots, I_P (moments of inertia in $\text{m}_e a_0^2$)
SIGMA	$\sigma_2, \sigma_3, \dots, \sigma_P$ (symmetry numbers, which are unitless integers)
W	W_1, W_2, \dots, W_P (barrier heights in cm^{-1})
ISB	i_1, i_2, \dots, i_N (the indices of the atoms in the first subgroup)
NBOND	indices of the two atoms that define the rotation axis (used only when the curvilinear option is used to calculate the moment(s) of inertia)

If scheme = CO or C ω , then the sum of the masses of the atoms in ISB must be less than the sum of the masses of the atoms not in ISB. Note that the values of P and N that appear in these definitions are specified using the TOR keyword. In the current version of POLYRATE, the values of OMEGA, U, and W are assumed to be independent of the value of the reaction coordinate s . That is, only values for the stationary points can be read in, and for the non-stationary points along the reaction path, the only hindered rotor parameters that can depend on s are the value of ω_1 , which is obtained directly from generalized normal mode analysis if the OW, RO, or CO scheme is used, and from the moments of inertia of the lowest minimum along the torsion mode, which are computed from the geometry if the CO or CW scheme is used and from the geometry and normal mode eigenvector if the RO or TW schemes is used.

The set of data types expected in the TOROPT list for a given mode depend on the choices specified in the TOR keyword list. In particular, the expected records are:

Choices in TORDATATYPES expected in TOROPT

<u>P</u>	<u>LEVEL</u>	<u>SCHEME</u>	
1	any	OW	W (W_1 only)
1	any	RO	ISB
1	any	CO	ISB, NBOND
1	any	RW	ISB, W (W_1 only)
1	any	CW	ISB, NBOND, W (W_1 only)
>1	FULL	OW	SIGMA, U, OMEGA, W
>1	FULL	RO	SIGMA, U, I, ISB, OMEGA
>1	FULL	CO	SIGMA, U, I, ISB, NBOND, OMEGA
>1	FULL	RW	SIGMA, U, I, ISB, W
>1	FULL	CW	SIGMA, U, I, ISB, NBOND, W
>1	SF	OW	SIGMA, U, W
>1	SF	RO	SIGMA, U, I, ISB
>1	SF	CO	SIGMA, U, I, ISB, NBOND
>1	SF	RW	SIGMA, U, I, ISB, W (W_1 only)
>1	SF	CW	SIGMA, U, I, ISB, NBOND, W (W_1 only)
>1	SC	OW	SIGMA, W (W_1 only)
>1	SC	RO	SIGMA, ISB
>1	SC	CO	SIGMA, ISB, NBOND
>1	SC	RW	SIGMA, ISB, W (W_1 only)
>1	SC	CW	SIGMA, ISB, NBOND, W (W_1 only)

If values are not expected but are given anyway, then these values will be ignored. The program will not check the correspondence of the TOR and TOROPT keyword options; it is users' responsibility to check if the calculation is meaningful. The defaults are $\omega_j =$ frequency obtained from normal mode analysis at point on reaction path ($j = 1, 2, \dots, P$), $W_j = 1000 \text{ cm}^{-1}$ ($j = 1, 2, \dots, P$), $U_j = 0$, $\sigma_j = 1$ ($j = 2, 3, \dots, P$), ISBN = {1 2 3 4}, and NBOND = {4 5}.

Example for using OW:

```
TOR
  17  2  1  OW          FULL  4
END
TOROPT
  17  SIGMA  2
  17  U      1000.
  17  ISB    1 2 3 4
  17  OMEGA  110.
END
```

Example for using CW:

```
TOR
  18  1  1  CW  SC  3
END
TOROPT
  18  ISB    3 4 5
  18  NBOND  2  6
  18  W      1450.0
END
```

VANHAR

VANHAR is a list keyword that is used to specify variable anharmonicity, i.e., that each vibrational mode is to be treated differently. The default are that all modes be treated harmonically. This keyword is used to specify those modes that are to be treated anharmonically. The valid choices are *harmonic*, *morse*, *morseqq*, *qqwkb*, *qqsemi*, and *tor*. (Note that modes for which the harmonic option is wanted need not be included in the list since it is the default.) Note that for each of these anharmonicity options there may be other variables that have to be set, and in each case there is keyword with the same name as the choices just given, i.e., *harmonic*, *morse*, *morseqq*, *qqwkb*, *qqsemi*, and *tor*. In specifying the modes, note that they are always numbered in order of decreasing frequency, regardless of symmetry, and that degenerate modes get two numbers.

The input required is:

```
mode          anharmonicity method
```

Example:

```
VANHAR
      3      tor
      5      qqsemi
END
```

VIB

If the option of the STATUS keyword is 6, frequencies of the stationary point should be read. VIB is a list keyword that is used to specify the frequencies (a.u. or cm^{-1} as determined by FREQUIT) of a reactant, product, or saddle point. The number of frequencies to be read is always F , where F is $3N - 6$ for nonlinear stationary points, $3N - 5$ for linear stationary points, and $3N$ for solid-state stationary points. For reactants and products there should be F real frequencies, and they should be given in decreasing order of their magnitude. For saddle points, the $F - 1$ real frequencies should be given first in decreasing order followed by the imaginary frequency given last as a negative number.

Example:

```
VIB
 1.72881D-02 1.72881D-02 1.62501D-02 8.24925D-03
 8.07930D-03 8.07884D-03 7.06232D-03 7.06187D-03
 6.58527D-03 3.28102D-03 3.28102D-03 -1.24729D-02
END
```

VRANGE

VRANGE is a list keyword similar to VANHAR except that a range of s values can also be specified for each mode. This means that except for the given modes and their respective ranges of s , each mode will be treated harmonically. Again the valid anharmonicity choices are: *harmonic*, *morse*, *morseqq*, *qqwk*, *qqsemi*, and *tor*. Note that for each of these anharmonicity options there may be other variables that have to be set. For more information, please see the keyword glossary entry for each method. This keyword should only be used in the START section.

The input required is:

```
smin          smax          mode number    anharmonicity
```

The example below shows the VRANGE input if one wanted to use the Morse I approximation in the regions $0.01 \leq s \leq 0.5$ and $-0.15 \leq s \leq -0.01$ for mode #2 and the harmonic approximation everywhere else.

Example:

```
VRANGE
 -0.15          -0.01          2      morsei
 0.01           0.50           2      morsei
END
```

WKB

WKB is a list keyword that is used if the zero point eigenvalues are to be computed using the WKB method with the true potential. See Sections 10.B and 11.B.2. The WKB keyword applies only to the ground state; another anharmonicity method must be chosen for the excited states, or else the excited states will be calculated harmonically. The options in this list control the WKB calculation. They are defined as follows:

<u>Option</u>	<u>Description</u>	<u>Default</u>
<i>wkbtol</i>	tolerance for finding WKB eigenvalues in hartrees	1×10^{-8}
<i>kbquad</i>	number of quadrature points used in calculating WKB phase integrals	40
<i>kbprint</i>	switch for printing values of turning points	off

Example:

```
WKB
  wkbtol      0.0000002
  kbquad      30
  kbprint
END
```

11.A.7. PATH section

The PATH section contains the keywords that control the calculation of the minimum energy path and the properties along the path. This section is not used in a restart calculation. For simplicity the keywords have been broken into five categories: basic, integrators, step sizes for the MEP, extrapolation, and print. This is only for clarity in the manual. The keywords in any of these five categories can be used in any order. Also for simplicity the keywords in each category have been labeled as either common or other. The keywords labeled common are those that might be expected to be found reasonably frequently in input files for typical POLYRATE runs. This labeling is only intended as an aid to the novice user. Again an alphabetical glossary of all the keywords follows the five summary tables.

If only a conventional transition state theory rate constant calculation is desired, one needs to include this section only if the Hessians are input in mass-scaled coordinates and SCALEMASS does not equal 1.0 amu.

Basic:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<i><u>Common:</u></i>			
CARTRP	Switch	on	Whether the normal mode analysis for reactants and products is carried out in cartesian coordinate
COORD	Variable	<i>cart</i>	type of coordinate system to be used for computing vibrational frequencies and generalized normal modes along the MEP
IDIRECT	Variable	<i>1</i>	initial direction off of saddle pt.
INTDEF	List	not used	define curvilinear internal coordinates
NSTEPS	Variable	<i>99999</i>	max no. of steps along MEP
SADDLE	Switch	yes	whether the reaction has a saddle point
SCALEMASS	Variable	<i>1.0</i>	mass-scaled coordinates scale factor
SIGN	Variable	<i>reactant</i>	direction of unbound eigenvector
SRANGE	List	not used	range of <i>s</i> for MEP
SLP	Variable	<i>+1.0</i>	upper limit on <i>s</i> before it is extended by interpolation
SLM	Variable	<i>-1.0</i>	lower limit on <i>s</i> before it is extended by interpolation
<i><u>Others:</u></i>			
CURV	Variable	<i>dgrad</i>	method for computing curvature
DLX3	Variable	<i>1 × 10⁻⁴</i>	step size for 3 rd derivatives of PES
FCSCALE	List	not used	the force constant scaling factors for specified internal coordinate
FIRSTSTEP	Variable	<i>nmode</i>	method for first step
FREQSCALE	Variable	<i>1</i>	Scale factor for frequencies
HESS	Switch	on	Compute the Hessian along MEP

Basic:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<i>Others:</i>			
INTMU	Variable	1	Number of points on each side of the saddle point for interpolation of $\mu_{\text{eff}}(s)$
ISWR	Variable	Not used	mode index of the reactant frequency which is to be scaled
ISWP	Variable	not used	mode index of the product frequency which is to be scaled
IVTSTMOPT	List	not used	IVTST-M options
LBEXP	Variable	3	mapping for curvature components
POLYS	switch	on	include 10 extra points for spline fitting
SINCW	variable	0.01	step size for 10 extra points
IVTST0FREQ	list	not used	modifies the frequency of a specified mode along the reaction path
NEXTPT	switch	off	only compute next point on MEP
REORDER	switch	off	order frequencies along MEP by overlaps?
SCLPT	switch	off	whether scale the potential if FCSCLAE is used
SPECSTOP	List	not used	special stop for MEP
CURVE	Variable	vag	the curve on which the check for the special stop is made
POINT	Variable	savegrid	the grid on which the end point for the special stop is chosen
PERCENTDOWN	Variable	50.	the percentage down from the saddle point for the special stop

Basic:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<i>Others:</i>			
SSPECIAL	list	no special s val.	special s values for N.M. analysis
SYMMETRY	switch	off	symmetric reaction path
VSCALE	variable	1	scale factor for MEP

Integrators:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<u>Common:</u>			
ES1OPT	list	not used	Euler with stabilization
DELTA2	variable	5×10^{-5}	step size δ_2 for stabilization
DIFFD	variable	1×10^{-8}	magnitude of difference vector
RPM	variable	<i>ESD</i>	reaction path following method
<u>Others:</u>			
RODS	switch	off	re-orient the dividing surface

Step sizes for the MEP:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<u>Common:</u>			
SSTEP	variable	5×10^{-4}	step size for computing MEP
<u>Others:</u>			
INH	variable	9	ratio of gradients to Hessians
INI	variable	1	interval of gradient interpolation step
SFIRST	list	not used	special step size for first n steps
NFSTEP	variable	0	number of initial special steps
FSIZE	variable	1×10^{-5}	step size for the special steps

Extrapolation:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<i>Others:</i>			
EXFIRST	list	not used	extrapolate first direction from s.p.
EXNSTEP	variable	0	number of steps
EXREACT	switch	off	extrapolation is towards reactants
EXPROD	switch	on	extrapolation is towards products
EXSTEP	variable	0.03	step size
EXALPHA	variable	1.0	extrapolation range parameter
EXSECOND	list	not used	extrapolate second direc. from s.p.
EXNSTEP	variable	0	number of steps
EXREACT	switch	on	extrapolation is towards reactants
EXPROD	switch	off	extrapolation is towards products
EXSTEP	variable	0.03	step size
EXALPHA	variable	1.0	extrapolation range parameter

Note: The extrapolation is not performed unless either or both the EXFIRST and EXSECOND keywords are specified.

Print:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<i>Others:</i>			
POTINF	variable	0	print <i>n</i> pieces of PES information
PRPATH	list	not used	print reaction-path information to units fu25-fu27
PRINTSTEP	switch	off	print geom. at each save point
PRDISTMX	variable	20	print interatomic distance matrices
PRSAVERP	variable	9999	print geom., <i>V</i> & <i>V'</i> at each save pt
PRSAVEMODE	variable	9999	print normal mode information at each save point
SDEBG1	variable	not used	<i>s</i> value for starting debug printing of generalized normal mode vibrational analysis along the MEP
SDEBG2	variable	not used	<i>s</i> value for ending debug printing of generalized normal mode vibrational analysis along the MEP
SPRNT	variable	0	extra RPH info. printed at large <i>s</i>
SSPECPR	switch	off	print info. at special <i>s</i> values

Nonequilibrium Solvation:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
BATH	list	not used	nonequilibrium solvation
BATHTEMP	variable	298.0	temperature of the solvent in K
CUPLCNST	list	not used	coupling constants in mass-scaled atomic units (A)
DIFFUSION	list	not used	diffusion coefficients in $10^{-5} \text{ cm}^2 \text{ s}^{-1}$
FRICTION	variable	0	solvation time in fs

Glossary of PATH keywords

BATH

BATH is a list keyword specifying the parameters used for a nonequilibrium solvation calculation. For detailed description, see Ref. 53 in Sect. 21. There are 4 options:

<u>Keyword</u>	<u>Description</u>
BATHTEMP	temperature of the solvent in K
CUPLCNST	atomic coupling constants of the solvent coordinate to the solute in mass-scaled atomic units
DIFFUSION	effective atomic diffusion constants in $10^{-5} \text{ cm}^2 \text{ s}^{-1}$
FRICION	solvent friction time in fs (Note: the friction time is also called the solvation time)

Example:

```
BATH
  FRICION  10.0    # in fs
  DIFFUSION          # in  $10^{-5} \text{ cm}^2 \text{ s}^{-1}$ 
    1    8.08
    2   12.62
    3    1.72
    4    3.04
    5   33.08
    6  192.27
    7   50.52
  END
  BATHTEMP 298.0
END
```

CARTRP (NOCARTRP)

The keyword CARTRP is used to declare explicitly that the reactant and product normal mode analysis is carried out in Cartesian coordinates. The purpose of adding this keyword is to allow the use of curvilinear coordinate normal mode analysis. In versions of POLYRATE earlier than 8.6, the normal mode analysis for reactants and products was always carried out in Cartesian coordinates even though the normal mode analysis for transition states can be carried out in curvilinear internal coordinates. The default is CARTRP.

Note: CARTRP should not be used when scaling factors for principal force constants in curvilinear internal coordinates (FCSCALE) are used.

COORD

COORD is a variable keyword specifying the type of coordinate system for calculating the bound-state vibrational frequencies and normal mode eigenvectors along the MEP. See Sections 6.G.1 and 10.M. The default is *cart* (Cartesian coordinates). The valid options are:

<u>Option</u>	<u>Definition</u>
<i>curv1</i>	use curvilinear internal coordinates for generalized normal mode analyses using method of Ref. 41 of Sect. 21
<i>curv2</i>	use curvilinear internal coordinates for generalized normal mode analyses using method of Ref. 43 of Sect. 21
<i>curv3</i>	use redundant internal coordinates for generalized normal mode analyses using method of Ref. 46 of Sect. 21
<i>cart</i>	use rectilinear, i.e., Cartesian coordinates for generalized normal mode analyses

Note that if the *curv1*, *curv2*, or *curv3* option is used, the INTDEF keyword discussed below is required.

Example:
 COORD *cart*

CURV

CURV is a variable keyword used to specify the method for computing the reaction path curvature components. See Sections 6.H and 10.F for more information about this issue. The default is *dgrad*. The valid options are:

<u>Option</u>	<u>Definition</u>
<i>dgrad</i>	use a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate
<i>oneside</i>	use one-sided differences for the derivative of the gradient with respect to the reaction coordinate
<i>dhess</i>	use Hessian, generalized-normal mode eigenvectors, and gradient to calculate the reaction path curvature

Example:
 CURV *dgrad*

DLX3

DLX3 is a variable keyword for specifying the step size in unscaled bohrs used for numerical differentiation in unscaled atomic Cartesians to obtain the 3rd derivatives of *V* from the Hessian at the saddle point. These derivatives are required when the cubic starting algorithm is employed (see FIRSTSTEP). The default is $1 \times 10^{-4} a_0$.

Example:
 DLX3 0.00005

ES1OPT

ES1OPT is a list keyword for using the Euler with stabilization integrator. The user must give the step size (in bohr) for the finite-difference derivative (DELTA2 keyword) and the tolerance parameter for the magnitude (in a.u.) of the difference vector between successive normalized gradients (DIFFD keyword). See Section 10.D. The defaults are $\text{DELTA2} = 5 \times 10^{-5} a_0$ and $\text{DIFFD} = 1 \times 10^{-8}$.

Example:

```
ES1OPT
  DELTA2      0.00001
  DIFFD       0.000001
END
```

EXFIRST

EXFIRST is a list keyword for specifying the extrapolation beyond the limits (SLM and SLP) of the Hessian grid for the first direction from the saddle point when using the hooks. (See Section 7.B for a discussion of which direction is first.) The EXFIRST and EXSECOND keywords cannot be used with units fu30, fu31, or fu40. The extrapolation procedure is different for reactions with and without a saddle point. The particular extrapolation procedure to be used is chosen based on the SADDLE(NOSADDLE) keyword in the PATH section.

For reactions without a saddle point (NOSADDLE specified in the PATH section) the extrapolation procedure is described in Section 10.A; it is designed to extrapolate $V_a^G(s)$ and $\mu_{\text{eff}}(s)$ toward bimolecular reactants or products and should not be used to extrapolate toward unimolecular reactants or products. If EXFIRST is omitted, no extrapolation is performed in the first direction, that is, by default no extrapolation is performed. If no extrapolation is performed, V_{MEP} is set equal to the reactant or product value beyond the limits, and μ_{eff} is set to μ .

For reactions with a saddle point (SADDLE is specified in the PATH section; or SADDLE is accepted as the default) the extrapolation procedure is based on the mapped interpolation algorithm, which is described in Sections 6.N.3 and 10.J and in Ref. 48 of Sect. 21. The mapped interpolation algorithm is used to extrapolate V_{MEP} , $|\mathbf{I}|$, ω , and B_{mF} . If EXFIRST is omitted, no extrapolation is performed in the first direction, that is by default no extrapolation is performed.

The EXALPHA keyword is only used for reactions without a saddle point.

If the EXFIRST keyword is not specified, then no extrapolation is performed. If the EXFIRST keyword is specified with an empty list then the extrapolation is performed with the default values listed below. The keywords in this list and their respective default values are explained below.

<u>Keyword</u>	<u>Description</u>	<u>Default</u>
EXREACT	extrapolation is towards reactants	off
EXPROD	extrapolation is towards products	on
EXNSTEP	number of steps to be taken in extrapolation region	0
EXSTEP	mass-scaled step size in bohr in the extrapolation region for computing the adiabatic potential and other quantities needed for the tunneling calculations	0.03
EXALPHA	exponential range parameter α in reciprocal bohrs (only used with NOSADDLE)	1.0

Example:

```
EXFIRST
      EXPROD
      EXNSTEP      50
      EXSTEP       0.1
      EXALPHA      0.5
END
```

Example (empty list, defaults only):

```
EXFIRST
END
```

EXSECOND

EXSECOND is a list keyword that is very similar to EXFIRST but is used for specifying the extrapolation beyond the limits (SLM and SLP) of the Hessian grid for the second direction from the saddle point when using the hooks. (See Section 7.B for a discussion of which direction is first.) The EXFIRST and EXSECOND keyword can not be used with units fu30, fu31, or fu40. See also the discussion of EXFIRST. The EXALPHA keyword is only used when NOSADDLE is specified in the PATH section. If the EXSECOND keyword is not specified then no extrapolation is performed. If the EXSECOND keyword is specified with an empty list then the extrapolation is performed with the default values listed below. The valid keyword options are defined below.

<u>Option</u>	<u>Description</u>	<u>Default</u>
EXREACT	extrapolation is towards reactants	on
EXPROD	extrapolation is towards products	off
EXNSTEP	number of steps to be taken in extrapolation region	0
EXSTEP	mass-scaled step size in bohr in the extrapolation region for computing the adiabatic potential and other quantities needed for the tunneling calculations	0.03
EXALPHA	exponential range parameter α in reciprocal bohrs	1.0

Example:

```
EXSECOND
  EXREACT
  EXNSTEP    50
  EXSTEP     0.1
  EXALPHA    0.5
END
```

Example (empty list, defaults only):

```
EXSECOND
END
```

FCSCALE

FCSCALE is a list keyword used to specify scaling factors for principal force constants of the selected curvilinear internal coordinates defined in the INTDEF keyword. This keyword should be used only when internal coordinates defined by COORD *curv2* or COORD *curv3* options are used, and the switch keyword **FREQ** is on. To specify the selected curvilinear internal coordinates involving force constant scaling, the connectivities of atoms are required to be defined using the INTDEF keyword as well as the FCSCALE keyword is used. Each item in the list is in the form of an integer sequence number followed by a floating-point scaling factor. The sequence number denotes the ordinal number of the specific coordinate in the list of all the curvilinear internal coordinates defined under the INTDEF keyword. (Note: This is not the ordinal number in internal storage after the code internally re-orders the coordinates.) The corresponding diagonal element in the force constant matrix will be scaled by the scaling factor. All scaling factors are set to 1.0000 as the default. In the following example, the sequence number 8 in FCSCALE represents the stretch between atom 1 and atom 2; thus the

diagonal element in the force constant matrix involving that stretch will be scaled by a scaling factor of 0.9734.

Example:

```
INTDEF
  1=2=3
  2-3-4 2-3-5 2-3-6 4-3-5 4-3-6 5-3-6
  1-2 2-3 3-4 3-5 3-6
  4-3-5-6
END

FCSCALE
  8 0.9734
END
```

When FCSCALE is used, NOCARTRP should be declared explicitly because the reactant and product normal mode analysis should be carried out in curvilinear coordinates.

The actual effect of the force constant scaling is controlled by the keyword SCLPT, in which either the scaling of the potential or the scaling of frequencies is specified. The default is scaling the frequencies only (NOSCLPT).

Note: Another different way to scale frequencies is with FREQSCALE, in which all frequencies are scaled by the same scaling factors.

FIRSTSTEP

FIRSTSTEP is a variable keyword specifying the method for taking the first step off the saddle point or starting geometry. The default is *nmode*. The valid arguments are:

<u>Option</u>	<u>Definition</u>
<i>nmode</i>	take the first step along the imaginary-frequency normal mode
<i>cubic</i>	use the cubic starting algorithm (see STEP 3)
<i>gradi</i>	take the first step along the negative of the gradient direction [use this only if NOSADDLE is selected; see Section 6.G]

Example:

```
FIRSTSTEP  cubic
```

FREQSCALE

FREQSCALE is a variable keyword for specifying the factor by which to scale the harmonic vibrational frequencies for partition function calculations. If FREQSCALE is used, the scaling will be applied to all stationary point frequencies and generalized normal mode frequencies for generalized transition states along the reaction path. Furthermore, the vibrationally adiabatic potential energy curve V_a^G will be calculated by adding the scaled ZPE to V_{MEP} and all tunneling calculations will be based on this new V_a^G . The default is not to scale the frequencies.

Example:
FREQSCALE 0.91

HESS (NOHESS)

HESS is a switch keyword that determines if the Hessian should be computed along the MEP. This switch is defaulted to on and in most cases one would want to compute the Hessian. Most likely this switch will only be used for a special case when the Hessian is not desired, and thus NOHESS will be used. The default is HESS.

IDIRECT

IDIRECT is a variable keyword used to specify the direction of the initial step from the starting point. The allowed arguments for this variable are ± 1 : +1 if the initial direction is the direction toward products and -1 if it is towards reactants. See Section 7.B. The default is 1.

Example:
IDIRECT 1

INH

INH is a variable keyword used to specify the ratio of the number of gradients to the number of Hessians calculated along the MEP. INH must be an integer value. The default is 9.

Example:
INH 4

INI

INI is a variable keyword used to specify the interval of the gradient interpolation steps in the IVTST-M calculation. INI must be an integer value less than or equal to INH. The default is 1.

Example:
INI 2

INTDEF

INTDEF is a list keyword used to specify the curvilinear internal coordinates used for the vibrational frequencies and calculations of generalized normal modes. The allowed types are stretches ($i-j$), bends ($i-j-k$), dihedrals ($i-j-k-l$), and doubly degenerate linear bends ($i=j=k$) where i , j , k , and l are the atom-number-labels defined in the GENERAL section. Note: when the bond angle is close to 180 degrees (i.e., greater than about 175 degrees), using the doubly degenerate linear bending mode approximation will prevent dividing by zero or a number close to zero during computations.

Example:

```
INTDEF
    1-2 1-3 3-4 2-1-3 1-3-4 2-1-3-4
END
```

Note: The pound sign (#) can not be used as a comment card within the INTDEF keyword list.

INTMU

INTMU is a variable keyword that indicates the number of points on each side of the saddle point used for the interpolation of the effective reduced mass at the saddle point. See Section 6.H. The default value is 1. The valid options are:

<u>Option</u>	<u>Definition</u>
1	1 point on either side of the saddle point; thus interpolation is based on 2 points
3	3 points on either side of the saddle point; thus interpolation is based on 6 points

Example:

```
INTMU    1
```

ISWP

ISWP is a variable keyword for specifying the index of the product frequency that will be scaled by a factor of (VSCALE)^{1/2}. This keyword is only needed if the VSCALE keyword is specified and there is only one product species.

Example:

```
ISWP    9
```

ISWR

ISWR is a variable keyword for specifying the index of the reactant frequency that will be scaled by a factor of (VSCALE)^{1/2}. This keyword is only needed if the VSCALE keyword is specified and there is only one reactant species.

Example:

```
ISWR    9
```

IVTSTMOPT

IVTSTMOPT is a list keyword for specifying the IVTST-M options for the curvature components and unimolecular reactions.

The mapping procedure for the curvature components is slightly different from the mapping for the rest of the parameters. The exponent in equation 19 of Ref. 48 in Sect. 21 has been optimized in order to make the changes of the interpolated curvature components along the reaction path as close as possible to the changes observed in converged calculations. Therefore, the user is strongly advised against changing this parameter without a good reason. *If* one does have a good reason, though, the exponent can be set by means of the LBEXP variable option. This keyword can only take positive odd integer values.

There are two methods for performing IVTST-M calculations for the unimolecular side (or sides) of reactions with one reactant or/and one product, or wells in one of the channels: (i) to carry out a spline-under-tension fit (as a function of z) based only on the input data, or (ii) to carry out a spline-under-tension fit based on ten extra energy points. The ten extra points are calculated by means of a cubic polynomial fitted to the unimolecular reactant, unimolecular product, or well, a point calculated from the lowest real frequency at that minimum, and the two points on the reaction path with the largest $|s|$ values on the unimolecular side. In the second option generating the extra point from the frequency requires the choice of a small step-size, whose value is recommended to be the same as the gradient step-size used in the calculation. The *polys* option specifies the treatment to be given to the unimolecular side (or sides), and, the *sincw* option specifies the value of the step-size. The default is to use a cubic polynomial for estimating 10 extra energies, with a step-size along the lowest frequency mode of $0.01 a_0$ in s , which is mass scaled.

<u>Option</u>	<u>Definition</u>
<i>lbexp</i>	value of the exponent for interpolating the curvature components. The default is 3.
<i>polys</i>	use a cubic polynomial for estimating 10 energy points before spline-fitting $V_{\text{MEP}}(s)$ (default).
<i>nopolys</i>	do not include 10 extra points, and base the spline fit on input data only plus one extra computed as described in Section 10.J.
<i>sincw</i>	variable keyword to indicate the step-size (in mass-scaled coordinate space in atomic units) along the lowest-frequency mode to fit the cubic polynomial. The default value is $0.01 a_0$. this value is used only if <i>polys</i> is selected.

Example:

```
IVTSTMOPT
  lbexp      5
  polys
  sincw     0.005
END
```

IVTST0FREQ

IVTST0FREQ is a list keyword that modifies the frequency of a particular mode along the reaction path. A section for each mode begins with the list option keyword MODE which contains lines with the following keyword options:

<u>Option</u>	<u>Definition</u>
<i>index</i>	index of the vibrational mode to be interpolated (this is determined from the frequencies of the saddle point arranged in decreasing order).
<i>reactant x</i>	used to indicate the frequency (in cm^{-1}) at the reactant side ($s < 0$) for the interpolation with value x . When the reactant is unimolecular or a reactant well present, the s value has to be provided with option SR. If a reactant well exists for a bimolecular reaction, the interpolation uses the frequency at the well instead of the reactants.
<i>start x</i>	used to indicate the frequency at $s = 0$ for the interpolation.
<i>product x</i>	used to indicate the frequency at the product side ($s > 0$) for the interpolation with value x . When the product is unimolecular or a product well present, the s value has to be provided with option SP. If product well exists for a bimolecular reaction, the interpolation uses the frequency at the well in stead of the products.
<i>sr x</i>	used to indicate the s value (in bohr) at the limit of reactant side ($s < 0$). The default is -1.0 bohr.
<i>sp x</i>	used to indicate the s value (in bohr) at the limit of product side ($s > 0$). The default is $+1.0$ bohr.

The IVTST0FREQ algorithm is similar to the frequency interpolation algorithm of the VTST-IC method. See Section 10.M for a detailed description.

Notice that IVTST0FREQ can be used to interpolate a given mode, say i , (where modes are numbered in order of decreasing frequency) from the saddle point to reactants and products. It cannot interpolate between mode i at the saddle point and mode $i + 1$ at reactants (which might sometimes be useful for unimolecular reactions). It cannot interpolate between mode i at the saddle point and mode i at an intermediate location between the saddle point and reactants.

Note that IVTST0FREQ is currently incompatible with using fu31.

Example:

For a bimolecular/bimolecular case

```
IVTST0FREQ
  MODE
    index      11
    reactant   0
    start      300
    product    0
  END
```

END

For a unimolecular/bimolecular case

```
IVTST0FREQ
```

```

MODE
  index      11
  reactant   50
  start      200
  product    0
  sr         -2.0
END
END

```

NEXTPT (NONEXTPT)

NEXTPT is a switch keyword that only works if `POTENTIAL = unit30` or `unit 40`. When turned on it will cause POLYRATE to compute only the next point on the MEP by taking a single step from the last point in the electronic structure input file, and then stop. The method used for the step will be ESD (default), or Page-McIver (if PAGEM is specified). The step size used in this procedure is specified using the SSTEP keyword. The direction in which one is following the path (towards reactants or products) is specified using the IDIRECT keyword in conjunction with NEXTPT (IDIRECT = 1 to calculate the next point to the right; IDIRECT = -1 to calculate the next point to the left). For example, using the ESD method, turning on NEXTPT causes the following calculation to be performed

$$\mathbf{x}_{\text{next-point}} = \mathbf{x}_{\text{current}} - \delta s \hat{\mathbf{g}}_{\text{current}}$$

$$s_{\text{next-point}} = s_{\text{current}} + (\text{IDIRECT}) \delta s$$

where $\mathbf{x}_{\text{current}}$ is the mass-scaled geometry corresponding to the last point input on fu30 or fu40, and $\hat{\mathbf{g}}_{\text{current}}$ is a unit vector in the direction of the mass-scaled gradient at that point. Thus, to go left, the last point read in should have negative s , and to go right, it should have positive s .

This option is useful when using POLYRATE as a driver to determine geometries for ab initio calculations. The only print option in unit fu30 compatible with the use of NEXTPT is `LOPT(1) = 0`. The default is NONEXTPT.

NSTEPS

NSTEPS is a variable keyword for specifying the maximum number of steps (after the first step) along the negative of the gradient of V to be taken in each direction from the saddle point or starting full-system geometry. The minimum number of steps is the number required to reach the limits specified by SRANGE. The default is 99999.

Example:

```

NSTEPS      5000

```

POTINF

POTINF is a variable keyword that specifies how many pieces of data from the potential energy subroutine are written along the MEP to FORTRAN unit fu6. The default is 0. See Subsection 8.C.1 for details.

Example:
 POTINF 4

PRDISTMX

PRDISTMX is a variable keyword that indicates the interval along the Hessian grid for the printing of interatomic distance matrices. The i - j element of the interatomic distance matrix is the distance (in mass-scaled bohr) between atoms i and j . The default is set to print the interatomic distance matrix at every 20th point along the Hessian grid.

Example:
 PRDISTMX 20

PRINTSTEP (NOPRINTSTEP)

PRINTSTEP is a switch keyword that specifies that the geometry at each step along the MEP be printed. The default is NOPRINTSTEP. See the SPRNT keyword below.

PRPATH

PRPATH is a list keyword used to set the options for printing reaction-path information to units fu25-fu27. This keyword is especially useful for analyzing reaction-path data. The unit fu25 and fu26 output can easily be imported to a spreadsheet, and unit fu27 output is designed to be read into the XMOL program to visualize the geometry changes along the reaction path. (The XMOL program is a X-windows-based molecule viewer and format converter developed by Wasikowski and Klemm at the Minnesota Supercomputer Center, Inc.) When this keyword is specified, the program will automatically write the energetic information (V_{MEP} and V_{a}^{G}) to unit fu25. The valid options are defined below.

<u>Option</u>	<u>Description</u>	<u>Default</u>
<i>coord</i> a_i	up to 4 atoms, from the full system, can be specified for which the bond distances(2, 3, or 4 atoms specified), the bond angles(3 or 4 atoms specified), and the dihedral angle(4 atoms specified) are calculated and written to unit fu28.	off
<i>interval</i> n	write the reaction-path information every n Hessian grid points.	1
<i>freq</i> m_i	GTS frequency mode numbers to be printed to unit fu26; a total of up to ten modes can be specified and printed.	not used
<i>xmol</i>	write XMOL XYZ input file to unit fu27.	off

Example:
 PRPATH
 COORD 1 2 3 6
 INTERVAL 5
 FREQ 1 2 3
 XMOL
 END

PRSAVEMODE

PRSAVEMODE is a variable keyword that specifies that (in addition to the items listed for PRSAVERP) the normal mode eigenvectors, energies, curvatures, and frequencies be printed at a given multiple of the save point. The default is 9999.

PRSAVERP

PRSAVERP is a variable keyword that specifies that the geometry, potential (V), curvatures, and gradient (V') at a given multiple of the save point along the MEP be printed. The default is 9999.

REORDER (NOREORDER)

REORDER is a switch keyword that specifies that the frequencies be ordered according to overlap calculations along the MEP. The default is NOREORDER, which means the frequencies are in canonical order. See Sections 6.E, 6.F, and 6.G for further discussion.

RODS (NORODS)

RODS is a switch keyword which specifies that the RODS algorithm is to be used to re-orient the generalized transition state theory dividing surface. The RODS algorithm can be used with any of the integrators, ESD, ES1, or PAGEM. See Sect. 10.D. The default is NORODS, which turns RODS off. When RODS is off, the generalized-transition-state-theory dividing surface is normal to the gradient.

RPM

RPM is a variable keyword that indicates which integrator is to be used for following the reaction path. The default is *esd*. The valid options are defined below.

<u>Option</u>	<u>Description</u>
<i>esd</i>	use the Euler steepest-descents integrator to follow the reaction path; this is the name we now use for the Euler algorithm, which is often called the Euler single-step algorithm in textbooks
<i>es1</i>	use the Euler-with-stabilization integrator to follow the reaction path
<i>pagem</i>	use the Page-McIver integrator to follow the reaction path
<i>vrpe</i>	use the VRP(ESD) algorithm to follow the reaction path

Example:

RPM *pagem*

SADDLE (NOSADDLE)

SADDLE is a switch keyword to indicate whether the reaction has a saddle point or not. See Sect. 6.G. The default value is SADDLE.

SCALEMASS

SCALEMASS is a variable keyword for the reduced mass μ in amu used in defining the mass-scaled coordinates. This reduced mass is called the scaling mass. See Refs. 1 and 5 of Sect. 21 for an explanation of the scaling mass. Notice that no physical observable depends on μ , but a single value must be used consistently in the whole calculation. Note also that certain nonobservable intermediate quantities, most notably the numerical value of s at any point along the MEP, *do* depend on μ . The default is 1.0 amu which means that the mass-scaled coordinates in bohrs have the same numerical values as mass-weighted coordinates in amu^{1/2} bohrs. (These are the mass-weighted coordinates made popular by Wilson, Decius, and Cross in Ref. 36 of Sect. 20.) The variable μ is discussed in Section 6.A of this manual.

Example:

```
SCALEMASS 0.195
```

SCLPT (NOSCLPT)

SCLPT is a switch keyword to specify the behavior of the force constant scaling option. (Note: Details of the force constant scaling option can be found in the description of the keyword FCSCALE).

If the keyword SCLPT is turned on along with the keyword FCSCALE, the potential surface is actually scaled. The scaled force constant matrix will be used to obtain both the frequencies and the normal mode eigenvectors in the normal mode analysis. Thus the MEP is altered when FCSCALE is used with those force-constant involved reaction-path following algorithms, such as VRP(ESD), Page-McIver, etc. Also, the first step on the reaction path will be changed as well if the normal mode eigenvector with the imaginary frequency is used to determine the direction of the first step.

However, if this switch is turned off using NOSCLPT, the force constant scaling is actually just a way to change some of the frequencies. After frequencies are obtained from the scaled force constant matrix, the un-scaled force constants are recovered. The normal mode eigenvectors all come from the un-scaled force constants. Users should be aware that the MEP is not altered using this strategy. The only quantity changed is the frequencies. Those scaled frequencies will be used in the calculations of ZPE, vibrational partition functions, and tunneling probabilities.

SDEBG1 SDEBG2

SDEBG1 and SDEBG2 are variable keywords used to specify the s limits (in bohr) of the interval on the MEP at which to have debug information of the vibrational frequencies and generalized normal-mode calculations output to unit fu6. (Note: the value of SDEBG1 has to be smaller than SDEBG2)

Example:

```
SDEBG1 0.499
SDEBG2 0.501
```

SFIRST

SFIRST is a list keyword that allows a special step size for the first n steps. The user must give the number of initial steps to be taken in each direction from the initial geometry, including the initial step (NFSTEP) and the step size in bohr for each of these steps (FSIZE). The defaults are NFSTEP = 0 and FSIZE = 1×10^{-5} .

Example:

```
SFIRST
      NFSTEP      10
      FSIZE       0.0001
END
```

SIGN

SIGN is a variable keyword used to ensure the conventional definition of the sign of s , $s < 0$ for the reactant side and $s > 0$ for the product side, is followed. The allowed arguments for this variable are *reactant* and *product*: *product* if the unbound eigenvector at the saddle point points toward the product side and *reactant* if the eigenvector points toward the reactant side. The SIGN keyword is discussed further in Section 7.B. The default is *reactant*.

Example:

```
SIGN product
```

SPECSTOP

SPECSTOP is a list keyword that may be used to specify the range over which the MEP is to be computed. This keyword can function in conjunction with the SRANGE keyword that is presented below. The SPECSTOP keyword may be used to limit the range over which the MEP is calculated to a smaller range than that specified by the SRANGE keyword. The MEP calculation will be stopped when the value of the potential energy (V_{MEP}) or the value of the vibrationally adiabatic ground-state potential energy (V_{aG}) becomes smaller than a critical value. This critical value is defined based on the values of potential energy or zero-point-inclusive potential energy at the saddle point and reactants or products. For example, if the V_{aG} curve is chosen, the MEP will be determined until the $V_{\text{aG}}(s)$ becomes smaller than the value calculated as: $[(1-x) V_{\text{aG}}(s=0) + x V_{\text{+ZPE}}(\text{R or P})]$ where $V_{\text{+ZPE}}(\text{R or P})$ is the higher of the zero-point-inclusive potential energy for the reactants and products, and x is equal to $X/100$ (X is the argument of PERCENTDOWN subkeyword). Note that $V_{\text{+ZPE}}(\text{R or P})$ is equivalent to $V_{\text{aG}}(\text{R or P})$ for a bimolecular reaction but not for an unimolecular reaction for which $V_{\text{aG}}(\text{R or P})$ does not include the zero-point-energy contribution of the reaction coordinate motion. Similarly, if the V_{MEP} curve is chosen, the MEP will be calculated until the $V_{\text{MEP}}(s)$ becomes smaller than the value calculated as: $[(1-x) V_{\text{MEP}}(s=0) + x V_{\text{MEP}}(\text{R or P})]$ where $V_{\text{MEP}}(\text{R or P})$ is the higher of the zero-point-exclusive potential energy for the reactants and products, and x is again equal to $X/100$ (X is the argument of PERCENTDOWN subkeyword). Finally, the MEP limits can be chosen either on the fine grid (also called gradient grid) or the save grid. In the current version of the code, this keyword cannot be used if the calculation requires the use of a restart file (fu1, fu2, or fu3) or the use of an electronic structure input file (fu29, fu30, fu31, or fu40). The valid keyword options are defined below.

<u>Option</u>	<u>Description</u>	<u>Default*</u>
CURVE	the curve on which comparison is made (the options are <i>vmep</i> and <i>vag</i>)	<i>vag</i>
POINT	the grid on which the end point is chosen (the options are <i>savegrid</i> and <i>finegrid</i>)	<i>savegrid</i>
PERCENTDOWN	percent down from the saddle point	50.

Example:

```
SPECSTOP
      CURVE          vmep
      POINT          finegrid
      PERCENTDOWN    63.4
END
```

Example (empty list, defaults only):

```
SPECSTOP
END
```

* These are the defaults if SPECSTOP is specified. If it is not specified, then PERCENTDOWN is effectively infinity, i.e., only SRANGE is used to stop the reaction path calculation.

SPRNT

SPRNT is a variable keyword that, if PRINTSTEP is on, requires that $|s| > \text{SPRNT}$ for any reaction-path information to be printed. Default is 0. The unit is bohr.

Example:

```
SPRNT  0.1
```

SRANGE

SRANGE is a list keyword needed to specify the limits (in bohr) on the reaction coordinate, s (in bohr) for which the MEP is to be computed. These limits, SLM and SLP, are the ranges over which the MEP is computed, i.e., $\text{SLM} \leq s \leq \text{SLP}$. The default values are $\text{SLM} = -1.0$ and $\text{SLP} = +1.0$.

Example:

```
SRANGE
  SLP  1.50
  SLM -1.75
END
```

SSPECIAL

SSPECIAL is a list keyword used to give all the special s values at which normal mode analyses are to be performed. These values of s (in bohr) will be included in the Hessian grid. The default is to not include any special s values. See keyword SSPECPR below.

Example:

```
SSPECIAL
  0.14
  0.34
 -0.76
END
```

SSPECPR

SSPECPR is a switch keyword that works in conjunction with SSPECIAL. It turns on the printing of generalized normal mode vibrational energies, frequencies, and eigenvectors for the special s values regardless of whatever other print keywords are specified. The default is off.

SSTEP

SSTEP is a variable keyword that specifies the step size in bohr along the mass-scaled MEP for following the gradient of V , except that the first n steps can be different (see SFIRST). The default is $5 \times 10^{-4} a_0$.

Example:

```
SSTEP  0.0001
```

Note that SSTEP is ignored if POTENTIAL = *unit30*, *unit31*, or *unit40* is specified in the GENERAL Section. See Section 9.C.

SYMMETRY (NOSYMMETRY)

SYMMETRY is a switch keyword that specifies that the reaction path is symmetric. When this is turned on only one direction of the MEP from the saddle point is calculated, and the MEP information from the other direction is obtained by reflection. Default is NOSYMMETRY.

VSCALE

VSCALE is a variable keyword for specifying the factor by which to scale the potential energy along the MEP. If VSCALE is 0 or 1, no scaling is done. The default is 1.

Example:

```
VSCALE 1
```

11.A.8. TUNNEL section

The TUNNEL section contains the keywords that control the tunneling calculations. This section is not needed if tunneling calculations are not desired. If this section is not included then all defaults are ignored, and the default SCT, ZCT, and Wigner calculations are not performed. For simplicity the keywords have been broken into two categories: basic and large-curvature. This is only for clarity in the manual. The keywords in any of these categories can be used in any order. Also for simplicity the keywords in each category have been labeled as either *common* or *other*. The keywords labeled *common* are those that might be expected to be found reasonably frequently in input files for typical POLYRATE runs. This labeling is only intended as an aid to the novice user. Again an alphabetical glossary of all the keywords follows the two summary tables.

Basic:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<i><u>Common:</u></i>			
SCT	switch	on	small-curvature tunneling
ZCT	switch	on	zero-curvature tunneling
<i><u>Others:</u></i>			
EMIN	variable	0	cutoff E for computing percent tunneling
QRST	list	not used	quantized-reactant-state tunneling
HARMONIC	switch	on	harmonic approx.
HARMWKB	switch	off	harmonic-WKB approx.
SYMWKB	switch	off	WKB and symmetric V_a^G
MODE	variable	required	mode number of the rph mode
SRW	variable	required	s value of the reactant
STATES	variable	all	number of states included in calc.
QUAD	list	no changes	change quadrature parameters
NSEGBOLTZ	variable	1	segments for Boltzmann integral
NSEGTHETA	variable	1	segments for theta integral
NQE	variable	40	quad. pts. For Boltzmann integral
NQTH	variable	40	quad. pts. For theta integral
SCTOPT	list	<i>spline</i>	small-curvature tunneling options
LAGRANGE	variable	4	lagrange interpolation for μ_{eff}
SPLINE	switch	on	spline fit for interpolating μ_{eff}
WIGNER	switch	on	Wigner tunneling

Large-curvature tunneling:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<i>Common:</i>			
ALLEXCIT	switch	on	tunneling into all excited states
EXCITED	variable	0.0	Tunneling into nth excited state
LCT	switch	off	large-curvature tunneling
LCTDETAIL	list	Not used	print extra details in LCT calculations
<i>Others</i>			
LCTOPT	list	Not used	large-curvature tunneling options
ILCT	switch	off	interpolated LCG calculation
INTERPOLATE	variable	4	order of interpolation (see glossary)
LCGMETHOD	variable	4	method for LCT calculations
LCTRST	switch	off	restart a previous LCT calculation
LCTSTR	switch	on	store information from a LCT calculation for a possible restart
NGAMP	variable	120	quad. pts. for tunneling amplitude
NGTHETA	variable	120	quad. pts. for theta integral
NILCGPT	variable	9	number of interpolated points at each tunneling energy
PRFREQ	switch	off	print information (see glossary)
PRPROB	switch	off	print uniform probabilities
RTPJAC	switch	off	print information in pseudo-Jacobian coordinates
VADAVG	switch	off	average the overlapping of two adiabatic potentials
VEFSRCH	variable	0	search for possible nonadiabatic LCG4 regions

Glossary of TUNNEL keywords

ALLEXCIT (NOALLEXCIT)

ALLEXCIT is a switch keyword that specifies that the large-curvature tunneling calculations (see LCT) are to be performed with tunneling into all accessible excited states in the LCT exit arrangement. Default is on. The opposite switch, NOALLEXCIT, specifies that tunneling calculations be performed with tunneling into only the ground state.

Note that the maximum number of excited states is set by the MAXPS parameter in the param.inc INCLUDE file (see Subsection 5.A.2).

EMIN

EMIN is a variable keyword whose argument specifies the energy (in kcal/mol) that is the cutoff point in computing the percent tunneling that comes from below E_{\min} . The default is *0.0*.

Example:

```
EMIN 48.35
```

EXCITED

EXCITED is a variable keyword whose argument is the state n . The LCT tunneling calculation is performed with tunneling into up to the state with quantum number n , where $n = 0, 1, \dots$. Default is 0. The example below is for the case where tunneling will be performed for the first excited state in the LCT exit arrangement (if accessible) as well as for the ground state. Note that this keyword shouldn't be used together with the ALLEXCIT or NOEXCIT keyword.

Example:

```
EXCITED 1
```

LCT3 or LCT4

One of these two keywords is compulsory if large curvature calculations are required. The default is off. If LCT3 (the old LCT) or LCT4 (see Section 10.4. for details) are on, then the COMP and μ OMT results (described in Section 6.I) are always calculated.

LCTDETAIL

LCTDETAIL is a list keyword that prints extra details of the LCT calculations to Fortran unit fu41 to fu47 files. The valid subkeyword options are defined below. See Section 12.

<u>Option</u>	<u>Description</u>
<i>state n</i>	<i>n</i> is the quantum number of the vibrational state (0 is the ground state) for which information is to be printed.
<i>interval m</i>	<i>m</i> is the number of intervals of the energy grid the user wants to print out for this particular quantum state; LB and UB are the indices of the lower and upper bounds of the energy grid interval, respectively.
<i>lb₁ ub₁</i>	
<i>lb₂ ub₂</i>	
⋮	
<i>lb_m ub_m</i>	

Example:

```
LCTDETAIL
  STATE      0
  INTERVAL    2
    20      22
    30      32

  STATE      1
  INTERVAL    1
    30      31
END
```

LCTGRID

This is a list keyword that specifies the grid to carry out the **ILCT1D** or **ILCT2D** calculations. The valid options are defined below

<u>Option</u>	<u>Description</u>
<i>state n</i>	<i>n</i> is the quantum number of the vibrational state (0 is the ground state) for which information is to be printed.
<i>Npoints</i>	are the single point energy calculations carried out at each tunneling energy and fitted to a spline. <i>Npoints</i> must be specified with ILCT1D . The default is <i>Npoints</i> equal to 9.
<i>NumberE NumberX</i>	The 2D grid to perform the ILCT2D calculation can be specified here. <i>NumberE</i> are the number of tunneling energies and <i>NumberX</i> is the number of points within each tunneling energy.

LCTOPT

LCTOPT is a list keyword that is used to set options for the large-curvature tunneling calculations. If this keyword is used, then it is assumed that the large-curvature tunneling calculations are desired and LCT is automatically switched on. The valid options are defined below.

<u>Option</u>	<u>Description</u>	<u>Default</u>
<i>ILCT1D</i>	This switch keyword is used to calculate LCT3 or LCT4 transmission factors. At each tunneling energy performs a 1D spline under tension fit of the nonadiabatic region. The number of energies calculated along a given tunneling path is specified in LCTGRID.	off
<i>ILCT2D</i>	This switch keyword can be used either to calculate LCT3 or LCT4 transmission factors, although we recommend its use to evaluate only the latter. This is a 2D spline under tension fit based on a grid of tunneling energies and points along the tunneling path for each tunneling energy. The grid is specified in LCTGRID.	off
<i>interpolate</i>	order of interpolation used for the LCG3 theta integrals, the excited state potential, the vibrational period, and the sine of the angle between the MEP and the tunneling path.	4
<i>lcgmethod</i>	Select the method for LCT calculations. Two values are possible: 3 and 4, for LCG3 and LCG4 methods respectively.	4
<i>lctrst</i>	restart a previous large curvature calculation from unit fu48. This is useful to calculate LCT4 transmission factors from a previous LCT3 run.	off
<i>lctstr</i>	store information in unit fu49 for a possible restart.	on
<i>ngamp</i>	number of points for Gauss-Legendre quadrature and trapezoidal rule in computing the total tunneling amplitude along the incoming trajectory in the LCG3 and LCG4 methods.	120
<i>ngtheta</i>	number of points for Gauss-Legendre quadrature in computing the theta integral over the straight-line tunneling path in the LCG3 and LCG4 methods	120
<i>prfreq</i>	print information about the frequency of the mode $p(s)$, the classical energy $V_{\text{MEP}}(s)$, and the vibrationally adiabatic potential energy as a function of s and $n_p(V_a(n_p, s))$, and all the uniformized probabilities at each n_p .	off
<i>prprob</i>	print all uniformized probabilities at each n_p	off
<i>rtpjac</i>	print the MEP and representative tunneling path turning points in mass-scaled pseudo-Jacobian coordinates to fu24	off
<i>vadavg</i>	average the overlapping of two adiabatic potentials instead of choosing the minimum of them.	off
<i>vefsrch numb</i>	search for possible nonadiabatic LCG4 regions in totally adiabatic LCG3 regions. The search starts from that tunneling energy until <i>numb</i>	0

Example:

```
LCTOPT
  lctrst
  ngamp      120
  ngtheta    120
END
```

QRST

QRST is a list keyword that is used to specify the options for the quantized-reactant-state tunneling calculations. This option should only be used with unimolecular reactions. It is described in Refs. 8 and 37 of Sect. 21. Be sure that MXWKB is large enough if you use this (See Section 5.A.2. MXWKB must be greater than or equal to the total number of energy levels.) STATES subkeyword determine the number of quantized reactant states to be included in the tunneling calculations. The options are *all* (all possible states are included in the calculation), *0* (only the vibrational ground state will be considered), *1* (only the ground state and the first vibrational states will be considered), etc. STQVIB subkeyword determine the number of quantized reactant states to be included in calculation of the reactant vibrational partition function. The options are *all* (all possible states are included in the calculation), and *same* (the number of states used for calculating the reactant vibrational partition function is the same as the number of states used for tunneling contributions). The valid options are defined below.

<u>Option</u>	<u>Description</u>	<u>Default</u>
<i>harmonic</i>	use the harmonic approximation to determine the energy levels of the reaction-path mode	on
<i>harmwkb</i>	between the reactant and saddle point use the WKB approximation to find the energy levels but use the harmonic potential as the effective potential to the left of the reactant	off
<i>symwkb</i>	use the WKB approximation to find the energy levels and assume that the effective potential is symmetric with respect to the reactant	off
<i>mode</i>	the mode number of the reaction-path mode	required
<i>srw</i>	the value of the reaction coordinate in bohr at the position of the reactant	required
<i>states</i>	the number of quantized reactant states to be included in the tunneling calculation	<i>all</i>
<i>stqvib</i>	the number of quantized reactant states to be included in the vibrational partition function calculation	<i>all</i>

Example:

```
QRST
  SYMWKB
  STATES      5
  MODE        3
  SRW         -2.42
  STQVIB     all
END
```

QUAD

QUAD is a list keyword that is used to specify the Gauss-Legendre quadrature points in each segment used in computing the Boltzmann average (NQE) and theta integrals (NQTH) in the tunneling calculations. The valid options are defined below.

<u>Option</u>	<u>Description</u>	<u>Default</u>
<i>nqe</i>	number of Gauss-Legendre quadrature points in each segment to be used for computing the Boltzmann average in the tunneling calculations; allowed values are 2-7, 9, 10, 12, 15, 20, 30, and 40	40
<i>nqth</i>	number of Gauss-Legendre quadrature points in each segment to be used for computing the theta integrals in the tunneling calculations; allowed values are 2-7, 9, 10, 12, 15, 20, 30, and 40	40
<i>nsegboltz</i>	number of equal segments for integrals that average the energy-dependent tunneling probabilities over a Boltzmann distribution of energies	1
<i>nsegtheta</i>	number of equal segments for the theta integrals	1

Example:

```
QUAD
  nqe          30
  nqth         30
  nsegboltz    2
  nsegtheta    2
END
```

SCT (NOSCT)

SCT is a switch keyword that specifies that small-curvature tunneling calculations be performed. Default is SCT.

SCTOPT

SCTOPT is a list keyword that is used to specify the method for interpolating the effective mass for the small-curvature tunneling calculations. See Section 10.E. The default is SPLINE. The valid options are defined below.

<u>Option</u>	<u>Description</u>	<u>Default</u>
<i>lagrange n</i>	Lagrangian interpolation of order n is used to interpolate $\mu_{\text{eff}}(s)$	4
<i>spline</i>	spline fit is used to interpolate $\mu_{\text{eff}}(s)$	on

Example:

```
SCTOPT
  lagrange    5
END
```

WIGNER

WIGNER is a switch keyword that specifies that the Wigner transmission coefficient be computed. Default is on because it is inexpensive, but it is not a reliable method. In the current version of POLYRATE, this option cannot be turned off.

ZCT (NOZCT)

ZCT is a switch keyword that specifies that zero-curvature tunneling calculations be performed. Default is ZCT.

11.A.9. RATE section

The RATE section contains the keywords that control the rate constant calculations. This section is not needed if rate constants are not desired. If this section is not included then all defaults are ignored, and the default CVT and TST calculations are not performed. For simplicity the keywords have been broken into three categories: basic, methods, and print. This is only for clarity in the manual. The keywords in any of these three categories can be used in any order. Also for simplicity the keywords in each category have been labeled as either *common* or *other*. The keywords labeled *common* are those that are found in most input files for a standard POLYRATE run. This labeling is only intended as an aid to the novice user. Again an alphabetical glossary of all the keywords follows the three summary tables.

Basic:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<i>Common:</i>			
BOTHK	switch	on	forward and reverse rate constants
EACT	list	not used	activation energy calculation
REVKEXP	variable	0	scale factor for reverse rates
SIGMAF	variable	1	sym. factor for forward reaction
SIGMAR	variable	1	sym. factor for reverse reaction
TEMP	list	see below	temperatures for rate calculations
<i>Others:</i>			
ANALYSIS	list	not used	detailed analysis of TST/CVT ratio
EDGEOK	switch	on	continue calc. even if ΔG_{\max} is near end of grid
FORWARDK	switch	off	forward rate constant only
GSPEC	list	not used	special grid for ΔG calculation
SMAX	variable	1.0	max. <i>s</i> for special grid
SMIN	variable	-1.0	min. <i>s</i> for special grid
GTEMP	list	not used	special grid for ΔG calc. for given <i>T</i>
GTLOG	switch	off	free energy of activation by logarithms
STATE	variable	<i>therm</i>	thermal or state-selected rate constant
STATEOPT	list	<i>thermal</i>	options for STATE keyword
SPTOPT	list		options for simple pert. theory
CORIOLIS	switch	off	include Coriolis term
PRINT	switch	off	extra data printed out
VPFEXP	variable	0	scale factor for vib. partition function

Methods:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<u>Common:</u>			
CVT	switch	on	canonical VTST
<u>Others:</u>			
TST	switch	on	conventional transition state theory
ICVT	switch	off	improved canonical VTST
MUVT	switch	off	microcanonical VTST
MUVTOPT	list		options for μ VT
FIT	variable	<i>gas53</i>	primary and secondary fits
NITER	variable	<i>30</i>	Gauss-Laguerre quadrature
PREENERGY	variable	<i>0</i>	# of energies on the grid to print
SLOWER	variable	<i>0.0</i>	lower limit of MEP for searching
SUPPER	variable	<i>0.0</i>	upper limit of MEP for searching
CUS	variable	off	canonical unified statistical model
US	switch	off	unified statistical calculation

Print:

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
<i><u>Others:</u></i>			
PRDELG	switch	off	print free energy curve
PRGIGT	switch	off	print improved gener. ΔG
PRPART	variable	not used	print react., prod., & s.p. part. func.
PRVIB	list	not used	region to print extra vib mode info
SX	variable	0.0	s at beginning of desired region
SY	variable	0.0	s at the end of desired region

Glossary of RATE keywords

ANALYSIS

ANALYSIS is a list keyword used to specify the temperatures (in Kelvin) at which a detailed analysis of the conventional TST/CVT ratio is to be performed. The default is not to do a detailed analysis at any temperature. (Note: the rate constant must be computed at all temperatures on this list; see the TEMP keyword.)

Example:

```
ANALYSIS
  298.
  400.
  800.
END
```

BOTHK (NOBOTHK)

BOTHK is a switch keyword that specifies that both the forward and reverse rate constants and the equilibrium constant are to be calculated. If NOBOTHK is specified no rate constants are computed; use FORWARDK if only the forward rate constant is desired. The default is BOTHK. (See REVKEXP for scaling the output for the reverse rate constant.)

CUS (NOCUS)

CUS is a variable keyword that specifies that rate constants should be calculated with the canonical unified statistical model. See Ref. 51 in Sect. 20. The input argument of this keyword indicates the maximum number of local maxima in the free energy of activation vs. reaction coordinate profiles that should be included at each temperature. If the number of local maxima that are found is less than this argument, then all maxima are used. The CUS calculations will also include the lowest local minimum between any pair of included local maxim. When the argument of CUS equals 1, the CUS result is exactly the same as CVT. The default is NOCUS.

Example:

```
CUS    2
```

CVT (NOCVT)

CVT is a switch keyword that specifies that generalized transition state theory and canonical variational theory rate constants be calculated. The default is CVT. If NOCVT is specified and NOTST is not specified, rate constants will be calculated but only at the conventional-TST level. In this case, the calculations are carried out in subprogram TSRATE (as opposed to RATE or GIVTST). TSRATE does not include anharmonicity.

EACT

EACT is a list keyword used to specify the pairs of temperatures (in Kelvin) for which to compute Arrhenius activation energies (maximum of 10 pairs) from the calculated rate constants. Each line should first give the lower temperature and then the upper temperature for the pair. The default is not to compute any activation energies. (Note: the rate constant must be computed at all temperatures in this list; see the TEMP keyword.)

Example:

```
EACT
    298.  400.
    400.  600.
END
```

EDGEOK (NOEDGEOK)

EDGEOK is a switch keyword that specifies that the calculation be continued even if V_a or ΔG^{ICVT} is near an end point, i.e., even if the grid does not appear wide enough to find variational transition states. The default is EDGEOK.

FORWARDK (NOFORWARDK)

FORWARDK is a switch keyword that specifies that only the forward rate constant be calculated. If FORWARDK is specified, equilibrium constants will not be computed. The default is NOFORWARDK.

GSPEC

GSPEC is a list keyword that is used to specify that the generalized free energy of activation profile only be computed for the range $SMIN \leq s \leq SMAX$ for all temperatures. The default is to compute the free energy curve over the entire range of s . The defaults for the options are $SMAX = 1$ and $SMIN = -1$.

Example:

```
GSPEC
    SMAX  1.50
    SMIN -0.05
END
```

GTEMP

GTEMP is a list keyword that is used to specify that the generalized free energy of activation profile be computed over a different range of s for each temperature. Each entry in the list should contain the temperature in Kelvin and then $SMAX$ and $SMIN$ (in bohr) for that temperature. The default is to compute the free energy curve over the entire range of s . The defaults for the options are $SMAX = 1$ and $SMIN = -1$.

Example:

```
GTEMP
  298.  1.50  -0.10
  400.  1.00  -0.25
END
```

GTLOG (NOGTLOG)

GTLOG is a switch keyword that specifies that the generalized free energy of activation should be calculated by using the logarithms of the vibrational partition functions. This type of calculation is preferred when the vibrational partition function has an extremely small value (as may be the case for very low temperatures and/or for very big systems because we put the zero of energy for the partition coefficient at the lowest value of the potential rather than at the zero point energy). The default is NOGTLOG.

ICVT (NOICVT)

ICVT is a switch keyword that specifies that ICVT rate constants should be calculated. The default is NOICVT.

MUVT (NOMUVT)

MUVT is a switch keyword that specifies that μ VT rate constants should be calculated. The default is NOMUVT if US and MUVTOPT are off, and the default is MUVT if US is on or MUVTOPT is given.

MUVTOPT (NOMUVTOPT)

MUVTOPT is a list keyword that is used to set options for the μ VT calculation. If MUVTOPT is given then it is assumed that the μ VT calculation is desired and the default for MUVT becomes on. The valid options are defined below.

<u>Option</u>	<u>Description</u>	<u>Default</u>
<i>niter</i>	number of Gauss-Laguerre quad. pts. for computing μ VT rate constants	30
<i>preenergy</i>	number of energies on the grid to be printed out; PREENERGY must be $\leq \min[6, niter]$	0
<i>slower</i>	lower limit (in bohr), respectively, of the region of the MEP over which the GT minimum of $N_{\mu VT}$ will be searched for. (This is only specified if one wishes to limit the search. Leave set to the default of zero to search over the entire s range.)	0.0
<i>supper</i>	upper limit (in bohr), of the region of the MEP over which the GT minimum of N will be searched for. (This is only specified if one wishes to limit the search. Leave set to the default of zero to search the entire s range.)	0.0
<i>fit</i>	variable that specifies the degree of the primary and secondary fits. The allowed values are given below. For a gas-phase reaction, <i>gas53</i> is the recommended value, and for a reaction at a gas-solid interface, <i>gsolid13</i> is recommended.	<i>gas53</i>

<u>value</u>	<u>primary</u>	<u>secondary</u>	<u>phase</u>
<i>gas53</i>	5	3	gas
<i>gas15</i>	1	5	gas
<i>gas35</i>	3	5	gas
<i>gsolid13</i>	1	3	gas-solid interface
<i>gsolid15</i>	1	5	gas-solid interface
<i>gsolid31</i>	3	1	gas-solid interface
<i>gsolid53</i>	5	3	gas-solid interface

Example:

```
MUVTOPT
  niter 40
END
```

PRDELG (NOPRDELG)

PRDELG is a switch keyword that specifies that $\Delta G(T,s)$ be printed. Default is NOPRDELG.

PRGIGT (NOPRGIGT)

PRGIGT is a switch keyword that specifies that the improved generalized free energy of activation curve be printed. Default is NOPRGIGT.

PRPART (NOPRPART)

PRPART is a variable keyword. Its argument specifies that the reactant and product (*rp*) partition functions, the saddle point (*t*) partition functions, or the reactant, product, and saddle point (*rpt*) partition functions should be printed. When the reactant partition function is printed for a gas-phase unimolecular reaction the internal free energy of reactant is also printed. The default is not to print any partition functions or free energies at the stationary points.

Example:

```
PRPART    rp
```

PRVIB

PRVIB is a list keyword that is used to specify the initial and final *s* values, with keywords *sx* and *sy* respectively, in bohr that delineate a region for which extra vibrational mode and Q_{vib} information be printed. The default is no extra printing.

Example:

```
PRVIB
  SX      0.0
  SY      0.6
END
```

REVKEXP

REVKEXP is a variable keyword whose argument specifies the value *n* in the scale factor 10^{-n} for the reverse rate constant. The final rate will be calculated in units of $10^{-n} \text{ cm}^3 \text{ molecule}^{-1} \text{ s}^{-1}$ if there are two products or in units of 10^{-n} s^{-1} if there is one product. Default is $n = 0$, i.e., a scale factor of 1.

Example:

```
REVKEXP   10
```

SIGMAF

SIGMAF is a variable keyword whose argument specifies the number of equivalent ways for forming the GTS for the forward reaction. The default value is 1.

Example:

```
SIMGAF    1
```

SIGMAR

SIGMAR is a variable keyword whose argument specifies the number of equivalent ways for forming the GTS for the reverse reaction. The default value is 1.

Example:

```
SIMGAR      1
```

STATE

STATE is a variable keyword whose argument specifies if this is to be a thermal or state-selected rate constant calculation. The default is *therm*. The valid arguments are:

<u>Option</u>	<u>Description</u>
<i>therm</i>	thermal run
<i>adiab</i>	adiabatic state-selected; STATEOPT is required
<i>diab</i> <i>x</i>	diabatic state-selected; <i>x</i> is the value of the reaction coordinate at which the adiabatic/diabatic switch is made; STATEOPT is required

Example:

```
STATE  diab
```

STATEOPT

STATEOPT is a list keyword that is used to specify how each mode in each reactant, product, and transition state is to be treated. By default each mode is treated thermally. The user needs to specify each mode explicitly that is to be restricted to the ground state or first excited state as shown below. Each input line in the list requires:

reatyp *nmode* *modtyp*

The allowed values of the variable *reatyp* are:

<u>Value</u>	<u>Description</u>
<i>react1</i>	reactant #1
<i>react2</i>	reactant #2
<i>prod1</i>	product #1
<i>prod2</i>	product #2
<i>ts</i>	transition state and along the reaction path
<i>switts</i>	reaction path after the adiabatic/diabatic switch (see STATE)

nmode is an integer specifying the vibrational mode number

The allowed values of the variable *modtyp* are:

<u>Value</u>	<u>Description</u>
<i>thermal</i>	mode is thermal (default)
<i>ground</i>	mode is restricted to the ground state
<i>first</i>	mode is restricted to the first excited state

Example:

```
STATEOPT
  react1  1  ground
  react2  1  first
  prod1   1  thermal
  prod1   2  ground
  prod1   3  first
  ts      4  ground
  ts      5  first
  switts  4  first
  switts  5  ground
END
```

TEMP

TEMP is a list keyword used to specify the temperatures (in Kelvin) at which rate constants be computed. Unless NOBOTHK or FORWARDK is selected the equilibrium constants will also be calculated at these temperatures. The default is to compute the rate constants at 7 temperatures: 200, 300, 400, 600, 1000, 1500, and 2400.

Example:

```
TEMP
  298.
  400.
  600.
```


800.
END

TST (NOTST)

TST is a switch keyword that specifies that the conventional transition state theory rate constants be calculated. The default of TST is on. If any other rate constants are calculated, i.e., CVT, ICVT, US, or μ VT, then the conventional rate constants will automatically be calculated (NOTST will have no effect).

US (NOUS)

US is a switch keyword that specifies that unified statistical rate constants be calculated. If US is on, then MUVT is automatically turned on. The default of US is off, which is the same as specifying NOUS.

VPFEXP

VPFEXP is a variable keyword that specifies that the total vibrational partition functions be scaled by 10^n where n is the argument. The default is no scaling.

Example:
VPFEXP 2

11.B. General description of file fu29 input

File fu29 is only needed if IVTST0 or IVTST1 is specified in the GENERAL section of the unit fu5 input. This unit contains input data for IVTST rate calculations. The first section of this input file must be *IVTST. All input sections must start with the section name preceded by an "*", and must end with an END line (a line containing "end"). Keywords and input data for each section are discussed further in the rest of this subsection. Sample inputs can be found in several test runs described in Section 14.

<u>Section Name</u>	<u>Description</u>
*E1GEOM	Cartesian coordinates of the extra point on the MEP
*E1GRAD	Cartesian gradient vector of the extra point on the MEP
*E1HESS	Cartesian Hessian of the extra point on the MEP
*IVTST	general data for IVTST calculations (must come first)
*P1FREQ	frequencies of the 1st product
*P2FREQ	frequencies of the 2nd product
*P1HESS	Cartesian Hessian of the 1st product
*P2HESS	Cartesian Hessian of the 2nd product
*R1FREQ	frequencies of the 1st reactant
*R2FREQ	frequencies of the 2nd reactant
*R1HESS	Cartesian Hessian of the 1st reactant
*R2HESS	Cartesian Hessian of the 2nd reactant
*TSH Hess	Cartesian Hessian of the saddle point
*TSFREQ	frequencies of the saddle point

When IVTST0 is specified, the user can choose between giving the Hessian or the frequencies for the stationary points (reactants, saddle point and products) according to the kind of information available for the system. When running an IVTST1 calculation

the use of Hessians for the stationary and extra points is required, since this information is needed in order to calculate the SCT transmission factors.

11.B.1. E1GEOM Section

The E1GEOM section of the file fu29 input is used to give the Cartesian coordinates of the extra point on the MEP. This section is only needed for first-order IVTST calculations.

Example:

```
*E1GEOM
 0.000000  0.000000  0.000000
 0.456364  1.749560  0.000000
 1.808100  0.000000  0.000000
END
```

11.B.2. E1GRAD Section

The E1GRAD section of the file fu29 input is used to give the Cartesian gradient vector of the extra point on the MEP. This section is only needed for first-order IVTST calculations.

Example:

```
*E1GRAD
 0.000000  0.000000  0.000000
 0.056364  0.009560  0.000000
 0.008100  0.000000  0.000000
END
```

11.B.3. E1HESS, P1HESS, P2HESS, R1HESS, R2HESS, and TSHESS sections

The E1HESS, P1HESS, P2HESS, R1HESS, R2HESS, and TSHESS sections of file fu29 input are used to give the Cartesian Hessian matrix of the extra point on the MEP, products, reactants, and the transition state in packed form, respectively. The E1HESS section is only needed for first-order IVTST calculations.

Example:

```
*E1HESS
 0.000000
 0.056364  0.009560
 0.008100  0.000000  0.000000
 0.008100  0.000000  0.000000  0.080100
 0.008100  0.000000  0.000000  0.080100  0.080100
 0.008100  0.000000  0.000000  0.080100  0.080100
 0.080100
END
```

11.B.4. IVTST Section

The IVTST section of the file fu29 input is used to give general data for IVTST calculations. The table below lists all valid subkeywords for the IVTST section. A detailed explanation of each keyword is given alphabetically in the glossary below.

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
BARRI	variable	required	classical barrier height in kcal/mol
DELTAE	variable	required	classical energy of the products in kcal/mol
DELEX1	variable	not used	the classical energy of the extra point in kcal/mol (only needed for the IVTST1)
PRINT	switch	0	output control
SEX1	variable	not used	the reaction coordinate at the extra point in atomic units (only needed IVTST1)
UNITS	switch	0	units for geometry, gradient, and Hessian matrix

Glossary of IVTST keywords

BARRI

BARRI is a keyword that specifies the classical barrier height in kcal/mol. In case a conversion of units is required for obtaining this value in kcal/mol, the conversion factor used should be the same as that internally employed by POLYRATE as specified in the percon.inc file.

Example:

```
BARRI      5.0
```

DELTAE

DELTAE is a keyword that specifies the classical energy of the products relative to the classical energy of the reactants in kcal/mol. This is a required keyword. In case a conversion of units is required for obtaining this value in kcal/mol, the conversion factor used should be the same as that internally employed by POLYRATE as specified in the percon.inc file.

Example:

```
DELTAE    -5.0
```

DELEX1

DELEX1 is a keyword that specifies the classical energy of the extra point relative to the classical energy of the reactants in kcal/mol. This keyword is only needed for first-order IVTST calculations.

Example:

```
DELEX1    -4.4
```

PRINT

PRINT is a keyword for output control: 0—normal (default), 1—debug.

Example:

```
PRINT      1
```

SEX1

SEX1 is a keyword that specifies the reaction coordinate at the extra point in bohr. This keyword is only needed for first-order IVTST calculations.

Example:

```
SEX1       0.1
```

UNITS

UNITS is a keyword that specifies the units for the geometry, gradients and Hessian matrix: UNITS = 0: The geometry will be in bohrs, and both the gradients, and the Hessian matrix will be in mass-scaled atomic units (default). UNITS = 1: The geometry will be in mass-scaled Å, and both the gradient vectors and the Hessian matrix will be in non-mass-scaled atomic units.

Example:

```
UNITS      0
```

11.B.5. P1FREQ, P2FREQ, R1FREQ, R2FREQ, and TSFREQ sections

The P1FREQ, P2FREQ, R1FREQ, R2FREQ, and TSFREQ sections of the file fu29 input are used to give the frequencies (in cm^{-1}) of the product and reactant molecules and the saddle point, respectively. The imaginary frequency of the saddle point must be the last input value in the TSFREQ section.

Example:

```
*R1FREQ
  2000.0  1000.0
END
```

11.C. General description of file fu30 input

File fu30 contains the information typically obtained from electronic structure calculations which is required for performing a direct dynamics calculation (see Section 9 for a detailed description of this capability). This option is selected by setting the keyword POTENTIAL in the ENERGETICS section equal to *unit30*.

File fu30 is organized in the following way: 1) a one-line title; 2) 2 lines with numerical options for routing and output; and 3) several blocks of data containing the specific information provided for each point on the reaction path according to the options selected (s, V, frequency matching data, Cartesian coordinates, gradient, frequencies or Hessian matrix, anharmonicity data, curvature component data, etc.).

The user should be aware that gradients provided in the fu30 file must correspond to structures that have not been arbitrarily reoriented along the reaction path in order to obtain meaningful results in the calculation of the curvature components B_{mF} . See Section 9 for a fuller discussion of this issue.

A record-by-record description (where a record is one or more lines) of the structure of the FORTRAN unit fu30 follows. If the keyword POTENTIAL = *unit30*, POLYRATE can be used with the dummy potential energy surface file provided, dumpot.f, since the potential routines SURF and SETUP will not be called.

<u>Record</u>		<u>Format</u>
	(Records 1 and 2 are read in subprogram RPHSET)	
1	One-line title	(80A1)
2	Routing and output options (LOPT(I), I = 1, 40)	(*)

LOPT(1) LEVEL OF PRINT

- ≤ 0, print out only titles and options
- = 1, also print out summary
- = 2, also echo input as it is read
- = 3, also print out intermediate information about fitting exponential functions and print out interpolated results each time RPHINT is called
- ≥ 4, also print out debugging information in exponential fitting

LOPT(2) MAXIMUM NUMBER OF POINTS USED IN LAGRANGE INTERPOLATION IN THE PROCEDURE OF SUBSECTION 9.C.

= -1, use IVTST-M for interpolating. See Sect. 9.C.

Note: When using file fu30, the user cannot select any options for the IVTST-M calculation; it will be performed according to the default options, and the number of gradient and Hessian points have to be the same and equal to NPTS (see below). Therefore, only IVTST-M-NPTS/NPTS calculations are allowed. The curvature components will be interpolated using a value of 3 for the exponent in the mapping function. For unimolecular reactions or reactions with wells, a cubic polynomial fitting to V is done previous to its spline fitting, taking a step of $0.01 a_0$ along the lowest frequency mode for performing this fitting. The only way to change any of these options or to use a different number of gradient and Hessian points is by using the fu31 input file, which is specifically designed for IVTST-M calculations.

LOPT(3) OPTIONS FOR COORDINATES (X) AND GRADIENTS (DX)

≤ -2,	X in mass-scaled Cartesians, DX in unscaled Cartesians
= -1,	X in unscaled Cartesians, DX in mass-scaled Cartesians
= 0,	X and DX in unscaled Cartesians
= 1,	X and DX in mass-scaled Cartesians
= 99,	X in unscaled Cartesians, -DX in mass-scaled Cartesians
= 100,	X in unscaled Cartesians, -DX in unscaled Cartesians

Note: DX denotes the gradient, and -DX denotes the negative of the gradient which is the force. For dynamics calculations in Cartesian coordinates, the normalized gradients (or forces) can be used without any effects on the results. However, unnormalized gradients have to be used for the curvilinear option. Notice the ‘Cartesian Forces’ from GAUSSIAN formatted check point output file are actually the Cartesian gradient components.

LOPT(4) OPTIONS FOR FREQUENCY OR HESSIAN MATRIX INPUT

There are three issues:

- (1) Are the frequencies read in from unit fu30, or are they obtained by diagonalizing the Hessian matrix \mathbf{F} or the projected Hessian \mathbf{F}^P ?
- (2) If the frequencies are calculated from the Hessian, how is \mathbf{F} read in, in packed (triangular) form, in packed form with full lines, or in unpacked (full matrix) form?
- (3) What coordinate system was used in computing \mathbf{F} ?

LOPT(4)	(1)	(2)	(3)
≤ -2,	diagonalization	packed	mass-scaled coordinates
= -1,	diagonalization	packed	unscaled coordinates
= 0,	frequencies are read		

= 1, diagonalization full matrix unscaled coordinates
 = 2, diagonalization full matrix mass-scaled coordinates

- Notes: (1) If LOPT(4) \neq 0, the choice of coordinates (mass-scaled or not) must be consistent with LOPT(3).
 (2) If the Hessian is read, the code assumes it is the non-projected Hessian.
 (3) Packed full lines means 5 elements per line; whereas packed triangular means F(1,1) on first line, F(2,1) and F(2,2) on second line, etc.

LOPT(5) ANHARMONICITY INPUT

\leq 0, no anharmonicity data read
 = 1, read the third derivatives k_3 with respect to the normal coordinates at each grid point
 \geq 2, also read the fourth derivatives k_4 with respect to the normal coordinates at each grid point

LOPT(6) OPTIONS FOR CURVATURE COMPONENT B_{mF} INPUT

\leq 0, B_{mF} are set to zero everywhere
 = 1, B_{mF} are read directly
 = 2, B_{mF} are computed. Derivative of the gradient with respect to s is obtained from a quadratic fit of the gradient at three points on the basic input grid (defined in Section 9.C), with the middle point being the current point on the Hessian grid.
 = 3, B_{mF} are computed. Derivative of the gradient is computed by one-sided finite difference, using the gradient on the basic input grid and a gradient input at one extra point near the basic input grid point.
 \geq 4, B_{mF} are computed. Derivative of the gradient is computed from a quadratic fit obtained using the gradient on the basic input grid and gradients input at an extra point on each side of the basic input grid point.

Note: For LOPT(6) $>$ 1, B_{mF} are computed and require the coefficient matrix from the diagonalization of the Hessian matrix. Therefore, LOPT(4) must be nonzero if LOPT(6) $>$ 1.

LOPT(7)-LOPT(40): Not used in the present version of POLYRATE.

Loop over records 3 through 14 for each reactant species and then for each product species. Records 3-25 are read in subprogram RPHRD2.

3 One-line title (80A1)

4 S, V (*)

S = value of s in bohrs at reactant (or product). This only has meaning for a unimolecular and/or association reaction, for which s is finite at the reactant and/or product; otherwise S is ignored.

V = potential (in atomic units) at reactants (or products).

For an atomic reactant or product, skip records 5 through 14, **add a blank line after record 4**, and then continue with the next species.

5 (IFREQ(I), I = 1, NFREQ) (*)

IFREQ = indices for reordering frequencies, anharmonicities, and curvature components in the interpolation table for the reactant or product species, in order to make the ordering consistent with that at the saddle point and/or along the MEP.

Note: Index 1 corresponds to the largest frequency at the saddle point, index 2 to the second largest frequency, and so on. The imaginary frequency is not included in this ordering.

Note that this matching is not necessary if you are just calculating conventional TST reaction rates.

NFREQ is the number of bound vibrational modes for the species.

For a Morse diatom reactant or product, read record 6; otherwise go to record 7.

6 SGN

SGN = sign used to multiply the third derivative of the diatomic potential. This can be used to make the sign of the third derivative of this mode in the asymptotic reactant or product region match up with the correlating mode in the interaction region. This ensures a smooth extrapolation into the asymptotic region.

For a Morse diatom reactant or product, skip records 7 through 14. If LOPT(4) = 0 (frequencies directly input), skip records 7 and 8.

7 One-line title (80A1)

8 N records of (F(I,J), J = 1, JF) N records of (*)

\mathbf{F} = Hessian matrix (in atomic units) for the reactant or product species. Second derivatives are with respect to Cartesian coordinates; LOPT(4) determines whether the coordinates are mass-scaled or not. $JF = N$ for LOPT(4) > 0 and $JF = I$ for LOPT(4) < 0, where $N = 3 \cdot \text{NRATOM}$, NRATOM = number of atoms in the species, and I indexes the N records in this set ($I = 1, \dots, N$).

If LOPT(4) \neq 0, skip records 9 and 10.

9 One-line title (80A1)

10 (FREQ(I), I = 1, NFREQ) (*)

FREQ = vibrational frequencies (**in atomic units**) for the reactant or product species.

If LOPT(5) < 1 (no anharmonicity), skip records 11 through 14.

11 One-line title (80A1)

12 (XK3(I), I = 1, NFREQ) (*)

XK3 = third derivatives of V with respect to motion along the normal coordinates for the reactant or product species.

If LOPT(5) < 2 (just third derivatives), skip records 13 and 14.

13 One-line title (80A1)

14 (XK4(I), I = 1, NFREQ) (*)

XK4 = fourth derivatives of V with respect to motion along the normal coordinates for the reactant or product species.

After each reactant or product species except for the last product, go back to record 3.
After the last product, go to the next record.

If there is no saddle point, skip records 15 through 27.

15 One-line title (80A1)

16 S, V (*)

S = a dummy quantity and is ignored.
 V = potential (in atomic units) at the saddle point.

17 (IFREQ(I), I = 1, NFREQ) (*)

These quantities are the same as in record 5, but now they refer to the bound vibrational modes of the saddle point. NFREQ is now the number of bound vibrational modes of the saddle point.

If LOPT(4) = 0, skip records 18 and 19.

18 One-line title (80A1)

19 $N3$ records of ($\mathbf{F}(I,J)$, $J = 1, JF$) $N3$ records of (*)

These quantities are the same as in record 8, but now they refer to the saddle point. $JF = N3$ for LOPT(4) > 0 and $JF = I$ for LOPT(4) <

0, where I indexes the $N3$ records in this set ($I = 1, \dots, N3$), and where $N3 = 3 * \text{NATOM}$, NATOM = number of atoms in the system.

If LOPT(4) \neq 0, skip records 20 and 21.

20 One-line title (80A1)

21 (FREQ(I), I = 1, NFREQ) (*)

These quantities are the same as in record 10, but now they refer to the bound vibrational modes of the saddle point.

If LOPT(5) < 1 (no anharmonicity), skip records 22 through 25.

22 One-line title (80A1)

23 (XK3(I), I = 1, NFREQ) (*)

These quantities are the same as in record 12, but now they refer to the bound vibrational modes of the saddle point.

If LOPT(5) < 2 (just third derivatives), skip records 24 and 25.

24 One-line title (80A1)

25 (XK4(I), I = 1, NFREQ) (*)

These quantities are the same as in record 14, but now they refer to the bound vibrational modes of the saddle point.

If LOPT(4) \neq 0, skip records 26 and 27.
Records 26-31 are read in subprogram RPHRD1.

26 One-line title (80A1)

27 WSTAR (*)

WSTAR = absolute value of the imaginary frequency at the saddle point.

If the reaction path is calculated, read records 28 to 63.

28 One-line title (80A1)

29 NPTS (*)

NPTS = number of s points along the MEP (not including the saddle point) at which RPH data is to be read (maximum of 200). Unless only TST reaction rates are calculated, file fu30 must contain a minimum of one point on each side of the saddle point. Unless NEXTPT is on, these points must be arranged in order of increasing S .

Loop over records 30 through 52 NPTS times, **from the lowest value of s to the highest value of s** . Records 30-50 are read in subprogram RPHRD2.

30 One-line title (80A1)

31 S, V (*)

S = value of s (in bohr) at the grid location along the MEP.

V = potential (in hartrees) at s on the MEP.

32 One-line title (80A1)

33 $(X(I), I = 1, N3)$ (*)

X = Cartesian coordinates of the GTS at s on the MEP.

$N3 = 3 * (\text{number of atoms in the total system})$. Ordering is the same as specified in the ATOM keyword of the FORTRAN unit fu5 data.

Note: LOPT(3) determines whether the coordinates are mass-scaled or not.

34 One-line title (80A1)

35 $(DX(I), I = 1, N3)$ (*)

DX = gradient vector (in atomic units) at s along the MEP. DX can be the vector of derivatives of the potential with respect to the Cartesian coordinates or it can be normalized. LOPT(3) determines whether the coordinates are mass-scaled or not. Ordering is the same as for $X(I)$ in record 33.

Note: Some electronic structure codes print the force rather than the gradient. Recall that the force is the negative of the gradient.

If LOPT(6) ≤ 2 , skip records 36 through 41.

36 One-line title (80A1)

37 $SB(1)$ (*)

$SB(1)$ = value of s (in bohr) at a location along the MEP close to the current grid point.

38 $(DX(I), I = 1, N3)$ (*)

These quantities are the same as in record 35, but now they refer to the location $SB(1)$ on the MEP.

If LOPT(6) ≤ 3 , skip records 30 through 41.

39 One-line title (80A1)

40 SB(2) (*)

SB(2) = value of s (in bohr) at another location along the MEP close to the current grid point.

41 (DX(I), I = 1, N3) (*)

These quantities are the same as in record 35, but now they refer to the location SB(2) on the MEP.

42 (IFREQ(I), I = 1, NFREQ) (*)

These quantities are the same as in record 17, but now they refer to the bound generalized normal modes of the current grid point along the MEP.

If LOPT(4) = 0, skip records 43 and 44.

43 One-line title (80A1)

44 N3 records of (F(I,J), J = 1, JF) (*)

These quantities are the same as in record 19, but now they refer to the current grid point along the MEP.

If LOPT(4) \neq 0, skip records 45 and 46.

45 One-line title (80A1)

46 (FREQ(I), I = 1, NFREQ) (*)

These quantities are the same as in record 21, but now they refer to the current grid point along the MEP.

If LOPT(5) < 1 (no anharmonicity), skip records 47 through 50.

47 One-line title (80A1)

48 (XK3(I), I = 1, NFREQ) (*)

These quantities are the same as in record 23, but now they refer to the current grid point along the MEP.

If LOPT(5) < 2 (just third derivatives), skip records 51 and 52.

49 One-line title (80A1)

50 (XK4(I), I = 1, NFREQ) (*)

These quantities are the same as in record 25, but now they refer to the current grid point along the MEP.

If LOPT(6) = 1 (\mathbf{B}_{mF} are read), read records 51 and 52.
Records 51-63 are read in RPHRD1.

51 One-line title (80A1)

52 (BFS(I), I = 1, NFREQ) (*)

BFS = reaction-path curvature components (in atomic units) at the
current grid point along the MEP.

Return to record 30 until records 30 through 52 have been read NPTS times.

11.D. General description of file fu31 input

This file is only needed if the POTENTIAL keyword of the ENERGETICS section of file fu5 is equal to *unit31*. The input is in free format keyword style and is composed of several sections. When *unit31* is used, the code always uses the IVTST-M algorithm.

11.D.1. Section descriptions

<u>Section Name</u>	<u>Description</u>
*GENERAL	general data such as options for input and IVTST-M
*REACT1	information for the first reactant
*REACT2	information for the second reactant (if any)
*PROD1	information for the first product
*PROD2	information for the second product (if any)
*WELLR	information for the well on the reactant side (if any)
*WELLP	information for the well on the product side (if any)
*SADDLE	information for the saddle point
*POINT	information for a point along the MEP

Each section must appear once and only once except POINT, which has to appear at least twice (since IVTST-M is not defined for less than one point on each side of the reaction path). The POINT section should be repeated once for each of the nonstationary points along the MEP. A saddle point is also required for carrying out an IVTST-M calculation. Therefore, the fu31 input file cannot be used for reactions without a saddle point.

The program is insensitive to the order in which the points are entered. This is one of the main differences between the fu31 input file and the fu30 and fu40 files. This allows us to use an fu31 file written by means of the *writefu31* option for the RESTART keyword in the GENERAL section of the fu5 input file without any change. Another difference is that Hessians are not required at each point; since the calculation of the reaction path always gives a value of the reaction coordinate, *s*, the energy, geometry, and

gradients, this information is required, but the Hessian information is not. This way, the user can take advantage of the information calculated for points between two Hessian points.

Since the IVTST-M calculation is always based on frequencies ordered in canonical order, no frequency indexes are required. Moreover, since anharmonicity is not supported for IVTST-M calculations, in the current version of POLYRATE, no anharmonic data is allowed. From all other points of view, the fu31 input file is very similar to a fu30 or fu40 input file.

11.D.2 GENERAL section

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
GEOM	list	no changes	change defaults for reading geometries
GRADS	list	no changes	change defaults for reading gradients
HESSIAN	list	no changes	change defaults for reading Hessians
INTMUEFF	switch	<i>off</i>	interpolation of the exponent in the effective tunneling mass
KIES	switch	<i>off</i>	recalculation of the <i>s</i> values

Glossary of GENERAL keywords

GEOM

GEOM is a list keyword for selecting the units of the geometries input in the fu31 file. The units are completely defined by a combination of two keywords; thus, if we want to input the mass-scaled geometries in angstroms we will have to select both *ms* and *ang*. If one of the options is not given, the default value is assumed.

<u>Option</u>	<u>Definition</u>
<i>ang</i>	Geometries are given in Å.
<i>au</i>	Geometries are given in atomic units (default).
<i>ms</i>	Geometries are given in mass-scaled Cartesian coordinates.
<i>uns</i>	Geometries are given in unscaled Cartesian coordinates (default).

Example:

```
GEOM
      ang
      ms
END
```

GRADS

GRADS is a list keyword for selecting the units of the gradients (or forces) input in the fu31 file. The units are completely defined by a combination of three keywords.

<u>Option</u>	<u>Definition</u>
<i>ang</i>	Gradients are given in Å.
<i>au</i>	Gradients are given in atomic units (default).
<i>ms</i>	Gradients are given in mass-scaled Cartesian coordinates.
<i>uns</i>	Gradients are given in unscaled Cartesian coordinates (default).
<i>forces</i>	Forces are given; the gradients are obtained from them.
<i>grads</i>	Gradients are given (default).

Example:

```
GRADS
      ang
      uns
      forces
END
```

HESSIAN

HESSIAN is a list keyword for selecting the units of the Hessian matrices input in the fu31 file. The units are completely defined by a combination of three keywords.

<u>Option</u>	<u>Definition</u>
<i>ang</i>	Gradients are given in Å.
<i>au</i>	Gradients are given in atomic units (default).
<i>ms</i>	Gradients are given in mass-scaled Cartesian coordinates.
<i>uns</i>	Gradients are given in unscaled Cartesian coordinates (default).
<i>full</i>	Full Hessian matrices are given.
<i>packed</i>	Packed Hessian matrices are given (default).

Example:

```
HESSIAN
  full
END
```

If *packed* is selected, the lower triangular elements of **F** should be listed in order such that $F_{ij} = P(\frac{1}{2}i(i-1) + j)$, where $P(k)$ is element k of the input list and F_{ij} is the element of **F** in row i and column j . The number of elements listed per line is arbitrary.

Examples of both input forms for a 6 x 6 matrix is shown below:

Full matrix - example 1

```
F11 F12 F13 F14 F15 F16
F21 F22 F23 F24 F25 F26
F31 F32 F33 F34 F35 F36
F41 F42 F43 F44 F45 F46
F51 F52 F53 F54 F55 F56
F61 F62 F63 F64 F65 F66
```

Full matrix - example 2

```
F11 F12 F13 F14 F15 F16 F21 F22 F23
F24 F25 F26 F31 F32 F33 F34 F35 F36
F41 F42 F43 F44 F45 F46 F51 F52 F53
F54 F55 F56 F61 F62 F63 F64 F65 F66
```

Packed matrix - example 1

```
F11
F21 F22
F31 F32 F33
F41 F42 F43 F44
F51 F52 F53 F54 F55
F61 F62 F63 F64 F65 F66
```

Packed matrix - example 2

```
F11 F21 F22 F31 F32 F33
F41 F42 F43 F44 F51 F52
F53 F54 F55 F61 F62 F63
```

F_{64} F_{65} F_{66}

The order of the coordinates is explained in Section 11.E.1.

Note that GAUSSIAN94 writes the packed Hessian matrix in a slightly different way in the standard output file, but the Hessian in the formatted checkpoint file (requested by their FormCheck keyword) is in precisely the packed form accepted by POLYRATE and so is convenient for copying and pasting into POLYRATE fu40 files.

Note that the exact \mathbf{F} matrix is symmetric, but due to finite precision in computations, the actual matrix one has to input is usually nonsymmetric. If one inputs a nonsymmetric full matrix, the code uses the unsymmetric matrix. If a user wants the code to use a symmetric matrix, we recommend taking an arithmetic average of F_{ij} and F_{ji} for the resulting symmetrized elements in packed form.

INTMUEFF (NOINTMUEFF)

The interpolation of the effective mass for CD-SCSAG tunneling can be done in two ways in an IVTST-M calculation: by means of the interpolation of the curvature components and evaluation of the effective mass, or by means of the direct interpolation of the effective mass, specifically, of the term inside the exponential of the equation 16 in Ref. IVTST-M. The latter option can be selected by means of the INTMUEFF switch keyword, but the former option (NOINTMUEFF) is recommended and it is set as the default option.

KIES (NOKIES)

In a calculation of kinetic isotope effects the values of the reaction coordinate, s , calculated for one isotopomer are not valid for another isotopomer, since the values of s are defined in a mass-scaled space. The KIES keyword allows the calculation of the kinetic-isotope-effects-specific values of s . These values are calculated as the straight line between two consecutive points in the isotope-specific mass-scaled space. Therefore, if the KIES keyword is on, the values of s included in the fu31 input file will only be used for ordering the points along the reaction path according to the standard procedure: first the points with negative s , then the saddle point, and then the points with positive s , ordered in increasing value of s . The geometries in file fu31 must be given in unscaled coordinates (GEOM = *uns*). Nevertheless, if the calculation of new values of s gives place to a permutation in the position of two points (which will not occur as a consequence of just a change in an atomic mass) the program will not detect it, and the interpolation will probably fail. The default is off (NOKIES) and its use is only recommended for a KIE's calculation with RODS using a the reaction path calculated with a different isotope.

11.D.3. REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and SADDLE sections

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
VMEP	variable	(see Glossary)	value of $V_{MEP}(s)$
SMEP	variable	(see Glossary)	value of s
GEOM	list	(see Glossary)	geometries
GRADS	list	(see Glossary)	gradients
HESSIAN	list	(see Glossary)	Hessians
ENERXN	variable	(see Glossary)	reaction energy
ENERSAD	variable	(see Glossary)	barrier height
ENEWELLR	variable	(see Glossary)	energy of the well on the reactant side of the reaction-path
ENEWELLP	variable	(see Glossary)	energy of the well on the product side of the reaction-path

Glossary of REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and SADDLE keywords

ENERSAD

ENERSAD is a variable keyword that indicates the value of the barrier height, in atomic units (measured by taking the zero of energy at reactants). Its only acceptable position is in the SADDLE section, where it is required.

ENERWELLR

ENERWELLR is a variable keyword that indicates the energy of the well on the reactant side of the reaction path, in atomic units (measured by taking the zero of energy at reactants). Its only acceptable position is in the WELLR section.

ENERWELLP

ENERWELLP is a variable keyword that indicates the energy of the well on the product side of the reaction path, in atomic units (measured by taking the zero of energy at reactants). Its only acceptable position is in the WELLP section.

ENERXN

ENERXN is a variable keyword that indicates the reaction endoergicity, in atomic units (measured by taking the zero of energy at reactants). Its only acceptable positions are the PROD2 section for reactions with two products, and the PROD1 section for reactions with one product.

GEOM

GEOM is a list keyword that specifies the Cartesian coordinates at the current grid point. Its only acceptable position is the POINT sections, where it is required; the geometries for the stationary points are read from the fu5 input file.

GRADS

GRADS is a list keyword that specifies the Cartesian gradients (or forces) at the current grid point. Its only acceptable position is the POINT sections, where it is required, since the gradients for the stationary points are assumed to be zero.

HESS

HESS is a list keyword that specifies the Cartesian Hessian matrices. It is required in all the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and SADDLE sections (if they exist and they are not atomic species). It is not required in all the POINT sections, but there must be at least one POINT section with Hessian information on each side of the reaction path.

SMEP

SMEP is a variable keyword that specifies the value of the reaction coordinate s at the current grid point, in mass-scaled atomic units. This keyword is required and only accepted in the POINT sections; for reactants, products, and wells the value of s is calculated as described in Ref. 48 in Sect. 21. If the KIES keyword is on, the values in SMEP are only used for ordering the points, but they could take any value that allow the program to sort the points in its right order.

VMEP

VMEP is a variable keyword that specifies the value of $V_{\text{MEP}}(s)$ at the current grid point, in atomic units and setting the zero of energy at reactants. This keyword is required and only accepted in the POINT sections.

11.E. General description of file fu40 input

This file is needed only if the POTENTIAL keyword of the ENERGETICS section of file fu5 is equal to *unit40*. The input from FORTRAN unit fu40 is in free format keyword style. All input sections must start with the section name preceded by an “*”.

11.E.1. Section Descriptions

<u>Section Name</u>	<u>Description</u>
*GEN40	general data such as options for input and extrapolation
*EXTRAP40	extrapolation information for the reactants and products
*R140	information for the first reactant
*R240	information for the second reactant
*P140	information for the first product
*P240	information for the second product
*WR40	information for reactant side well
*WP40	information for product side well
*SADDLE40	information for the saddle point
*POINT40	information for a point along the MEP

The POINT40 section should be repeated once for each of the points along the MEP excluding the reactant limit, product limit, points used for extrapolation, and saddle point; i.e., POINT40 should be repeated once for each of the points $s_1, s_2, s_3, \dots, s_N$ (in the notation of pages 96-99 of Ref. 14 of section 21) except the saddle point, and if there is a saddle point, that point should not be specified in a POINT40 section. Each section appears at most once except POINT40, which appears NPTS times. The sections can be in any order except that the POINT40 sections must be in order of increasing s . The saddle point, if any, is specified in the SADDLE40 section. In reading in gradients and Hessians, the coordinates are always in this order: x, y, z of atom 1, x, y, z of atom 2, etc.

11.E.2 GEN40 section

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
TITLE	list	blank	job title or comment
PRINT	variable	<i>noextra</i>	print options
MAXLPTS	variable	3	maximum number of points used in Lagrange interpolation
GEOMUNIT	variable	<i>bohr</i>	unit for coordinate input
GRADUNIT	variable	<i>bohr</i>	unit for gradient input
FREQSOURCE	variable	<i>read</i>	option for obtaining frequencies
FREQUNIT	variable	<i>waven</i>	option for frequency input
HESSFORM	variable	<i>full</i>	form of Hessian matrix
HESSUNIT	variable	<i>bohr</i>	option for Hessian matrix input
ANHARM	variable	<i>none</i>	option for anharmonicity input
CURVATURE	variable	<i>setzero</i>	curvature calculation options
GRADDER	variable	<i>noextra</i>	option for computing the derivative of the gradient
NPTS	variable	2	number of points along the MEP at which data is to be read

Glossary of GEN40 keywords

ANHARM

ANHARM is a variable keyword for selecting anharmonicity input.

<u>Option</u>	<u>Definition</u>
<i>none</i>	No anharmonic data is read (default).
<i>cubic</i>	Read the third derivatives k_3 with respect to the normal coordinates at each grid point.
<i>quartic</i>	Also read in fourth derivatives k_4 with respect to the normal coordinates at each grid point.

Example:

```
ANHARM cubic
```

CURVATURE

CURVATURE is a variable keyword used to select the way in which information about the curvature components B_{mF} is input. For the *compute* option to be used, *freqsource* must be set to *hessian* (and the Hessians must be provided in SADDLE40 and POINT40).

<u>Option</u>	<u>Definition</u>
<i>setzero</i>	The B_{mF} are set to zero everywhere (default).
<i>read</i>	The B_{mF} are read directly.
<i>compute</i>	The B_{mF} are computed from the gradient and Hessian as discussed in Sections 6.H and 10.F.

Example:

```
CURVATURE read
```

FREQSOURCE

FREQSOURCE is a variable keyword for selecting how the frequencies should be determined.

<u>Option</u>	<u>Definition</u>
<i>read</i>	Frequencies are read directly (default).
<i>hessian</i>	Frequencies are obtained by diagonalizing the Hessian or projected Hessian matrix.

Example:

```
FREQSOURCE hessian
```

FREQUNIT

FREQUNIT is a variable keyword that specifies the units of the input frequencies.

<u>Option</u>	<u>Definition</u>
<i>waven</i>	Frequencies are given in cm^{-1} (default)
<i>au</i>	Frequencies are given in atomic units.

Example:

```
FREQUNIT au
```

Note: all frequencies must be input in the same units (all wavenumbers or all atomic units), including the imaginary frequency, WSTAR.

GEOMUNIT

GEOMUNIT is a variable keyword that specifies the units of the Cartesian or mass-scaled Cartesian coordinates. If one of the mass-scaled options is chosen, then the scaling mass μ must be specified in unit fu5 by the variable SCALEMASS in section PATH. If $\mu = 1$ amu is used for scaling, then SCALEMASS does not need to be specified since 1 amu is the default.

<u>Option</u>	<u>Definition</u>
	coordinates are in:
<i>bohr</i>	unscaled bohrs (default)
<i>msbohr</i>	mass-scaled bohrs
<i>ang</i>	unscaled angstroms
<i>msang</i>	mass-scaled angstroms
<i>si</i>	unscaled meters

Example:

```
GEOMUNIT ang
```

GRADDER

GRADDER is a variable keyword used to select how the derivative of the gradient is obtained for use in calculation of the curvature components B_{mF} . This keyword is utilized only if CURVATURE is set to *compute*.

<u>Option</u>	<u>Definition</u>
<i>noextra</i>	Derivative of the gradient with respect to s is obtained from a quadratic fit of the gradient at three points on the basic input grid, with the middle point being the current point on the Hessian grid (default). In the notation of Section 9.C, <i>noextra</i> means that the gradient grid is the basic input grid.
<i>oneextra</i>	Derivative of the gradient is computed by one-sided finite difference using the gradient on the basic input grid and a gradient input at one extra point near the basic input grid point. <i>oneextra</i> means that the gradient grid has twice as many points as the basic input grid.
<i>twoextra</i>	Derivative of the gradient is computed from a quadratic fit obtained using the gradient on the basic input grid and gradients input at an extra point near and on each side of the basic input grid point.

Example:

```
GRADDER  oneextra
```

GRADUNIT

GRADUNIT is a variable keyword that specifies the units of the gradients (or, if forces are used, the unit of the force).

<u>Option</u>	<u>Definition</u>
	gradients are in:
<i>bohr</i>	hartrees per unscaled bohr (default)
<i>msbohr</i>	hartrees per mass-scaled bohr
<i>ang</i>	hartrees per unscaled Å
<i>msang</i>	hartrees per mass-scaled Å
<i>si</i>	joules per unscaled meter

Example:

```
GRADUNIT  msbohr
```

HESSFORM

HESSFORM is a variable keyword that, if *hessian* is specified in FREQSOURCE, specifies the form of the Hessian matrix.

<u>Option</u>	<u>Definition</u>
<i>full</i>	F is input as the full matrix (default).
<i>packed</i>	F is input in packed form.

Example:

```
HESSFORM  packed
```

If *packed* is selected, the lower triangular elements of **F** should be listed in order such that $F_{ij} = P(\frac{1}{2}i(i-1) + j)$, where $P(k)$ is element k of the input list and F_{ij} is the element of **F** in row i and column j . The number of elements listed per line is arbitrary. Examples of both input forms for a 6 x 6 matrix is shown below:

Full matrix - example 1

```
F11 F12 F13 F14 F15 F16
F21 F22 F23 F24 F25 F26
F31 F32 F33 F34 F35 F36
F41 F42 F43 F44 F45 F46
F51 F52 F53 F54 F55 F56
F61 F62 F63 F64 F65 F66
```

Full matrix - example 2

```
F11 F12 F13 F14 F15 F16 F21 F22 F23
F24 F25 F26 F31 F32 F33 F34 F35 F36
F41 F42 F43 F44 F45 F46 F51 F52 F53
F54 F55 F56 F61 F62 F63 F64 F65 F66
```

Packed matrix - example 1

```
F11
F21 F22
F31 F32 F33
F41 F42 F43 F44
F51 F52 F53 F54 F55
F61 F62 F63 F64 F65 F66
```

Packed matrix - example 2

```
F11 F21 F22 F31 F32 F33
F41 F42 F43 F44 F51 F52
F53 F54 F55 F61 F62 F63
F64 F65 F66
```

The order of the coordinates is explained in Section 11.E.1.

Note that GAUSSIAN94 writes the packed Hessian matrix in a slightly different way in the standard output file, but the Hessian in the formatted checkpoint file (requested by their FormCheck keyword) is in precisely the packed form accepted by POLYRATE and so is convenient for copying and pasting into POLYRATE fu40 files.

Note that the exact **F** matrix is symmetric, but due to finite precision in computations, the actual matrix one has to input is usually nonsymmetric. If one inputs a nonsymmetric full matrix, the code uses the unsymmetric matrix. If a user wants the code to use a symmetric matrix, we recommend taking an arithmetic average of F_{ij} and F_{ji} for the resulting symmetrized elements in packed form.

HESSUNIT

HESSUNIT is a variable keyword that, if *hessian* is specified in FREQSOURCE, specifies the units of the Hessian matrix **F**. The choice of coordinates (mass-scaled or not) must be consistent with GEOMUNIT.

<u>Option</u>	<u>Definition</u>
	Hessians are in:
<i>bohr</i>	hartrees per unscaled bohr ² (default)
<i>msbohr</i>	hartrees per mass-scaled bohr ²
<i>ang</i>	hartrees per unscaled Å ²
<i>msang</i>	hartrees per mass-scaled Å ²
<i>si</i>	joules per unscaled meter ²

Example:

```
HESSUNIT msbohr
```

Users calculating Hessians by an electronic structure package should read the manual for that package to see what units the Hessian is in. For example, GAUSSIAN94 prints the Hessian in hartrees per unscaled bohr².

MAXLPTS

MAXLPTS is a variable keyword with an integer argument that specifies the maximum number of points used in Lagrange interpolation in the procedure of Subsection 9.C. The default value for this keyword is 3. If this keyword is set to -1 , the program will perform an IVTST-M calculation using the information read from the fu40 input file. When using file fu40, the user cannot select any options for the IVTST-M calculation; it will be performed according to the default options, and the number of gradient and Hessian points have to be the same and equal to NPTS (see below). Therefore, only IVTST-M-NPTS/NPTS calculations are allowed. The curvature components will be interpolated using a value of 3 for the exponent in the mapping function. For unimolecular reactions or reactions with wells, a cubic polynomial fitting to V is done previous to its spline fitting, taking a step of $0.01 a_0$ along the lowest frequency mode for performing this fitting. The only way to change any of these options or to use a different number of gradient and Hessian points is by using the fu31 input file, which is specifically designed for IVTST-M calculations.

NPTS

NPTS is a variable keyword with an integer argument that specifies the number of points along the MEP (excluding the reactant limit, product limit, points used for extrapolation, and saddle point) at which data is to be read. Unless only TST reaction rates are calculated, file fu40 must contain a minimum of one point on each side of the saddle point. Unless NEXTPT is on, these points must be arranged in order of increasing s . The section POINT40 should occur NPTS times in the file 40 input. The default value for NPTS is 2.

PRINT

PRINT is a variable keyword for selecting print options.

<u>Option</u>	<u>Definition</u>
<i>noextra</i>	Print out titles and options (default).
<i>summary</i>	Also print out summary.
<i>input</i>	Also echo input as it is read.
<i>intermediate</i>	Also print out intermediate information about fitting exponential functions and print out interpolated results each time RPHINT is called.
<i>debugging</i>	Also print out debugging information in exponential fitting.

Example:

```
PRINT summary
```

TITLE

TITLE is a list keyword that allows the user to give a one-line title to the electronic structure file.

11.E.3 R140, R240, P140, P240, WR40, and WP40 sections

The R140, R240, P140, P240, WR40, and WP40 sections all have the same set of keywords. A separate section is needed for each reactant, each product, and each well.

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
SVALUE	variable	<i>dgr</i> (or <i>dgp</i>)	value of <i>s</i> (in a.u.) at the reactant (or product)
VVALUE	variable	(required)	potential (in a.u.) at reactant (or product)
HESSIAN	list	(see Glossary)	Hessian matrix for the reactant or product species
IFREQ	list	<i>1 2 3 ...</i>	indices for reordering frequencies, anharmonicities, and curvature components
FREQ	list	(see Glossary)	vibrational frequencies for the reactant or product species
3RDDRIV	list	(see Glossary)	third derivatives of <i>V</i> with respect to motion along the normal coordinates for the reactant or product
4THDERIV	list	(see Glossary)	fourth derivatives of <i>V</i> with respect to motion along the normal coordinates for the reactant or product

Glossary of R140, R240, P140, P240, WR40, and WP40 keywords

3RDDERIV

3RDDERIV is a list keyword that is used to specify the third derivatives of V along the normal coordinates at the reactant, product, or well. If ANHARM is set to *cubic* or *quartic* in the GEN40 section, this keyword is required unless the reactant or product under consideration is monatomic.

4THDERIV

4THDERIV is a list keyword that is used to specify the fourth derivatives of V along the normal coordinates at the reactant, product, or well. If ANHARM is set to *quartic* in the GEN40 section, this keyword is required unless the reactant or product under consideration is monatomic.

FREQ

FREQ is a list keyword that is used to specify the frequencies for each normal mode at the reactant, product, or well species. This keyword is required if FREQSOURCE is set to *read* (the default value) in the GEN40 section and is not used otherwise. The units of the frequencies are determined by the FREQUNIT keyword in the GEN40 section.

HESSIAN

HESSIAN is a list keyword that is used to specify the Hessian matrix at the reactant, product, or well species. This keyword is required if FREQSOURCE is set to *hessian* in the GEN40 section and is not used otherwise. The form and units of the Hessian matrix are determined by the HESSFORM and HESSUNIT keywords in the GEN40 section, respectively.

IFREQ

IFREQ is a required list keyword that is used to specify indices for reordering frequencies, anharmonicities, and curvature components in the interpolation table in order to make the ordering consistent with that at the saddle point and/or along the MEP. The imaginary frequency at the saddle point is not included in this ordering. Therefore, index 1 corresponds to the largest frequency at the saddle point, index 2 corresponds to the second largest frequency, and so on.

SVALUE

SVALUE is a variable keyword that specifies the value of s at the reactant if the reaction is unimolecular or at the well on the reactant side or the value of s at the product if there is only one product or at the well on the product side. The argument of SVALUE can be either a numerical value (in bohr), *dgr* (default guess for reactant or well), or *dgp* (default guess for product or well). SVALUE for reactants has meaning only for a unimolecular reaction, and SVALUE for products has meaning only for a unimolecular rearrangement or an association reaction. SVALUE should be omitted for the reactants if there are two or more reactants, and SVALUE should be omitted for the products if there are two or more products. In the following definitions, s^R is the value of s at the reactant or well on the reactant side, and s^P is the value of s at the product or well on the product side.

<u>Option</u>	<u>Definition</u>
(numerical value)	value of s^R (R140 or WR40 section) or s^P (P140 or WP40 section)
<i>dgr</i>	$s = \max \begin{cases} -2.0a_0 \\ s_1 - (s_2 - s_1) \end{cases}$
<i>dgp</i>	$s = \max \begin{cases} 2.0a_0 \\ s_N + (s_N - s_{N-1}) \end{cases}$

VVALUE

VVALUE is a required variable keyword that specifies the value of the potential (in a.u.) at the reactant or product.

11.E.4 SADDLE40 section

The SADDLE40 section is used to provide information at the saddle point, if it exists. This section should be omitted if there is no saddle point.

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
VVALUE	variable	(required)	potential at saddle point
IFREQ	list	1 2 3 ...	indices for reordering frequencies, anharmonicities, and curvature components
HESSIAN	list	(see Glossary)	Hessian matrix for the saddle point
FREQ	list	(see Glossary)	vibrational frequencies for the saddle point
3RDDRIV	list	(see Glossary)	third derivatives of V with respect to motion along the normal coordinates for the saddle point
4THDERIV	list	(see Glossary)	fourth derivatives of V with respect to motion along the normal coordinates for the saddle point
WSTAR	list	(see Glossary)	absolute value of the imaginary frequency at the saddle point

Glossary of SADDLE40 keywords

3RDDERIV

3RDDERIV is a list keyword that is used to specify the third derivatives of V along the bound normal coordinates at the saddle point. This keyword is required if ANHARM is set to *cubic* or *quartic* in the GEN40 section.

4THDERIV

4THDERIV is a list keyword that is used to specify the fourth derivatives of V along the bound normal coordinates at the saddle point. This keyword is required if ANHARM is set to *quartic* in the GEN40 section.

FREQ

FREQ is a list keyword that is used to specify the frequencies for each bound normal mode at the saddle point. This keyword is required if FREQSOURCE is set to *read* (the default value) in the GEN40 section and is not used otherwise. The units of the frequencies are determined by the FREQUIT keyword in the GEN40 section.

HESSIAN

HESSIAN is a list keyword that is used to specify the Hessian matrix at the saddle point. This keyword is required if FREQSOURCE is set to *hessian* in the GEN40 section and is not used otherwise. The form and units of the Hessian are determined by the HESSFORM and HESSUNIT keywords in the GEN40 section, respectively.

IFREQ

IFREQ is a required list keyword that is used to specify indices for reordering frequencies, anharmonicities, and curvature components in the interpolation table for the saddle point.

VVALUE

VVALUE is a required variable keyword that specifies the value of the potential (in a.u.) at the saddle point.

WSTAR

WSTAR is a variable keyword that is used to specify the value of the imaginary frequency at the saddle point. This keyword is required if FREQSOURCE is set to *read* (the default value) in the GEN40 section and is not used otherwise. The default unit for the imaginary frequency is cm^{-1} , but can be changed to atomic units by setting FREQUIT to *au*.

11.E.5 POINT40 section

This section should be repeated for each point along the MEP except for the reactant limit, product limit, points used for extrapolation, and saddle point. The points must be arranged starting from the one with the most negative s value and continuing in the direction of increasing s .

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
SMEP	variable	(required)	value of s at the current grid location along the MEP
VMEP	variable	(required)	potential at the current grid location on the MEP
GEOM	list	(required)	Cartesian coordinates at the current grid point
GRADS	list	(required)	gradient vector at s along MEP
FORCES	list	(required)	force vector at s along MEP
SB1	variable	(see Glossary)	value of s at a location along the MEP near the current grid point
GRADSB1	list	(see Glossary)	gradient vector at the extra point SB1 for computing the curvature B
FORCESB1	list	(see Glossary)	force vector at the extra point SB1 for computing the curvature B
SB2	variable	(see Glossary)	value of s at another location along the MEP near the current grid point
GRADSB2	list	(see Glossary)	gradient vector at the extra point SB2 for computing the curvature B
FORCESB2	list	(see Glossary)	force vector at the extra point SB2 for computing the curvature B
IFREQ	list	1 2 3 ...	indices for reordering frequencies, anharmonicities, and curvature components

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
HESSIAN	list	(see Glossary)	Hessian matrix at the current grid point
FREQ	list	(see Glossary)	vibrational frequencies at the current grid point
3RDDRIV	list	(see Glossary)	third derivatives of V with respect to motion along the normal coordinates for the current grid point
4THDERIV	list	(see Glossary)	fourth derivatives of V with respect to motion along the normal coordinates for the current grid point
BFS	list	(see Glossary)	reaction path curvature components at the current grid point along the MEP

Glossary of POINT40 keywords

3RDDERIV

3RDDERIV is a list keyword that is used to specify the third derivatives of V along the bound normal coordinates at the current grid point. This keyword is required if ANHARM is set to *cubic* or *quartic* in the GEN40 section.

4THDERIV

4THDERIV is a list keyword that is used to specify the fourth derivatives of V along the bound normal coordinates at the current grid point. This keyword is required if ANHARM is set to *quartic* in the GEN40 section.

BFS

BFS is a list keyword that is used to specify the reaction path curvature components at the current grid point along the MEP. This keyword is required if CURVATURE is set to *read* in the GEN40 section and is not used otherwise.

FORCES

FORCES is a list keyword that is used to specify the forces at the current grid point. This keyword is required if GRADS is not used. The units of the force are determined by the GRADUNIT keyword in the GEN40 section.

FORCESB1

FORCESB1 is a list keyword that is used to specify the force at $s = SB1$. This keyword is required if GRADDER is set to *oneextra* or *twoextra* in the GEN40 section and GRADSB1 is not used; otherwise, it is not used.

FORCESB2

FORCESB2 is a list keyword that is used to specify the force at $s = SB2$. This keyword is required if GRADDER is set to *twoextra* in the GEN40 section and GRADSB2 is not used; otherwise, it is not used.

FREQ

FREQ is a list keyword that is used to specify the frequencies for each bound generalized normal mode at the current grid point. This keyword is required if FREQSOURCE is set to *read* (the default value) in the GEN40 section and is not used otherwise. The units of the frequencies are determined by the FREQUIT keyword in the GEN40 section.

GEOM

GEOM is a list keyword that specifies the Cartesian coordinates at the current grid location, in units determined by the GEOMUNIT keyword in the GEN40 section.

GRADS

GRADS is a list keyword that is used to specify the gradient at the current grid point. This keyword is required if FORCES is not used. The units of the gradient are determined by the GRADUNIT keyword in the GEN40 section.

GRADSB1

GRADSB1 is a list keyword that is used to specify the gradient at $s = SB1$. This keyword is required if GRADDER is set to *oneextra* or *twoextra* in the GEN40 section and FORCESB1 is not used; otherwise, it is not used.

GRADSB2

GRADSB2 is a list keyword that is used to specify the gradient at $s = SB2$. This keyword is required if GRADDER is set to *twoextra* in the GEN40 section and FORCESB2 is not used; otherwise, it is not used.

HESSIAN

HESSIAN is a list keyword that is used to specify the Hessian matrix at the current grid point. This keyword is required if FREQSOURCE is set to *hessian* in the GEN40 section and is not used otherwise. The form and units of the Hessian are determined by the HESSFORM and HESSUNIT keywords in the GEN40 section, respectively.

IFREQ

IFREQ is a list keyword that is used to specify indices for reordering frequencies, anharmonicities, and curvature components in the interpolation table for the current grid point.

SMEP

SMEP is a required variable keyword that specifies the value of s (in bohr) at the current grid location along the MEP.

SB1

SB1 is a variable keyword that specifies the value of s (in bohr) at a grid location along the MEP close to SMEP. This keyword is required if GRADDER is set to *oneextra* or *twoextra* in the GEN40 section and is not used otherwise.

SB2

SB2 is a variable keyword that specifies the value of s (in bohr) at a grid location along the MEP close to SMEP, but on the opposite side of SMEP from SB1. This keyword is required if GRADDER is set to *twoextra* in the GEN40 section and is not used otherwise.

VMEP

VMEP is a required variable keyword that specifies the value of the potential at the current grid location.

11.F. General description of unit fu50 input

This file is only needed if the DL keyword is set to *ioC* in the GENERAL section of the unit fu5 input. This allows one to calculate zero-order interpolated VTST or interpolated optimized corrections. The input from FORTRAN unit fu50 is in free format keyword style. In the current version of POLYRATE there is only one section, VTSTIC, in the unit fu50 input file, and the first keyword in this file must be *VTSTIC. A sample unit fu50 input file is cmcic.dat for the cmctr2 test run. A description of the input to the POLYRATE code from FORTRAN unit fu50 follows.

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
OIVTST	switch	off	zero-order IVTST calculation
(NO)DETM	list	no correction	corrected det I
ENERXN	variable	required	corrected exoergicity
ENESAD	variable	required	corrected classical barrier height
(NO)FREQMAT	list	no matching	frequency matching information
(NO)HRMI	variable	99	correct the moment of inertia
FREQIMAG	variable	required	imaginary freq. at saddle point
LOWFREQ	list	no correction	IVTST treatment of low freq.
MEPTYPEP	variable	two	type of reactant-side MEP
MEPTYPEP	variable	two	type of product-side MEP
MIRP	list	not used	corrected moments of inertia for the reactant and product
MITS	list	not used	corrected moments of inertia for the transition state
LCTCOR	variable	no	correction of <i>V</i> in the nonadiabatic region in LCT calculations

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
PCFREQA	list	not used	corrected frequencies of the product-side stationary point
PCFREQS	list	not used	lower-level frequencies of the product-side stationary point
PCINFO	list	not used	product-side stationary point info.
SPC	variable	not used	<i>s</i> value at this point
ENEPCS	variable	not used	lower-level <i>V</i> at this point
ENEPCA	variable	not used	corrected <i>V</i> at this point
(NO)PRODFREQ	list	not used	corrected product frequencies
RCFREQA	list	not used	corrected frequencies of the reactant-side stationary point
RCFREQS	list	not used	lower-level frequencies of the reactant-side stationary point
RCINFO	list	not used	reactant-side stationary point info.
SRC	variable	not used	<i>s</i> value at this point
ENERCS	variable	not used	lower-level <i>V</i> at this point
ENERCA	variable	not used	corrected <i>V</i> at this point
RANGEIC	list	not used	range parameters
RPL	variable	1.0	range parameter L for an Eckart function
BR	variable	5.0	range parameter for a cut-off gaussian on the reactant side
BP	variable	5.0	range parameter for a cut-off gaussian on the product side
REACFREQ	list	not used	corrected reactant frequencies
SADFREQ	list	not used	corrected saddle point frequencies
UNITMI	variable	<i>au</i>	units used for det I input

Glossary for file fu50 keywords

OIVTST

OIVTST is a switch keyword that specifies that the zero-order IVTST calculation will be performed. The default is off.

DETM I (NODETM I)

DETM I is a list keyword used to input the corrected determinant of the moment of inertia tensor. Five values of det **I** (which is the determinant of the inertia tensor, also equal to the product of the three principal moments of inertia) have to be input. They are, in the order of input, for the first reactant species, the second reactant species, the saddle point, the first product species, and the second product species. The units used are specified by the UNITM I keyword, which is discussed later in this section. For an atomic species, the user should input a value of zero for det **I**. The default is NODETM I, i.e., no correction for det **I**.

Example:

```
DETM I
  5.7220E16
  4.9906E13
  9.8871E17
  5.9964E16
  2.3769E13
END
```

ENERXN

ENERXN is a variable keyword used to input the corrected energy of the reaction in kcal/mol, excluding zero point contributions. This is a required keyword. It should be positive if the reaction is endoergic and negative if it is exoergic.

Example:

```
ENERXN      -5.0
```

ENESAD

ENESAD is a variable keyword used to input the corrected classical barrier height in kcal/mol. This is a required keyword.

Example:

```
ENESAD      6.0
```

FREQMAT (NOFREQMAT)

FREQMAT is a list keyword used to specify the frequency matching information. The input consists of n lines where n is the number of real saddle point frequencies. Each line needs three numbers; the first one is the mode number of a mode at the saddle point, and the second and the third ones are the mode numbers of the modes at the reactant-side and product-side stationary points which match that saddle point mode. The default is NOFREQMAT, i.e., the vibrational frequencies are automatically matched in decreasing order. However, if the keywords MEPTYPER and MEPTYPEP, which are discussed later in this section, are not both equal to $\tau\omega$, this keyword is required.

Example:

```
FREQMAT
  1      2      1
  2      3      3
  3      4      2
  4      1      4
  5      5      5
  6      6      6
  7      8      7
  8      7      8
  9      9      9
END
```

HRMI (NOHRMI)

HRMI is a variable keyword that specifies the correction method for the moment of inertia of hindered-rotor modes. The default is NOHRMI. The valid arguments for the HRMI keyword are:

<u>Option</u>	<u>Description</u>
<i>all</i>	correct the moments of inertia of all the hindered-rotor modes of the reactants, products, and saddle point.
$-n$ ($n=1, 2, \dots$)	same as above, and set the moments of inertia of the lowest n GTS hindered-rotor modes for all values of s to their values at the saddle point.
n ($n=1, 2, \dots$)	no correction to the moment of inertia at stationary points, but set the moments of inertia of the lowest n GTS hindered-rotor modes for all values of s to their lower-level values at the saddle point.

Example:

```
HRMI  -1
```

FREQIMAG

FREQIMAG is a variable keyword used to input the absolute value of the corrected imaginary frequency at saddle point in cm^{-1} . This is a required keyword.

Example:

```
FREQIMAG  900.0
```

LCTCOR

LCTCOR is a variable keyword that specifies the method to correct the potential energy in the nonadiabatic region in LCT calculations. This keyword has an effect only when LCT calculations are performed. The default is *quadlin*, which provides the standard version of the VTST-IC method as explained in Section 3.1 of Ref. 36 of Sect. 21. The valid arguments for this keyword are:

<u>Option</u>	<u>Description</u>
<i>quadlin</i>	linear or quadratic correction depending on the magnitude of the corrections in different regions
<i>linear</i>	linear correction only
<i>nocor</i>	no correction

Example:

```
LCTCOR      nocor
```

Note: The user should take special care to make sure that this correction yields physical results.

LOWFREQ

LOWFREQ is a list keyword used to specify that certain GTS mode frequencies are to be interpolated using the IVTST-0 formula based on their values at three points. The functional forms used to interpolate the specified frequencies are the same as the IVTST-0 functions as explained in Section 4.1 of Ref. 36 of Sect. 21. Notice that this is similar to the IVTST0FREQ keyword of the PATH section of fu5. IVTST0FREQ is used for single-level calculations and lower levels of dual-level calculations, whereas LOWFREQ is used for the higher level of dual-level runs. If, for a given mode, IVTST0FREQ and LOWFREQ are both specified, then LOWFREQ overrides IVTST0FREQ.

Example:

```
LOWFREQ
  11
  12
  13
  14
END
```

MEPTYPER

MEPTYPER is a variable keyword that specifies the type of reactant-side MEP that will be considered in the VTST-IOC calculation. The default value is *two*. The valid arguments for this keyword are:

<u>Option</u>	<u>Description</u>
<i>one</i>	There is only one reactant species and interpolation is carried out between this species and the saddle point.
<i>two</i>	There are two reactant species and interpolation is carried out between these species (using the merged set of frequencies) and the saddle point.
<i>well</i>	The correction on the reactant side is based on a well.

Example:

```
MEPTYPER one
```

MEPTYPEP

MEPTYPEP is a variable keyword that specifies the type of product-side MEP that will be considered in the VTST-IOC calculation. The default is *two*. The valid arguments for this keyword are:

<u>Option</u>	<u>Description</u>
<i>one</i>	There is only one product species and interpolation is carried out between this species and the saddle point.
<i>two</i>	There are two product species and interpolation is carried out between these species (using the merged set of frequencies) and the saddle point.
<i>well</i>	The correction on the product side is based on a well.

Example:

```
MEPTYPEP well
```

MIRP

MIRP is a list keyword used to input the corrected moments of inertia in atomic units of the reactant and product hindered-rotor modes. This moment of inertia is for the internal rotation, which is not the same as the moment of inertia of the overall system. This keyword is needed if HRMI is set to *all* or is less than zero. The input order is reactant 1, reactant 2 (if any), product 1, and product 2 (if any); and within each reactant and product species the modes should be ordered from the highest frequency hindered-rotor mode to the lowest frequency hindered-rotor mode. This keyword can be omitted if there are no hindered-rotor modes in reactants and products.

Example:

```
MIRP
      960.50
END
```

MITS

MITS is a list keyword used to input the corrected moments of inertia in atomic units of the saddle point hindered-rotor modes. The moment of inertia is for the internal rotation, and it is not the same as the moment of inertia of the overall system. This keyword is needed if HRMI is set to *all* or is less than zero. The input should be ordered from the highest frequency hindered-rotor mode to the lowest frequency hindered-rotor mode. This keyword can be omitted if there are no hindered-rotor modes at the saddle point. In the example below, which is for the CH₄ + OH reaction, the symmetry number for the internal rotation of the CH₃ group around the C–H axis in the TS is 3:

Example:

```
MITS
      7690
END
```

PCFREQA

PCFREQA is a list keyword used to input the corrected frequencies (in cm⁻¹) of the product-side stationary point used for the IOC correction. If the number of real frequencies is less than that of the saddle point (i.e., when MEPTYPEP is *two*), add zeros after the last non-zero frequency to input a total of *n* frequencies, where *n* is the number of real frequencies at the saddle point. There should be *n* + 1 input frequencies if MEPTYPEP is not *two*. The order of the frequencies in the input must match the order of the lower-level frequencies, i.e., the first frequency must correspond physically to the highest lower-level frequency, and the second frequency must correspond physically to the second highest lower-level frequency, and so on.

Example:

```
PCFREQA
 3124.0    2282.0    2282.0    2279.0    1489.0
 1489.0    1287.0    1287.0    1111.0    1072.0
 1033.0    1033.0     565.0     415.0     411.0
   0.0      0.0      0.0      0.0      0.0
END
```

(In this example MEPTYPEP is *two* and there are nine atoms in the system. The first product has four atoms and the second product has five atoms. The saddle point has twenty real frequencies but the products have only fifteen real frequencies, so we add five zeros at the end of input.)

PCFREQS

PCFREQS is a list keyword used to input the lower-level frequencies (in cm^{-1}) of the product-side stationary point. This keyword is only needed when MEPTYPEP is *well*. Otherwise the frequencies are just the product frequencies, and the program will have this information when it calculates the product properties. There should be $n+1$ input frequencies, where n is the number of the real frequencies at the saddle point. The frequencies in the input must be in decreasing order.

Example:

```
PCFREQS
 3124.0   2282.0   2282.0   2279.0   1489.0
 1489.0   1287.0   1287.0   1111.0   1072.0
 1033.0   1033.0    565.0    415.0    411.0
END
```

PCINFO

PCINFO is a list keyword used to specify the product-side stationary point information. This keyword is needed only if MEPTYPEP is *one* or *well*. The valid subkeyword options are defined below. There are no defaults for these subkeywords, and the user must supply all the values, except as noted.

<u>Option</u>	<u>Description</u>
<i>spc</i>	s value in bohrs at this point
<i>enepcs</i>	lower-level potential energy in kcal/mol at this point (required only if MEPTYPEP is <i>well</i>)
<i>enePCA</i>	corrected potential energy in kcal/mol at this point (required only if MEPTYPEP is <i>well</i>)
<i>nofreqcorr</i>	no frequency correction at the product complex; when this keyword is used the keywords PCFREQA and PCFREQS are not needed

Example:

```
PCINFO
  SPC          3.0
  ENEPCS      -10.0
  ENEPCA       -9.0
END
```


PRODFREQ (NOPRODFREQ)

PRODFREQ is a list keyword used to input the corrected product frequencies in cm^{-1} . It is only needed when MEPTYPEP is *well*; otherwise the corrected product frequencies are input in the PCFREQA keyword. NOPRODFREQ specifies that no correction is needed for the product frequencies, but this is allowed only if MEPTYPEP is *well*. The default is to correct the product frequencies when MEPTYPEP is *well*, so if NOPRODFREQ is not specified, the PRODFREQ keyword is required. See keyword PCFREQA for the required input order for the frequencies.

Example:

```
PRODFREQ
    3092.0    3092.0    2981.0    1453.0    1453.0
    1377.0    1017.0    1017.0     741.0
END
```

RCFREQA

RCFREQA is a list keyword used to input the corrected frequencies (in cm^{-1}) of the reactant-side stationary point that the IOC correction is based on. If the number of real frequencies is less than that of the saddle point (i.e., when MEPTYPER is *two*), add zeros after the last non-zero frequency to input a total of n frequencies, where n is the number of real frequencies at the saddle point. There should be $n+1$ input frequencies if MEPTYPER is not *two*. The order of the frequencies in the input must match the order of the lower-level frequencies, i.e., the first frequency must correspond physically to the highest lower-level frequency, and the second frequency must correspond physically to the second highest lower-level frequency, and so on.

Example (see the example for the PCFREQA keyword):

```
RCFREQA
    3124.0    2282.0    2282.0    2279.0    1489.0
    1489.0    1287.0    1287.0    1111.0    1072.0
    1033.0    1033.0     565.0     415.0     411.0
     0.0         0.0         0.0         0.0         0.0
END
```

RCFREQS

RCFREQS is a list keyword used to input the flower-level frequencies (in cm^{-1}) of the reactant-side stationary point. This keyword is only needed when MEPTYPER is *well*. Otherwise the frequencies are just the reactant frequencies, and the program will have this information when it calculates the reactant properties. There should be $n + 1$ input frequencies, where n is the number of the real frequencies at the saddle point. The frequencies in the input must be in decreasing order.

Example:

```
RCFREQS
    3124.0    2282.0    2282.0    2279.0    1489.0
    1489.0    1287.0    1287.0    1111.0    1072.0
    1033.0    1033.0     565.0     415.0     411.0
END
```

RCINFO

RCINFO is a list keyword used to specify the reactant-side stationary point information. This keyword is needed only if MEPTYPEP is *one* or *well*. The valid subkeyword options are defined below. There are no defaults for these subkeywords, and the user must supply all the values, except as noted.

<u>Option</u>	<u>Description</u>
<i>src</i>	<i>s</i> value in bohrs at this point
<i>enercs</i>	lower-level potential energy in kcal/mol at this point (required only if MEPTYPEP is <i>well</i>)
<i>enerca</i>	corrected potential energy in kcal/mol at this point (required only if MEPTYPEP is <i>well</i>)
<i>nofreqcorr</i>	no frequency correction at the reactant complex; when this keyword is used the keywords RCFREQA and RCFREQS are not needed

Example:

```
RCINFO
  SRC      -3.0
  ENERCS   -5.0
  ENERCA   -4.0
END
```

RANGEIC

RANGEIC is a list keyword used to input the range parameters for the correction functions in the VTST-IOC calculations. The valid subkeyword options are defined below.

<u>Option</u>	<u>Description</u>	<u>Default</u>
<i>rpl</i>	range parameter L in bohrs for an Eckart function	1.0
<i>br</i>	range parameter for a cut-off gaussian function on the reactant side	5.0
<i>bp</i>	range parameter for a cut-off gaussian function on the product side	5.0

Example:

```
RANGEIC
  RPL      0.3
  BR      10.0
  BP       8.0
END
```

REACFREQ

REACFREQ is a list keyword used to input the corrected reactant frequencies in cm^{-1} . It is required if MEPTYPE is *well*. Otherwise the reactant frequencies are input in RCFREQA keyword. See keyword RCFREQA for the requirement on the input order of the frequencies.

Example:

```
REACFREQ
  3092.0    3092.0    2981.0    1453.0    1453.0
  1377.0    1017.0    1017.0     741.0
END
```

SADFREQ

SADFREQ is a list keyword used to input the corrected real saddle point frequencies in cm^{-1} . It is a required keyword. See keyword RCFREQA for the required input order for the frequencies.

Example:

```
SADFREQ
  3748.0    3130.0    3259.0    3255.0    1414.0
  1521.0    1494.0    1298.0    1206.0     933.0
  769.0     357.0     316.0     40.0
END
```

UNITMI

UNITMI is a variable keyword that specifies the units used for the DETMI keyword input. The default is *au*. The valid arguments are:

<u>Option</u>	<u>Description</u>
<i>au</i>	atomic units
<i>si</i>	SI units, i.e., kg m^2 or $(\text{kg m}^2)^3$
<i>cgs</i>	c.g.s. units, i.e., g cm^2 or $(\text{g cm}^2)^3$
<i>scale</i>	scaled units, i.e., 10^{-40}g cm^2 or $(10^{-40} \text{g cm}^2)^3$
<i>amu</i>	atomic mass unit, i.e., a.m.u. bohr ² or (a.m.u. bohr ²) ³ .

Example:

```
UNITMI    AMU
```

11.G. General description of unit fu51 input

This file is only needed if the *ispe* option is specified with the DL keyword in the GENERAL section of unit fu5. The single-point energy correction (VTST-ISPE) is used to correct the energies along the reaction path. The input from FORTRAN unit fu51 is in free format keyword style. A description of the input to the POLYRATE code from FORTRAN unit fu51 follows.

11.G.1. Section Descriptions

<u>Section Name</u>	<u>Description</u>
*ISPEGEN	general section for stationary point information
*POINT	section for the extra points along the reaction path

The *POINT section should be repeated once for each of the points along the reaction path excluding the reactants, products, saddle point, and wells.

11.G.2 ISPEGEN section

The *ISPEGEN section is used to provide information at the stationary points.

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
EXERXN	variable	required	corrected exoergicity
ENESAD	variable	required	corrected classical barrier height
MEPTYPEP	variable	<i>two</i>	type of reactant-side MEP
MEPTYPEP	variable	<i>two</i>	type of product-side MEP
PCINFO	list	not used	product-side stationary point data
SPC	variable	not used	value of s at this point
ENEPCS	variable	not used	lower-level V at this point
ENERCA	variable	not used	corrected V at this point
RCINFO	list	not used	reactant-side stationary point data
SRC	variable	not used	value of s at this point
ENERCS	variable	not used	lower-level V at this point
ENERCA	variable	not used	corrected V at this point

Glossary of ISPEGEN keywords

EXERXN

EXERXN is a variable keyword used to input the corrected classical energy change of the reaction in kcal/mol, excluding zero-point contributions. This is a required keyword. It should be positive if the reaction is endoergic and negative if it is exoergic. In other words, this is the corrected product energy relative to the reactant energy.

Example:

```
EXERXN      -5.0
```

ENESAD

ENESAD is a variable keyword used to input the corrected classical barrier height in kcal/mol. This is a required keyword.

Example:

```
ENESAD      6.0
```

MEPTYPEP

MEPTYPEP is a variable keyword that specifies the type of product-side MEP that will be considered in the VTST-ISPE calculation. The default is *two*. The valid arguments for this keyword are:

<u>Option</u>	<u>Description</u>
<i>one</i>	There is one product and wells, if any, are not recognized.
<i>two</i>	There are two products and wells, if any, are not recognized.
<i>well</i>	A product well is recognized.

Example:

```
MEPTYPEP   one
```

MEPTYPER

MEPTYPER is a variable keyword that specifies the type of reactant-side MEP that will be considered in the VTST-ISPE calculation. The default is *two*. The valid arguments for this keyword are:

<u>Option</u>	<u>Description</u>
<i>one</i>	There is one reactant and wells, if any, are not recognized.
<i>two</i>	There are two reactants and wells, if any, are not recognized.
<i>well</i>	A reactant well is recognized.

Example:

```
MEPTYPER   one
```

PCINFO

PCINFO is a list keyword used to specify the product-side stationary point information. This keyword is needed only if MEPTYPEP is set to *one* or *well*. There are no defaults for these options, the user must supply all values except as otherwise noted. The valid options for this keyword are:

<u>Option</u>	<u>Description</u>
<i>spc</i>	value of <i>s</i> , in bohrs, at this point
<i>enepcs</i>	lower-level potential energy, in kcal/mol, at this point with respect to the reactant (required if and only if MEPTYPEP is set to <i>well</i>)
<i>enepca</i>	corrected potential energy, in kcal/mol, at this point with respect to the reactant (required if and only if MEPTYPEP is set to <i>well</i>)

Example:

```
PCINFO
  spc      3.0
  enepcs  -10.0
  enepca   -9.0
```

RCINFO

RCINFO is a list keyword used to specify the reactant-side stationary point information. This keyword is needed only if MEPTYPEP is set to *one* or *well*. There are no defaults for these options, the user must supply all values except as otherwise noted. The valid options for this keyword are:

<u>Option</u>	<u>Description</u>
<i>src</i>	value of <i>s</i> , in bohrs, at this point
<i>enercs</i>	lower-level potential energy, in kcal/mol, at this point with respect to the reactant (required if and only if MEPTYPEP is set to <i>well</i>)
<i>enerca</i>	corrected potential energy, in kcal/mol, at this point with respect to the reactant (required if and only if MEPTYPEP is set to <i>well</i>)

Example:

```
PCINFO
  src      -3.0
  enercs   -5.0
  enerca   -4.0
```

11.G.3 POINT section

The POINT section should be repeated for each point along the MEP, except for the reactants, products, saddle point, and wells. Each new POINT section must begin with *POINT. The points can be arranged in any order.

<u>Keyword</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
DVMEP	variable	variable	difference in the corrected energy at point
SMEP	variable	required	value of s at point
VMEP	variable	variable	corrected energy at point

Notice that in the POINT section, the keyword SMEP is required and either VMEP or DVMEP is needed for indicating the energy at a higher-level or difference of the energy at a higher-level to the lower-level one.

Glossary of POINT keywords

DVMEP

DVMEP is a variable keyword that can be used to specify the energy difference between the higher-level along the MEP from the lower-level ones. Using the VMEP keyword to indicate the higher-level potential energies along the lower-level MEP has a limitation, namely that the point chosen has to be a save-grid point of the lower-level path. However, using the DVMEP keyword, the point chosen does not have to be a save-grid point of the lower-level path because the difference in potential energies from the higher-level to the lower-level is read instead of calculated.

Example:

```
VMEP          5.1608
```

SMEP

SMEP is a variable keyword that specifies the value of s , in bohrs, at the current point along the MEP.

Example:

```
SMEP          -0.8
```


VMEP

VMEP is a variable keyword that specifies the energy at the current point along the MEP.

Example:

VMEP 10.5

12. GENERAL DESCRIPTION OF LCT FILES

In the calculation of LCT probabilities, relevant quantities such as theta integrals (i.e., imaginary action integrals) are evaluated on the grid defined by the input parameters NQTH and NGTHETA. The stored information is then used for interpolation. When extra output of the LCT calculations is chosen (LCTDETAIL keyword), the details of the calculation of these quantities at certain grid points, which are specified by user, are given in files poly.fu41 through poly.fu47. The contents of each output file are listed in the table below. In the table, 'each tunneling path' refers to a tunneling path corresponding to an energy quadrature point, IE, that is within the range specified in the keyword LCTDETAIL in the TUNNEL section of unit fu5 input.

File name	Fortran Unit	Contents
poly.fu41	fu41	the theta integrand and the weighting factor at each point on each tunneling path, within its adiabatic region.
poly.fu42	fu42	the theta integrand, the effective potential, the classical potential, and the potential correction terms, for each point on each tunneling path, in its non-adiabatic region
poly.fu43	fu43	the theta integrals and the boundaries of the nonadiabatic region of each tunneling path
poly.fu44	fu44	the length of the tunneling path
poly.fu45	fu45	the corresponding s value on the MEP for each point on each tunneling path
poly.fu46	fu46	the adiabatic potential and the cosine of the angle between the tunneling path and the gradient vector of the MEP at the s value to which the point on the tunneling path corresponds for each point on each tunneling path
poly.fu47	fu47	the vibrational turning points, the projection onto each normal mode coordinate, and the curvature components for each point on each tunneling path

13. INSTALLING AND USING POLYRATE

A step-by-step procedure for installing, testing, and using POLYRATE is given in this section. The file structure and the C shell scripts should make POLYRATE reasonably easy to install, test, and use.

Subsection 13.A discusses running POLYRATE in a Unix environment, including Linux. Subsection 13.B contains information for users who do not wish to use the provided Unix scripts or who wish to run POLYRATE in a non-Unix environment.

13.A. Instructions for a Unix environment

If the POLYRATE program has been received in tar format, the recommended directory structure will be created automatically when this file is untarred, and the files will be put in their canonical places as listed in Subsection 13.A.1.

13.A.1. Installation

Before you start, make sure all the file names are in lower case. For the purpose of illustration, we use upper case for sample directory names, e.g., PDIR. Here PDIR is the directory above the polyrate9.4.1 directory, i.e., it is the parent directory you are going to install POLYRATE in.

POLYRATE may be downloaded as a single tar file with the following structure.

STEP 1 Verify that the following directory structure was created properly when the file was untarred. Use may use the 'ls -R' command.

```
user@machine [~/polyrate9.4.1] % ls -R
```

```
PC/      doc/      obj/      script/  src/      testrun/
configure*  exe/      poten/    set_machine.jc*  testo/    util/
./PC:
check_test.pl  poly_ch4cl/  poly_ch5/  poly_cwmc/  poly_hni/  poly_oh3/
clean_test.bat  poly_ch4o/  poly_clhbr/  poly_h2ni/  poly_ho2/  poly_ohcl/
clean_update.bat  poly_ch4oh/  poly_cmc/  poly_h3/  poly_nh3/  run_all.bat

./PC/poly_ch4cl:
poly_ch4cl.mak  poly_ch4cl.mdp  run_ch4cl.bat

./PC/poly_ch4o:
poly_ch4o.mak  poly_ch4o.mdp  run_ch4o.bat

./PC/poly_ch4oh:
run_ch4oh.bat

./PC/poly_ch5:
poly_ch5.mak  poly_ch5.mdp  run_ch5.bat  run_lola.bat  run_lola.bat.bak

./PC/poly_clhbr:
poly_clhbr.mak  poly_clhbr.mdp  run_clhbr.bat

./PC/poly_cmc:
poly_cmc.mak  poly_cmc.mdp  run_cmc.bat

./PC/poly_cwmc:
poly_cwmc.mak  poly_cwmc.mdp  run_cwmc.bat

./PC/poly_h2ni:
poly_h2ni.mak  poly_h2ni.mdp  run_h2ni.bat

./PC/poly_h3:
poly_h3.mak  poly_h3.mdp  run_h3.bat

./PC/poly_hni:
hnitrl.dat  hnitrl.fu15  hnitrl.fu6  poly_hni.mak  pothni.dat
hnitrl.fu14  hnitrl.fu5  hnitrl.fu61  poly_hni.mdp  run_hni.bat

./PC/poly_ho2:
poly_ho2.mak  poly_ho2.mdp  run_ho2.bat

./PC/poly_nh3:
poly_nh3.mak  poly_nh3.mdp  run_nh3.bat

./PC/poly_oh3:
poly_oh3.mak  poly_oh3.mdp  run_oh3.bat

./PC/poly_ohcl:
poly_ohcl.mak  poly_ohcl.mdp  run_ohcl.bat
```

```

./doc:

./exe:

./obj:

./poten:
ch4o.f      clhbr.f      dumpot.f      nh3.f      potch5j2.dat  poth2niads.dat surf3.f
ch4o.inc    cmc.f        h2ni.f        oh3.f      potcmcl.dat  poth2nilat.dat surf4.f
ch5.f      coord3.f     h3.f          ohcl.f     potcmc2.dat  pothni.dat
chain3.f    coord4.f     hni.f         potch4o.dat potcwmc3b.dat setup3.f
chain4.f    cwmc.f       ho2.f         potch5j1.dat potcwmcvib.dat setup4.f

./script:
comp_pot.jc* comp_src.jc* exe_chk.jc* link_exe.jc* potcomp.jc*

./src:
acespoly.f  dummy_mpi.f  headr.f       interface.f  param.inc    percon.inc    polyhl.f
acesrate/  ef.f         hooks.f       ivtstm.f    param1.inc   poly31.f      polymq.f
common.inc  energetics.f intbsv1.f     main.f      param2.inc   poly40.f      polyrr.f
dateclock.c fromblas.f   intbsv2.f     morate/     param3.inc   poly_mpi.f    polysz.f
dattim.f   givtst.f    intbsv3.f     mpif.h      param4.inc   polyag.f      rtpjac.f

./src/acesrate:
acespoly.f

./src/morate:
mopoly.f

./testo:
ch4cltr1.fu15  ch5fu40tr4.fu15  ch5j1tr3.fu15  cmctr1.fu15  nh3tr1.fu6  oh3tr10.fu15
ch4cltr2.fu15  ch5fu40tr5.fu15  ch5j1tr4.fu15  cmctr2.fu15  nh3tr2.fu15  oh3tr2.fu15
ch4ohtr1.fu15  ch5fu40tr6.fu15  ch5j1tr5.fu15  cmctr3.fu15  oh3fu30tr1.fu15  oh3tr3.fu15
ch4ohtr2.fu15  ch5fu50tr1.fu15  ch5j1tr6.fu15  cwmctr1.fu15  oh3fu30tr2.fu15  oh3tr4.fu15
ch4otr1.fu15   ch5fu50tr2.fu15  ch5j1tr7.fu15  cwmctr2.fu15  oh3fu30tr3.fu15  oh3tr5.fu15
ch5fu29tr1.fu15  ch5fu50tr3.fu15  ch5j1tr8.fu15  cwmctr3.fu15  oh3fu30tr4.fu6  oh3tr6.fu15
ch5fu29tr2.fu15  ch5fu51tr1.fu15  ch5j1tr9.fu15  h2nitr1.fu15  oh3fu30tr5.fu6  oh3tr7.fu15
ch5fu30tr1.fu15  ch5icfu30tr1.fu15  ch5j2itr1.fu15  h3tr1.fu15  oh3fu30tr6.fu15  oh3tr8.fu15
ch5fu30tr2.fu15  ch5icfu30tr2.fu15  ch5j2itr2.fu15  h3tr2.fu15  oh3fu31tr1.fu15  oh3tr9.fu15
ch5fu30tr3.fu15  ch5icfu31tr1.fu15  ch5j2tr1.fu15  h3tr3.fu15  oh3fu31tr2.fu15  ohcltr1.fu15
ch5fu30tr4.fu15  ch5icfu40tr1.fu15  ch5j2tr2.fu6   h3tr4.fu15  oh3fu40tr1.fu15  ohcltr2.fu15
ch5fu30tr5.fu15  ch5icfu40tr2.fu15  ch5j2tr3.fu15  hnitrl.fu15  oh3fu40tr2.fu15  ohcltr3.fu15
ch5fu30tr6.fu15  ch5j1tr1.fu15    clhbrtr1.fu15  ho2tr1.fu15  oh3fu40tr3.fu15  ohcltr4.fu15
ch5fu31tr1.fu15  ch5j1tr10.fu15   clhbrtr2.fu15  ho2tr2.fu15  oh3fu40tr4.fu6
ch5fu40tr1.fu15  ch5j1tr11.fu15   clhbrtr3.fu15  ho2tr3.fu15  oh3fu40tr5.fu6
ch5fu40tr2.fu15  ch5j1tr12.fu15   clhbrtr4.fu15  ho2tr4.fu15  oh3fu40tr6.fu15
ch5fu40tr3.fu15  ch5j1tr2.fu15    clhbrtr5.fu15  ho2tr5.fu15  oh3tr1.fu15

./testrun:
ch4cl/  check_all.jc* cmc/  hni/  ohcl/  runch4oh.jc*  runewmc.jc*  runho2.jc*
ch4o/   check_test.jc* cwmc/  ho2/  run_all.jc*  runch5.jc*  runh2ni.jc*  runnh3.jc*
ch4oh/  clean_up.jc*  h2ni/  nh3/  runch4cl.jc*  runchlbr.jc*  runh3.jc*  runoh3.jc*
ch5/   clhbr/  h3/   oh3/  runch4o.jc*  runemc.jc*  runhni.jc*  runohcl.jc*

```

./testrun/ch4cl:

ch4cltr1.dat ch4cltr1.fu29 ch4cltr1.jc* ch4cltr2.dat ch4cltr2.fu29 ch4cltr2.jc*

./testrun/ch4o:

ch4otr1.dat ch4otr1.jc*

./testrun/ch4oh:

ch4ohic.dat ch4ohiv.dat ch4ohiv2.dat ch4ohtr1.dat ch4ohtr1.jc* ch4ohtr2.dat ch4ohtr2.fu29
ch4ohtr2.jc*

./testrun/ch5:

ch5fu29tr1.dat ch5fu30tr5.fu30 ch5fu40tr4.jc* ch5icfu30tr1.dat ch5icfu40tr2.jc* ch5j1tr7.dat
ch5fu29tr1.fu29 ch5fu30tr5.jc* ch5fu40tr5.dat ch5icfu30tr1.fu30 ch5j1tr1.dat ch5j1tr7.jc*
ch5fu29tr1.jc* ch5fu30tr6.dat ch5fu40tr5.fu40 ch5icfu30tr1.fu50 ch5j1tr1.jc* ch5j1tr8.dat
ch5fu29tr2.dat ch5fu30tr6.fu30 ch5fu40tr5.jc* ch5icfu30tr1.jc* ch5j1tr10.dat ch5j1tr8.jc*
ch5fu29tr2.fu29 ch5fu30tr6.jc* ch5fu40tr6.dat ch5icfu30tr2.dat ch5j1tr10.jc* ch5j1tr9.dat
ch5fu29tr2.jc* ch5fu31tr1.dat ch5fu40tr6.fu40 ch5icfu30tr2.fu30 ch5j1tr11.dat ch5j1tr9.jc*
ch5fu30tr1.dat ch5fu31tr1.fu31 ch5fu40tr6.jc* ch5icfu30tr2.fu50 ch5j1tr11.jc* ch5j2itr1.dat
ch5fu30tr1.fu30 ch5fu31tr1.jc* ch5fu50tr1.dat ch5icfu30tr2.jc* ch5j1tr12.dat ch5j2itr1.fu29
ch5fu30tr1.jc* ch5fu40tr1.dat ch5fu50tr1.fu50 ch5icfu31tr1.dat ch5j1tr12.jc* ch5j2itr1.jc*
ch5fu30tr2.dat ch5fu40tr1.fu40 ch5fu50tr1.jc* ch5icfu31tr1.fu31 ch5j1tr2.dat ch5j2itr2.dat
ch5fu30tr2.fu30 ch5fu40tr1.jc* ch5fu50tr2.dat ch5icfu31tr1.fu50 ch5j1tr2.jc* ch5j2itr2.fu29
ch5fu30tr2.jc* ch5fu40tr2.dat ch5fu50tr2.fu50 ch5icfu31tr1.jc* ch5j1tr3.dat ch5j2itr2.jc*
ch5fu30tr3.dat ch5fu40tr2.fu40 ch5fu50tr2.jc* ch5icfu40tr1.dat ch5j1tr3.jc* ch5j2itr1.dat
ch5fu30tr3.fu30 ch5fu40tr2.jc* ch5fu50tr3.dat ch5icfu40tr1.fu40 ch5j1tr4.dat ch5j2itr1.jc*
ch5fu30tr3.jc* ch5fu40tr3.dat ch5fu50tr3.fu50 ch5icfu40tr1.fu50 ch5j1tr4.jc* ch5j2itr2.dat
ch5fu30tr4.dat ch5fu40tr3.fu40 ch5fu50tr3.jc* ch5icfu40tr1.jc* ch5j1tr5.dat ch5j2itr2.jc*
ch5fu30tr4.fu30 ch5fu40tr3.jc* ch5fu51tr1.dat ch5icfu40tr2.dat ch5j1tr5.jc* ch5j2itr3.dat
ch5fu30tr4.jc* ch5fu40tr4.dat ch5fu51tr1.fu51 ch5icfu40tr2.fu40 ch5j1tr6.dat ch5j2itr3.jc*
ch5fu30tr5.dat ch5fu40tr4.fu40 ch5fu51tr1.jc* ch5icfu40tr2.fu50 ch5j1tr6.jc*

./testrun/clhbr:

clhbrtr1.dat clhbrtr2.dat clhbrtr3.dat clhbrtr4.dat clhbrtr5.dat
clhbrtr1.jc* clhbrtr2.jc* clhbrtr3.jc* clhbrtr4.jc* clhbrtr5.jc*

./testrun/cmc:

cmcic.dat cmctr1.dat cmctr1.jc* cmctr2.dat cmctr2.jc* cmctr3.dat cmctr3.fu51 cmctr3.jc*

./testrun/cwmc:

cwmctr1.dat cwmctr1.jc* cwmctr2.dat cwmctr2.jc* cwmctr3.dat cwmctr3.jc*

./testrun/h2ni:

h2nitr1.dat h2nitr1.jc*

./testrun/h3:

h3tr1.dat h3tr1.jc* h3tr2.dat h3tr2.jc* h3tr3.dat h3tr3.jc* h3tr4.dat h3tr4.jc*

./testrun/hni:

hnitr1.dat hnitr1.jc*

./testrun/ho2:

ho2ic.dat ho2tr1.jc* ho2tr2.jc* ho2tr4.dat ho2tr5.dat ho2tr1.dat
ho2tr2.dat ho2tr3.dat ho2tr3.jc* ho2tr4.jc* ho2tr5.jc*

```
./testrun/nh3:
nh3ic.dat  nh3tr1.dat  nh3tr1.jc*  nh3tr2.dat  nh3tr2.jc*
```

```
./testrun/oh3:
RWMD.001      oh3fu30tr2.jc*  oh3fu30tr6.fu30  oh3fu40tr2.dat  oh3fu40tr5.jc*  oh3tr3.jc*
oh3tr8.jc*
RWMD.002      oh3fu30tr3.dat  oh3fu30tr6.jc*  oh3fu40tr2.fu40  oh3fu40tr6.dat  oh3tr4.dat
oh3tr9.dat
RWMD.003      oh3fu30tr3.fu30  oh3fu31tr1.dat  oh3fu40tr2.jc*  oh3fu40tr6.fu40  oh3tr4.jc*
oh3tr9.fu31
RWMD.004      oh3fu30tr3.jc*  oh3fu31tr1.fu31  oh3fu40tr3.dat  oh3fu40tr6.jc*  oh3tr5.dat
oh3tr9.jc*
RWMD.005      oh3fu30tr4.dat  oh3fu31tr1.jc*  oh3fu40tr3.fu40  oh3tr1.dat      oh3tr5.jc*
tr7old.fu30
RWMD.006      oh3fu30tr4.fu30  oh3fu31tr2.dat  oh3fu40tr3.jc*  oh3tr1.jc*      oh3tr6.dat
oh3fu30tr1.dat  oh3fu30tr4.jc*  oh3fu31tr2.fu31  oh3fu40tr4.dat  oh3tr10.dat     oh3tr6.jc*
oh3fu30tr1.fu30  oh3fu30tr5.dat  oh3fu31tr2.jc*  oh3fu40tr4.fu40  oh3tr10.jc*     oh3tr7.dat
oh3fu30tr1.jc*  oh3fu30tr5.fu30  oh3fu40tr1.dat  oh3fu40tr4.jc*  oh3tr2.dat      oh3tr7.fu30
oh3fu30tr2.dat  oh3fu30tr5.jc*  oh3fu40tr1.fu40  oh3fu40tr5.dat  oh3tr2.jc*      oh3tr7.jc*
oh3fu30tr2.fu30  oh3fu30tr6.dat  oh3fu40tr1.jc*  oh3fu40tr5.fu40  oh3tr3.dat      oh3tr8.dat
```

```
./testrun/ohcl:
ohcltr1.dat  ohcltr1.jc*  ohcltr2.dat  ohcltr2.jc*  ohcltr3.dat  ohcltr3.jc*  ohcltr4.dat  ohcltr4.jc*
```

```
./util:
ch4.mi      dattim.altix  dattim.f      dattim.ibm    detmi.doc     findl.f       reord30.f
ch4oh.mi    dattim.compaq  dattim.f90    dattim.iris  detmi.f       hrotor.doc    rotate/
cwmc.rmi    dattim.cray   dattim.gnu    dattim.linux  detmi.jc*     hrotor.f
dateclock.c  dattim.dec    dattim.hp     dattim.sun    findb.f       oh.mi
```

```
./util/rotate:
rotate.doc  rotate30/    rotate31/    rotate40/
```

```
./util/rotate/rotate30:
rotate30.f  rotate30.jc*  test30.dat  test30.fu30
```

```
./util/rotate/rotate31:
rotate31.f  rotate31.jc*  test31.dat  test31.fu31
```

```
./util/rotate/rotate40:
rotate40.f  rotate40.jc*  test40.dat  test40.fu40
```

STEP 2 Change your working directory to the polyrate9.4.1 directory and run the script configure by typing `./configure <Return>`. You will be asked if you want this version to be your default version of POLYRATE. If you select yes, then the hidden file `.poly_path` will point to the version that you are installing. Next, the script will automatically detect available compilers and ask if you want to use the detected options. Even if you want to use a different compiler, it is

highly recommended that you select yes and later edit the Makefile (which is found in the src directory.)

- STEP 3 Check the Makefile to ensure that the correct compiler options appear. POLYRATE has a known problem with dynamic memory, so compiler options that force static memory are typically required. For example, the g77 compiler requires the flag '-fno-automatic', and the Portland Group compiler requires the flag '-Msave'. Due to the various combinations of OS and compilers, the configure script does not always print these flags automatically.
- STEP 4 You may proceed directly to testing. The Makefile will compile any files that are needed to execute the testruns. If you want to use your own potential energy surface, you may add it to the Makefile. Comments in the Makefile will guide you when making additions.
- STEP 5 The following two compiling problems are common enough to merit special attention. If you have an error relating to module libc.a(shr.o), it is probably because other modules are interfering with the compilation. Unload any extra modules that are running and recompile. Problems with the date and time files are also relatively common. The configure script tries to pick the correct dattim.f file to use, but sometimes the user must override the choice. The user may search the util directory for the appropriate dattim file, and copy that file to src/dattim.f. Read Subsection 5.A.1 for more information.

13.A.2. Testing

The directory structure and files should be set up as instructed in Subsection 13.A.1. Note that some of the processes described in this section may be time consuming. In a Unix system, you can always add an '&' character at the end of a command line to put the process in the background. In the following instructions, whenever you see a command followed by an '&' character, it means that we suggest that the user run the command in the background.

It is recommended that a user run all the test runs in the distribution package before using POLYRATE. In version 9.4.1, we have made the testing very easy. In fact, we have

included many useful scripts to run the test runs, and you can choose how to run the test runs from among several different approaches.

(1) To run all the test runs:

The test runs in version 9.4.1 are grouped into 14 different sets, in particular, ch4cl, ch4oh, ch5, clhbr, cmc, cwmc, h2ni, h3, hni, ho2, nh3, oh3, and ohcl test runs. (See Subsection 14.C.) Each set has its own subdirectory in the polyrate9.4.1/testrun directory. (See Subsection 13.A.1.) The subdirectories contain the input data files and the job control files for the test runs. The only thing you need to do to run all the test runs is to change the working directory to the polyrate9.4.1/testrun directory and run the script run_all.jc.

```
./run_all.jc & <Return>
```

(or submit it as a batch job)

(2) To run a subset of the test runs

1. To run a set of test runs, use the scripts available in the polyrate9.4.1/testrun directory. For example, to run all the h3 test runs, change the working directory to the polyrate9.4.1/testrun directory and use the script runh3.jc.

(change to the polyrate9.4.1/testrun directory)

```
./runh3.jc & <Return>
```

(or any of the other run{name}.jc scripts, where {name} can be ch5, oh3, ho2, clhbr, cmc, cwmc, nh3, hni, h2ni, ch4cl, or ch4oh)

3. To run an individual test run, say ch5j1tr1, change directory to polyrate9.4.1/testrun/ch5 and use the script ch5j1tr1.jc.

```
./ch5j1tr1.jc & <Return>
```

(or any of the {test_run_name}.jc scripts in that directory)

Note: Do not run more than one test run in the same directory simultaneously. (See Subsection 13.B, item VI.)

(3) To check the test run results:

When you finish running the test runs, you should check if they produced the correct results by comparing the test run short output files (fu15 files) with the ones in the

polyrate9.4.1/testo directory. There is a useful script, `check_test.jc`, in the `polyrate9.4.1/testrun` directory to do this after you finish a whole set of test runs. For example, if you have finished all the `oh3` test runs and you want to compare the results with the correct ones, change the working directory to `polyrate9.4.1/testrun` and type `./check_test.jc oh3 <Return>`. The script will use the Unix command `diff` to compare your results with the ones in the directory `polyrate9.4.1/testo`, and generate an output file called `oh3.chk`. You can then look at the contents of that file and see if there are any differences. Since the reaction rates are not calculated in test runs `ch5j2tr2` and `nh3tr1`, the short output files for these two test runs are not created. Thus if you type `check_test.jc ch5 <Return>`, there will be no check for `ch5j2tr2`. You can still use the `diff` command to compare the long output files (`fu6` files) of these test runs with the ones provided in the `polyrate9.4.1/testo` directory. There is a similar script, `check_all.jc`, that you can use to test the full suite of test runs.

If your test runs reproduce the results in `polyrate9.4.1/testo`, the only differences will be the date and time. Otherwise, the script will point out where the results differ from each other. (Of course, you have to know a little bit about the `diff` command to understand its output. Type `man diff <Return>` if you need help.) In running the test runs on different machines, we found that the differences in the rate constants were never greater than 1%. If your rate constant results do not differ from the ones in the `polyrate9.4.1/testo` directory by more than this percentage, they are probably correct; otherwise, something unexpected might be happening.

(4) Cleaning up:

When you finish checking the `testrun` results, you may want to clean up the `testrun` directory to save disk space. The script, `clean_up.jc`, in the `polyrate9.4.1/testrun` directory can be used to remove all the files POLYRATE generated when running the test suite. These files include the following: `*.fu1`, `*.fu6`, `*.fu14`, `*.fu15`, `*.fu25`, `*.fu26`, `*.fu27`, and `*.fu61`.

13.A.3. Using POLYRATE

After installation you should have the default directory structure and the correct machine-dependent files. If you are planning to use POLYRATE with a direct dynamics code, such as GAUSSRATE or MULTILEVELRATE, read the instructions in the direct

dynamics manual. If you are planning to use POLYRATE as a stand-alone code, follow the instructions in this section:

1. *Unless you are performing an electronic structure input file calculation, prepare your own potential energy surface code to do the calculations.* See Section 8 for a discussion of user-supplied potential energy surfaces. Supposing that you have made your own potential energy surface subroutine, you need to compile it before you link it with the POLYRATE source code. This can be accomplished by adding the surface to the Makefile in the src directory. Comments in the Makefile will show you where to add lines. For example, suppose you have made a surface subroutine for a system that consists of 2 hydrogen atoms and 1 oxygen atom, and you have put it in the polyrate9.4.1/poten directory and named it oh2.f. You would add the following line where the other executables are located.

```
OH2S = ../exe/polyrate$(VERSION).oh2.serial.exe
```

Then add ../poten/oh2.f to the POT: list, using a \ in order to continue a line. Also, add a line to compile the surface, beginning with a tab.

```
$(F77C) -c ../poten/oh2.f
```

Finally, add the lines to create the executable. Use a tab to start the second line.

```
OH2S:$(OBJ) dummy_mpi.o oh2.o
```

```
$(F77C) -o $(OH2) $(OBJ) dummy_mpi.o oh2.o
```

For three- and four-atom systems, the user can take advantage of some pre-written subroutines, namely: setup, surf, coord, and chain. If these subroutines were not explicitly included in the oh2.f file, the lines to create the executable would look like this (without a break between oh2.o and setup3.o):

```
OH2S:$(OBJ) dummy_mpi.o oh2.o
      $(F77C) -o $(OH2) $(OBJ) dummy_mpi.o oh2.o
      setup3.o surf3.o coord3.o chain3.o
```

2. *Prepare the input data files.* Study Section 11 carefully to make the input data files correctly. Because some of the input data files are quite complicated, it may be easier to use some of the test run input data files as templates when preparing your own input data files.
3. *Run the dynamics calculations with POLYRATE.* After you finish steps 1 and 2, you are ready to use POLYRATE to do your calculations. Before you submit your job to the computer, it is recommended that you do the following:
 - A. Make a directory for each specific reaction. For example, you can make a directory called reaction in the polyrate9.4.1 directory, and when you are planning to do a calculation on a specific reaction, say $O + H_2$, you can create a directory called oh2 in the polyrate9.4.1/reaction directory. You can then use this directory as your working directory for this reaction and put all the input data files and output files in this directory.
 - B. Use a job control script (.jc) to handle all the processes involved in running POLYRATE. Because POLYRATE only recognizes a certain set of input file names and will generate a set of output files with certain names (see Sections 11, 13.B and 14.C), the user always needs to copy (or link) the input data files to the default names and rename the output files to more logical names. It is more convenient to use a script to control these tedious processes. Also, using a script can make it easier to run the job in background or to submit the POLYRATE run as a batch job. In fact, we have included job control files for all the test runs in the distribution package. You probably used them when you ran the test runs, and you can easily modify them for your own needs. They all reside in the subdirectories of the polyrate9.4.1/testrun directory. In your .jc script, be sure to change the 'set name' and 'gmake' lines to match your job. In the OH_2 example, if you decided the name your files oh2run1, the lines would look like this.

```
set name = oh2run1
```

```
gmake OH2S
```

- C. With gmake in the job control script, the executable will be compiled the first time it is used, and also anytime there is a change in one of the files that it is dependant on. If a change is made to the source code, the next time that the job control script

is run, gmake will recognize that a change has been made and compile the changed code and create a new executable. If the user does not desire this feature, the gmake line may be deleted from the job control script once the executable is compiled for the first time. Executables are put in the exe directory by default.

13.B. Comments for expert users

Of course, users need not use the provided Unix scripts, files, or the recommended directory structure. All that is needed is the compilation of the source code, linking the POLYRATE object files with appropriate potential energy surface object file(s), and running the executable file(s) with input data file(s). However, there are a few potential problems which should be pointed out. They are:

I Choice of a proper dattim.machine

If the machine on which POLYRATE is being installed is one of those on which the program has been tested, then the appropriate dattim.machine can be used; otherwise the user is advised to read Subsection 5.A.1.

II Editing the param.inc file

Previous versions of featured param1.inc, param2.inc, param3.inc, and param4.inc. Version 9.4.1 still distributes these files, but streamlines the process by using param.inc by default, which is identical to param4.inc. The user may wish to overwrite the param.inc file with one of the examples, or edit the values in the param.inc file, such as increasing the maximum number of atoms from the default value of 11.

III Compilation of the POLYRATE source code

Compile energetics.f, givtst.f, hooks.f, intbsv1.f, intbsv2.f, intbsv3.f, interface.f, main.f, polyrate.f, polyag.f, polyhl.f, polymq.f, polyrr.f, polysz.f, rtpjac.f, and the appropriate dattim.machine with the FORTRAN compiler. For execution on an IBM RS/6000, compile dattim.ibm with the FORTRAN compiler and dateclock.ibm with the C compiler.

The POLYRATE program should be compiled with 64-bit precision and static memory allocation for a Unix environment. Note: On the Cray Y-MP, use the -ev compiler option instead of the -a static option.

Note: The path used for searching for the INCLUDE files must be specified in the compile command on some machines, or the three INCLUDE files, param.inc, percon.inc, and common.inc, must be in the working directory specified by the compile command.

IV Compilation of the potential energy surface files

The potential energy surface subroutines provided with the POLYRATE program should be compiled with the same precision and memory specifications used for the POLYRATE source code. Any user-supplied potential energy surface subroutine must also be compiled with 64-bit precision, but specification for static memory allocation will depend on the way in which the subroutine is written.

V Creation of an executable file

Link the POLYRATE object file with potential object file(s) to create specific POLYRATE-potential executable file(s).

VI Running POLYRATE

The POLYRATE program opens all the input and output files needed for a calculation with the logical file names poly.fu#, where # is an integer. The files poly.fu5 and poly.fu6 are always opened. The other poly.fu# files are opened depending on which options are set. Therefore, all the necessary input data files must be copied or linked to files with the proper logical names. If the user wants the output files to have different names, the renaming of the poly.fu# output files must be performed after the calculation has completed.

Note: Because the POLYRATE program opens and closes its files with the file name specification poly.fu#, these files should be renamed at the end of each calculation, or there will be conflicts when two calculations are run in the same directory.

In addition to the POLYRATE input and output files, the user should ensure that the potential data files, if any, which are opened by the potential energy surface subroutine(s), have the correct file names as specified in the OPEN statement(s) in the potential subroutine source code. For the test runs, these names are listed after "FILE NAME =" in Subsection 14.C. Also, these data files need to be in the working directory of the calculation.

13.C. Running in parallel using MPI

The POLYRATE program has been expanded to work with multiple processors using MPI. MPI can be very machine dependent, therefore this section will highlight the areas that need to be correct for the parallel feature to work correctly. For samples on how to make MPI work, see the CH5P sections of the Makefile, and look at the job control scripts in the ch5 testruns. The MPI sections of the Makefile are only printed if an MPI compiler is found by the configure script.

- I. In the Makefile, MPIF77C needs to be set to the correct MPI compiler. The `poly_mpi.f` source file must be compiled using this compiler, not the serial compiler. Finally, the linking section must use the MPI compiler when creating the executable.

- II. In the job control script (*.jc file), set `$nproc` equal to the number of processors you wish to use. Make sure the `exe` variable is set to the proper executable. Most importantly, the platform on which you are running has a certain way that MPI jobs must be run. By default, the line looks like this:

```
(time mpirun -v -np $nproc $exe) >>& $name.time
```

where “`mpirun -v -np $nproc $exe`” is the part that actually tells the program how to run. This line will likely need to be edited to work with each platform.

- III. For an example, the testrun `ch5j1tr1` has edited this line in the job control script to work on a particular IBM regatta. The line now reads:

```
(time poe $exe -rmpool 1 -nodes 1 -procs $nproc) >>& $name.time
```

14. SAMPLE TEST RUNS

14.A. Chart of the sample param.inc files

The following chart illustrates which of the sample param.inc files are applicable to each POLYRATE test run. (An X denotes that the INCLUDE file at the top of that column is suitable for the test run on that line.)

TEST RUN	param4.inc	param3.inc	param2.inc	param1.inc
ch4cltr1– ch4cltr2	X	X	X	X
ch4ohtr1– ch4ohtr2	X	X		
ch4otr1	X			
ch5fu29tr1– ch5fu29tr2	X	X	X	
ch5fu30tr1– ch5fu30tr6	X	X	X	
ch5fu31tr1	X	X	X	
ch5fu40tr1– ch5fu40tr6	X	X	X	
ch5fu50tr1– ch5fu50tr3	X	X	X	
ch5fu51tr1	X	X	X	
ch5icfu30tr1– ch5icfu30tr2	X	X	X	
ch5icfu31tr1	X	X	X	
ch5icfu40tr1– ch5icfu40tr2	X	X	X	
ch5j1tr1– ch5j1tr2	X		X	
ch5j1tr3– ch5j1tr4	X	X	X	X
ch5j1tr5– ch5j1tr6	X		X	

TEST RUN	param4.inc	param3.inc	param2.inc	param1.inc
ch5j1tr7– ch5j1tr8	X	X	X	X
ch5j2tr9– ch5j2tr11	X		X	
ch5j2tr1	X	X	X	
ch5j2tr2	X	X	X	X
ch5j2tr3	X	X	X	
ch5j2itr1– ch5j2itr2	X	X	X	
clhbrtr1– clhbrtr4	X	X	X	
cmctr1–cmctr3	X	X	X	X
cwmctr1	X	X		
cwmctr2,3	X			
h2nitr1	X		X	
h3tr1	X	X	X	X
h3tr2	X		X	
h3tr3–h3tr4	X	X	X	X
hnitr1	X	X	X	X
ho2tr1–ho2tr5	X	X	X	X
nh3tr1–nh3tr2	X	X	X	X
oh3tr1–oh3tr10	X	X	X	X
oh3fu30tr1– oh3fu30tr6	X	X	X	X
oh3fu31tr1– oh3fu31tr2	X	X	X	X
oh3fu40tr1– oh3fu40tr6	X	X	X	X
ohcltr1–ohcltr4	X		X	

14.B. Chart of the potential energy functions and data files

The following chart gives the name of the potential energy function and data file(s) that are used for each test run. The are files located in the polyrate9.4.1/poten directory. When two data files are listed for the same test run, both data files are required. The file listed first should be assigned to unit 4, and the file listed second should be assigned to unit 7.

TEST RUN(S)	POTENTIAL SOURCE CODE	POTENTIAL DATA FILE(S)
ch5j1tr1-ch5j1tr11	ch5.f	potch5j1.dat
ch5j2tr1-ch5j2tr3	ch5.f	potch5j2.dat
ch5fu51.tr1	ch5.f	potch5j1.dat
clhbrtr1-clhbrtr4	clhbr.f	(none)
cmctr1, cmctr2	cmc.f	potcmc1.dat potcmc2.dat
cwmctr1-cwmctr3	cwmc.f	potcwmc3b.dat potcwmcvib.dat

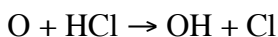
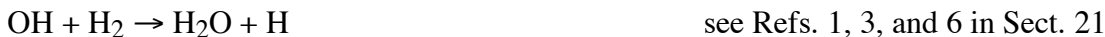
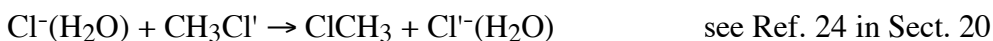
TEST RUN(S)	POTENTIAL SOURCE CODE	POTENTIAL DATA FILE(S)
ch4otr1	ch4o.f	potch4o.dat potch4o.inc
h3tr1-h3tr4	h3.f	(none)
nh3tr1-nh3tr2	nh3.f	(none)
ho2tr1-ho2tr5	ho2.f	(none)
oh3tr1-oh3tr10	oh3.f	(none)
hnitr1	hni.f	pothni.dat
h2nitr1	h2ni.f	poth2niads.dat poth2nilat.dat
ohcltr1-ohcltr4	ohcl.f	(none)

14.C. Descriptions of sample test runs

This section contains descriptions of the sample runs which make up the POLYRATE test suite. The input data files and the summary output files from unit fu15 for the test runs

are part of the distribution package. For convenience, we have named all the test runs according to the potential energy surface used for the test run.

The following is a list of reactions involved in the test suite and references to published calculations of their reaction rates using POLYRATE.



Various options were used for each of these systems to give a thorough test of POLYRATE. Some of the most critical options used are discussed in each description; the

keywords used for these options are specified in square brackets: e. g. [KEYWORD]. The potential energy surfaces used for the sample runs for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ are surfaces J1 and J2 of Joseph et al. [Ref. 17, Sect. 21]. Surface J1 is also used for the calculations for the reaction $\text{CH}_3 + \text{HPh} \rightarrow \text{CH}_4 + \text{Ph}$, where Ph denotes a single atom with the mass of a phenyl group (see Ref. 21 in Sect. 21). The calculations for the reaction $\text{Cl} + \text{HBr} \rightarrow \text{ClH} + \text{Br}$ use the extended LEPS surface of Babamov, Lopez, and Marcus [Ref. 26, Sect. 20]. The calculations for the $\text{Cl}^-(\text{H}_2\text{O}_m) + \text{CH}_3\text{Cl}^+ \rightarrow \text{ClCH}_3 + \text{Cl}^-(\text{H}_2\text{O}_m)$ reactions, $m = 0,1$, use the potential energy surface of Tucker and Truhlar [Ref. 23, Sect. 20] and Zhao et al. [Refs. 24 and 27, Sect. 20], while the DMBE surface of Varandas et al. [Ref. 28, Sect. 20] is used for the reaction $\text{H}_2 + \text{H}^+ \rightarrow \text{HH}^+ + \text{H}$. The calculations for the reaction $\text{OH} + \text{O} \rightarrow \text{HO}_2$ and $\text{O} + \text{H}_2 \rightarrow \text{HO}_2$ use the ab initio surface of Melius and Blint [Ref. 29, Sect. 20], and all the runs for the reaction $\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$ use the Schatz-Elgersma fit to the Walch-Dunning ab initio potential energy surface [Refs. 30 and 31, Sect. 20]. The embedded diatomics-in-molecules (EDIM) potential with parameter set 4 of Truong et al. [Refs. 25 and 32, Sect. 20] is used for the H/Ni(100) surface diffusion test run, and the Lee and DePristo [Ref. 33, Sect. 20] potential energy surface, as modified by Truong et al. [Ref. 19, Sect. 21], is used for the $\text{H}_2/\text{Ni}(100)$ system. Furthermore, both Ni surface test runs are carried out assuming that the entire metallic subsystem on which the diatoms are absorbed is rigid. The calculations for the reaction $\text{Cl}^- + \text{CH}_3\text{Cl}^+ \rightarrow \text{ClCH}_3 + \text{Cl}^-$ use the semiempirical surface of Tucker and Truhlar [Ref. 23, Sect. 20], and the calculations for the NH_3 inversion use potential energy surface II of Brown, Tucker, and Truhlar [Ref. 7, Sect. 21]. For the $\text{OH} + \text{CH}_4 \rightarrow \text{H}_2\text{O} + \text{CH}_3$ [Ref. 32, Sect. 21] and $\text{Cl} + \text{CH}_4 \rightarrow \text{HCl} + \text{CH}_3$ [Ref. 22, Sect. 21] reactions, rate calculations are carried out using the IVTST method [Ref. 26, Sect. 21] with *ab initio* data.

It should be pointed out that the results for the ClHBr and the H_3 test runs will differ from the published results. This difference occurs because the published results for those systems were determined using another program, ABCRATE, which calculates anharmonicity corrections differently than POLYRATE. In particular, ABCRATE uses curvilinear internal coordinates, and these produce quite different harmonic frequencies, as discussed elsewhere [Ref. 25 and 41, Sect. 21].

Note that the test runs are designed to illustrate the use of the POLYRATE program. No implication is made that these runs are converged with respect to all step sizes and grid ranges. One can use these runs to test that the code as received gives the same results on

various new platforms by comparing the output for these test runs to the distributed output, even though the results are not necessarily converged with respect to numerical parameters.

14.C.1. Test run ch5j1tr1

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with harmonic approximation

This sample run is for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with the J1 potential energy surface. All normal modes and generalized normal modes are treated using the default harmonic approximation. The rate constants are computed using CVT and ICVT [CVT, ICVT]. Tunneling corrections are computed using the MEPSAG, CD-SCSAG, and LCG3 methods [ZCT, SCT, SCTOPT, LCTOPT] Note that only the ground-state in the LCG3 exit arrangement is considered for the LCG3 calculation in this run [NOALLEXCIT] this option is chosen to simplify the calculation because we expect on physical grounds that only ground-state-to-ground-state tunneling is important in this case. The LCT keyword was not explicitly given since the default is on when LCTOPT is used. The rate constants are computed for ten temperatures [TEMP], and the activation energies are computed for six pairs of temperatures [EACT]. The MEP is followed using the Page-McIver method [RPM = *page*] with a step size of 0.001 bohr [SSTEP]. The Hessian matrix is recalculated and a mode analysis is performed every 0.01 bohr [INH = 100]. The curvature components are calculated using a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate [CURV = *dgrad*]. A restart file containing reaction-path information is written to unit fu1 [RESTART = *writeln*] for use in test run ch5j1tr2. For both reactants and the first product STATUS = 6 and the geometries, energies, and frequencies are read from fu5, for the second product STATUS = 2 and the geometry is read from fu5, and for the saddle point STATUS = 4 and the geometry, energy, and Hessian are read from fu5. The reactants are input in fu5 [ENERGY, FREQ] to reduce the number of calculations. The results may be compared to those in Table VII of Ref. 17 in Sect. 21.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j1tr1.dat	poly.fu5	input data file
ch5j1tr1.fu6	poly.fu6	long output file
ch5j1tr1.fu1	poly.fu1	reaction path info.
ch5j1tr1.fu14	poly.fu14	summary output file
ch5j1tr1.fu15	poly.fu15	summary output file
ch5j1tr1.fu61	esp.fu61	2nd restart option file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat UNIT = 4
 FILE NAME = potch5.dat

14.C.2. Test run ch5j1tr2

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with harmonic approximation

This sample run is for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with the J1 potential energy surface. All the parameters are the same as for test run ch5j1tr1 except that the reaction-path information is read from a restart file on unit fu1 [RESTART: readfu1]. In order to perform this test run, the restart file must first be created by the user by running test run ch5j1tr1.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j1tr2.dat	poly.fu5	input data file
ch5j1tr1.fu1	poly.fu1	reaction path info. (input data for a restart calculation)
ch5j1tr2.fu6	poly.fu6	long output file
ch5j1tr2.fu14	poly.fu14	summary output file
ch5j1tr2.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat UNIT = 4
 FILE NAME = potch5.dat

14.C.3. Test run ch5j1tr3

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with mass-weighted reaction coordinate

This sample run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ is similar to test run ch5j1tr1, except that the reduced mass [SCALEMASS] which defines the mass-scaled Cartesian coordinates, is set equal to the default, 1.0 amu. Thus, these coordinates have the same numerical values as the conventional mass-weighted coordinates in infrared spectroscopy. The vibrational partition functions are multiplied by a factor of 1×10^{10} to avoid underflow [VPFEXP]; note that these changes alone would not change the final rate constants. In addition, though, in this run the effective mass for the CD-SCSAG tunneling calculation is obtained by sixth-order Lagrangian interpolation [SCTOPT: lagrange] instead of a cubic spline fit. Also, there is no LCG3 calculation in this test run.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j1tr3.dat	poly.fu5	input data file
ch5j1tr3.fu6	poly.fu6	long output file
ch5j1tr3.fu14	poly.fu14	summary output file
ch5j1tr3.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat	UNIT = 4
	FILE NAME = potch5.dat

14.C.4. Test run ch5j1tr4

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with harmonic approximation in redundant internal coordinates and the Page-McIver method to calculate reaction-path curvature

This sample run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ demonstrates the calculation of reaction-path curvature components from the Hessian, generalized normal mode eigenvectors, and gradient [$\text{CURV} = \text{dhess}$] using eq. (48) of Page and McIver [Ref. 17, Sect. 20]. The MEP is followed using the Euler steepest-descents (ESD) method [$\text{RPM} = \text{esd}$] with a step size of 0.0001 bohr [SSTEP]. A generalized normal mode analysis is performed every 0.01 bohr along the MEP [$\text{INH} = 100$], and all the vibrations are treated harmonically in redundant internal coordinates [$\text{COORD} = \text{curv3}$]. Both CVT [CVT] and ICVT [ICVT] rate constants are calculated, and tunneling corrections are computed using both MEPSAG and CD-SCSAG methods [$\text{ZCT}, \text{SCTOPT}$].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j1tr4.dat	poly.fu5	input data file
ch5j1tr4.fu6	poly.fu6	long output file
ch5j1tr4.fu14	poly.fu14	summary output file
ch5j1tr4.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat	UNIT = 4
	FILE NAME = potch5.dat

14.C.5. Test run ch5j1tr5

CH₃ + HPh → CH₄ + Ph with LCG3 calculation for the ground state only

This sample run is for the CH₃ + HPh → CH₄ + Ph reaction, where Ph denotes a single particle with the mass of a phenyl group. The CH₅ analytical potential energy surface, J1, is used for this six-atom model problem; see Ref. 21 in Sect. 21 for further discussion. All normal modes and generalized normal modes are treated using the default harmonic approximation. TST, CVT, and ICVT rate constants are computed [TST, CVT, ICVT]. The tunneling calculations for ground-state transmission coefficients are computed using the MEPSAG, CD-SCSAG, and LCG3 methods [ZCT, SCTOPT, LCTOPT]. The program writes a restart file containing the reaction-path information to unit fu1 for use in test run ch5j1tr6 [RESTART = *writeln*]. The MEP is followed using the ESD integrator without stabilization (ESD method) [RPM = *esd*] with a step size of 0.0001 bohr [SSTEP], and a generalized normal mode analysis is performed every 0.01 bohr [INH = 100]. The curvature components are calculated using a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate [CURV = *dgrad*]. The results may be compared to those in Table 5 in Ref. 21 in Sect. 21, but it should be noted that there is a difference of a few percent between the LCG3 results in Ref. 21 and those calculated with this data file due to improved convergence.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j1tr5.dat	poly.fu5	input data file
ch5j1tr5.fu6	poly.fu6	long output file
ch5j1tr5.fu1	poly.fu1	reaction path info.
ch5j1tr5.fu14	poly.fu14	summary output file
ch5j1tr5.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat	UNIT = 4
	FILE NAME = potch5.dat

14.C.6. Test run ch5j1tr6

$\text{CH}_3 + \text{HPh} \rightarrow \text{CH}_4 + \text{Ph}$ with LCG3 calculation for all energetically accessible excited states from a restart file in which the LCG3 calculation was carried out for the ground state only; rate constants are also calculated at more temperatures. (Restart file for unit fu1 is from test run ch5j1tr5)

This sample run, like the preceding test run, is for the $\text{CH}_3 + \text{HPh} \rightarrow \text{CH}_4 + \text{Ph}$ reaction, where Ph denotes a single particle with the mass of a phenyl group. The potential energy surface and all the parameters are the same as for test run ch5j1tr5 except for the tunneling calculations and the temperatures for which the rate constants are calculated. In this run the tunneling calculations for ground-state transmission coefficients are performed with the MEPSAG, CD-SCSAG, and LCG3 methods [ZCT, SCTOPT, LCTOPT]. For the LCG3 calculation, the program determines the highest excited state of the LCG3 exit arrangement which is accessible, and those excited states are included in the LCG3 calculation [ALLEXCIT]. The results may be compared to those from test run ch5j1tr5, in which the LCG3 calculations were done for the ground state only; the present LCG3 transmission coefficients are up to 4% larger.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j1tr6.dat	poly.fu5	input data file
ch5j1tr5.fu1	poly.fu1	reaction path info. (input data for a restart calculation)
ch5j1tr6.fu6	poly.fu6	long output file
ch5j1tr6.fu14	poly.fu14	summary output file
ch5j1tr6.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat UNIT = 4
FILE NAME = potch5.dat

14.C.7. Test run ch5j1tr7

CH₃ + H₂ → CH₄ + H with Page-McIver mass-weighted reaction path and IVTST0FREQ

This sample run for the reaction CH₃ + H₂ → CH₄ + H is similar to test run ch5j1tr3, except that the reaction-path curvature components are calculated from the Hessian, generalized normal mode eigenvectors, and gradient [CURV = *dhess*] using eq. (48) of Page and McIver [Ref. 17, Sect. 20]. The MEP is followed using the Page-McIver method [RPM = *pagem*] with a step size of 0.001 bohr [SSTEP]. The Hessian matrix is recalculated and a mode analysis is performed every 0.01 bohr [INH = 100]. The vibrational partition functions are multiplied by a factor of 1 × 10¹⁰ to avoid underflow [VPFEXP]; note that this alone would not change the final rate constants. In addition, the effective mass for the CD-SCSAG tunneling calculation is obtained by sixth-order Lagrangian interpolation [SCTOPT: *lagrange*] instead of the default cubic spline fit. Also, there is no LCG3 calculation in this test run. The IVST0FREQ method [IVTST0FREQ] is used to modify the 10th and 11th vibrational modes along the reaction path.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j1tr7.dat	poly.fu5	input data file
ch5j1tr7.fu6	poly.fu6	long output file
ch5j1tr7.fu14	poly.fu14	summary output file
ch5j1tr7.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat UNIT = 4
 FILE NAME = potch5.dat

14.C.8. Test run ch5j1tr8

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with MEPSAG and CD-SCSAG calculations at two temperatures [one of the test cases in Ref. 27 in Sect. 21]

This sample run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ is one of the test cases in Ref. 27 in Sect. 21. (However, Ref. 27 used SCSAG whereas the current code uses CD-SCSAG). The MEP is calculated from $s = -1.3$ bohr to $s = 0.8$ bohr [SRANGE] using the Euler stabilization method [RPM = *es1*]. The gradient and save step sizes are all set to 0.047 bohr [SSTEP, INH = 1]. The finite difference step size and the tolerance parameter in the ES1 method are set to 0.072 ($1.532 \cdot \text{SSTEP}$) [ES1OPT: delta2] and 0.082 [ES1OPT: DIFFD], respectively. The CVT/MEPSAG and CVT/CD-SCSAG rate constants are calculated [CVT, ZCT, SCT] at 200K and 300K [TEMP]. All the rate constants are within 15% of the converged values obtained in the test run ch5j1tr1.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j1tr8.dat	poly.fu5	input data file
ch5j1tr8.fu6	poly.fu6	long output file
ch5j1tr8.fu14	poly.fu14	summary output file
ch5j1tr8.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat	UNIT = 4
	FILE NAME = potch5.dat

14.C.9. Test run ch5j1tr9

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with harmonic approximation

This sample run is for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with the J1 potential energy surface. All normal modes and generalized normal modes are treated using the default harmonic approximation. The rate constants are computed using CVT and ICVT [CVT, ICVT]. Tunneling corrections are computed using the MEPSAG, CD-SCSAG, and LCG4 methods [ZCT, SCT, SCTOPT, LCTOPT] Note that only the ground-state in the LCG4 exit arrangement is considered for the LCG4 calculation in this run [NOALLEXCIT] this option is chosen to simplify the calculation because we expect on physical grounds that only ground-state-to-ground-state tunneling is important in this case. The LCT keyword was not explicitly given since the default is on when LCTOPT is used. The rate constants are computed for ten temperatures [TEMP], and the activation energies are computed for six pairs of temperatures [EACT]. The MEP is followed using the Page-McIver method [RPM = *pagem*] with a step size of 0.001 bohr [SSTEP]. The Hessian matrix is recalculated and a mode analysis is performed every 0.01 bohr [INH = 100]. The curvature components are calculated using a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate [CURV = *dgrad*]. For both reactants and the first product STATUS = 6 and the geometries, energies, and frequencies are read from fu5, for the second product STATUS = 2 and the geometry is read from fu5, and for the saddle point STATUS = 4 and the geometry, energy, and Hessian are read from fu5. The reactants are input in fu5 [ENERGY, FREQ] to reduce the number of calculations. The results may be compared to those in Table VII of Ref. 17 in Sect. 21.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j1tr9.dat	poly.fu5	input data file
ch5j1tr9.fu6	poly.fu6	long output file
ch5j1tr9.fu1	poly.fu1	reaction path info.
ch5j1tr9.fu14	poly.fu14	summary output file
ch5j1tr9.fu15	poly.fu15	summary output file
ch5j1tr9.fu61	esp.fu61	2nd restart option file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat UNIT = 4
 FILE NAME = potch5.dat

14.C.10. Test run ch5j1tr10

$\text{CH}_3 + \text{HPh} \rightarrow \text{CH}_4 + \text{Ph}$ with LCG4 calculation for all energetically accessible excited states from a restart file in which the LCG4 calculation was carried out for the ground state only; rate constants are also calculated at more temperatures. (Restart file for unit fu1 is from test run ch5j1tr5)

This sample run, like the preceding test run, is for the $\text{CH}_3 + \text{HPh} \rightarrow \text{CH}_4 + \text{Ph}$ reaction, where Ph denotes a single particle with the mass of a phenyl group. The potential energy surface and all the parameters are the same as for test run ch5j1tr5 except for the tunneling calculations and the temperatures for which the rate constants are calculated. In this run the tunneling calculations for ground-state transmission coefficients are performed with the MEPSAG, CD-SCSAG, and LCG4 methods [ZCT, SCTOPT, LCTOPT]. For the LCG4 calculation, the program determines the highest excited state of the LCG4 exit arrangement which is accessible, and those excited states are included in the LCG4 calculation [ALLEXCIT].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j1tr10.dat	poly.fu5	input data file
ch5j1tr5.fu1	poly.fu1	reaction path info. (input data for a restart calculation)
ch5j1tr10.fu6	poly.fu6	long output file
ch5j1tr10.fu14	poly.fu14	summary output file
ch5j1tr10.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat UNIT = 4
FILE NAME = potch5.dat

14.C.11. Test run ch5j1tr11

$\text{CH}_3 + \text{HPh} \rightarrow \text{CH}_4 + \text{Ph}$ with interpolated LCG4 calculation with all energetically accessible excited states.

This sample run for the $\text{CH}_3 + \text{HPh} \rightarrow \text{CH}_4 + \text{Ph}$ is similar than the previous run where Ph denotes a single particle with the mass of a phenyl group. The only difference with the previous test is that the large curvature tunneling energies obtained by the LCG4 method are interpolated to 9 points in the nonadiabatic zone [ILCT1D with LCTGRD equal to 9 for all excited states]. The ILCG4 calculation includes all the final excited states that are accessible [ALLEXCIT]. It is interesting to compare the LCG4 transmission factors with the ones calculated in the previous run in which NGTHETA = 180.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j1tr11.dat	poly.fu5	input data file
ch5j1tr11.fu6	poly.fu6	long output file
ch5j1tr11.fu14	poly.fu14	summary output file
ch5j1tr11.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat UNIT = 4
 FILE NAME = potch5.dat

14.C.12. Test run ch5j2tr1

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with harmonic approximation and ES1 method

This sample run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with the J2 potential energy surface is similar to the one described in Subsection 14.C.1. In addition to employing a different potential energy surface, a major difference between this test run and test run ch5j1tr1 is the method of calculating the MEP. Also, the LCG3 calculation is omitted in this run. To calculate the MEP, a step size of 0.001 bohr is used for the first five steps from the saddle point [SFIRST] and then a larger step size (0.01 bohr) [SSTEP] is used for the remaining steps in combination with the Euler stabilization (ES1) method [RPM = *es1*]. In addition, sixth-order Lagrangian interpolation is used to obtain the values of the effective mass for the CD-SCSAG tunneling calculation [SCTOPT : *lagrange*].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j2tr1.dat	poly.fu5	input data file
ch5j2tr1.fu6	poly.fu6	long output file
ch5j2tr1.fu14	poly.fu14	summary output file
ch5j2tr1.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j2.dat	UNIT = 4
	FILE NAME = potch5.dat

14.C.13. Test run ch5j1tr12

CH3 + HPh CH4 + Ph with 2D interpolated LCG4 with all energetically accessible excited states.

In this run the CH3 + HPh CH4 + Ph thermal rate constants are calculated with the LCT4 method with all energetically accessible vibrational excited states. As in the previous run Ph denotes a single particle with the mass of the phenyl group. The only difference with previous test is that both the tunneling energies and the points along each tunneling energies are obtained from a grid specified by LCTGRD and fitted to a 2D spline under tension [option ILCT2D in LCTOPT]. The rectangular grid was built from 7 tunneling energies and 11 points along each tunneling energy. It is interesting to compare this run with the previous one, which uses a 1D interpolation and with test ch5j1tr10, which corresponds to the full LCT4 calculation.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME
ch5j1tr12.dat	poly.fu5 input data file
ch5j1tr12.fu6	poly.fu6 long output file
ch5j1tr12.fu14	poly.fu14 summary output file
ch5j1tr12.fu15	poly.fu15 summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1	UNIT = 4
	FILE NAME = potch5.dat

14.C.14. Test run ch5j2tr2

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with MEP only

This sample run is for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$. The RATE and TUNNEL sections are not given so the rate constants are not calculated. The MEP is calculated using the ESD method [RPM = *esd*]. The MEP information is printed [PRINTSTEP] at grid points separated by 0.01 bohr [SSTEP]. The overall translations and rotations are projected out at stationary points [PROJECT].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j2tr2.dat	poly.fu5	input data file
ch5j2tr2.fu6	poly.fu6	long output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j2.dat	UNIT = 4
	FILE NAME = potch5.dat

14.C.15. Test run ch5j2tr3

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with WKB approximation for two modes

This sample run is for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ using a combination of methods for treating the various normal modes [VANHAR]. All reactants and products are treated harmonically. Along the reaction path the two lowest frequency modes are treated using the uniform WKB method with a quadratic-quartic fit to the potential at two input distances along the generalized normal modes [QQSEMI] and the rest of the modes are treated using the harmonic approximation. The rate constants are calculated at five temperatures [TEMP], and four temperature pairs are used to compute activation energies [EACT]. Both CVT [CVT] and ICVT [ICVT] rate constants are calculated, using both MEPSAG and CD-SCSAG tunneling corrections [ZCT, SCT]. The ESD integrator [RPM = *esd*] is used to calculate the MEP, using a step size of 0.001 bohr [SSTEP]. The limits for the MEP are determined by using two keywords: SRANGE and SPECSTOP. A generalized normal mode analysis is performed every 0.01 bohr [INH = 10].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j2tr3.dat	poly.fu5	input data file
ch5j2tr3.fu6	poly.fu6	long output file
ch5j2tr3.fu14	poly.fu14	summary output file
ch5j2tr3.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j2.dat	UNIT = 4
	FILE NAME = potch5.dat

14.C.16. Test run ch5j2itr1

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with zero-order IVTST approximation

This zero-order IVTST test run [IVTST0] is for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ using the energies, geometries, and vibrational frequencies of the reactants, saddle point, and products from the J2 potential of Joseph et al. [Ref. 17, Sect. 21].* All reactants and products are treated harmonically. The rate constants are calculated at four temperatures [TEMP]. This test run reproduces the results from one of the test cases in Ref. 26 in Sect. 21.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j2itr1.dat	poly.fu5	input data file
ch5j2itr1.fu29	poly.fu29	input data file
ch5j2itr1.fu6	poly.fu6	long output file

* The data are input by means of the unit fu29 file.

14.C.17. Test run ch5j2itr2

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ with first-order IVTST approximation

This first-order IVTST test run [IVTST1] is for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ using the energies, geometries, and vibrational frequencies of the reactants, saddle point, products, and a point on the MEP at $s = 0.01$ bohr from the J2 potential of Joseph et al. [Ref. 17, Sect. 21].* All reactants and products are treated harmonically. The rate constants are calculated at four temperatures [TEMP]. This test run reproduces the results from one of the test cases in Ref. 26 in Sect. 21.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5j2itr2.dat	poly.fu5	input data file
ch5j2itr2.fu29	poly.fu29	input data file
ch5j2itr2.fu6	poly.fu6	long output file

* The data are input by means of the unit fu29 file.

14.C.18. Test run ch5fu29tr1

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ zero-order IVTST reaction rate with electronic structure data read from file fu29 and frequencies input directly.

This is a zero-order IVTST test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ [IVTST0]. Electronic structure information calculated at the UHF/STO-3G level (energies and frequencies) is read from unit fu29. At this level of electronic structure theory, CH_3 is nonplanar; therefore the symmetry number of the forward reaction is 2 [SIGMAF=2]. All reactants and products are treated harmonically and the frequencies are in cm^{-1} . The TST rate constants [TST, NOCVT] are calculated at four temperatures [TEMP] with zero-curvature transmission coefficients calculation. [ZCT] The zero-order IVTST calculation can also be done with the fu50 option. A similar test run (ch5fu50tr1) can be used to check the result.

POLYRATE I/O FILES

FILE NAME

ch5fu29tr1.dat
ch5fu29tr1.fu29
ch5fu29tr1.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu29 input data file
poly.fu6 long output file

14.C.19. Test run ch5fu29tr2

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ zero-order IVTST reaction rate with information read from fu29 and Hessian matrix input in packed form

This is a zero-order IVTST test run [TST, CVT] for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ [IVTST0]. Electronic structure information calculated at the UHF/STO-3G level (Hessians of reactants, saddle point, and products in packed format) is read from file fu29. At this level of electronic structure theory, CH_3 is nonplanar; therefore the symmetry number of the forward reaction is 2 [SIGMAF=2]. All reactants and products are treated harmonically. Hessians and geometries are input in unscaled Cartesian coordinates in bohrs. The rate constants are calculated at four temperatures [TEMP]. A similar calculation has been done in the ch5tr29tr1 and ch5fu50tr1 test runs.

POLYRATE I/O FILES

FILE NAME

ch5fu29tr2.dat
ch5fu29tr2.fu29
ch5fu29tr2.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu29 input data file
poly.fu6 long output file

14.C.20. Test run ch5fu30tr1

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ TST reaction rate with frequencies read from fu30 directly

This is a conventional TST test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ [NOCVT and TST]. Electronic structure information calculated at the UHF/STO-3G level (energies and frequencies of reactants, saddle point, and products) is read from file fu30 [POTENTIAL = *unit30*]. At this level of electronic structure theory, CH_3 is nonplanar; therefore the symmetry number of the forward reaction is 2 [SIGMAF=2]. Harmonic frequencies of reactants and products are input in atomic units [LOPT(4) = 0 in unit fu30]. The rate constants are calculated at four temperatures [TEMP].

POLYRATE I/O FILES

FILE NAME

ch5fu30rt1.dat
ch5fu30tr1.fu30
ch5fu30tr1.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu30 input data file
poly.fu6 long output file

14.C.21. Test run ch5fu30tr2

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ TST reaction rate with information read from fu30 and full Hessian matrices input

This is a conventional TST test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ [NOCVT and TST]. Electronic structure information calculated at the UHF/STO-3G level (energies and full matrix format Hessians of reactants, saddle point, and products) is read from file fu30 [POTENTIAL = *unit30*]. At this level of electronic structure theory, CH_3 is nonplanar; therefore the symmetry number of the forward reaction is 2 [SIGMAF=2]. All reactants and products are treated harmonically. Hessians and geometries are input in unscaled Cartesian coordinates in atomic units [LOPT(4) = 1 and LOPT(3) = 0, respectively, in unit fu30]. The rate constants are calculated at four temperatures [TEMP].

POLYRATE I/O FILES

FILE NAME

ch5fu30tr2.dat
ch5fu30tr2.fu30
ch5fi30tr2.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu30 input data file
poly.fu6 long output file

14.C.22. Test run ch5fu30tr3

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ TST reaction rate with information read from fu30 and Hessian matrix input in packed form

This is a conventional TST test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ [NOCVT and TST]. Electronic structure information calculated at the UHF/STO-3G level (energies and Hessians of reactants, saddle point, and products in packed format) is read from file fu30 [POTENTIAL = *unit30*]. At this level of electronic structure theory, CH_3 is nonplanar; therefore the symmetry number of the forward reaction is 2 [SIGMAF=2]. All reactants and products are treated harmonically (anharmonicity is not supported for conventional TST calculations in POLYRATE-9.4.1). Hessians are input in unscaled Cartesian coordinates in atomic units [LOPT(4) = -1 in unit fu30]. The rate constants are calculated at four temperatures [TEMP].

The electronic structure information for this run was generated with the GAMESS electronic structure package (Aug. 11, 1994 – IBM version), which yields an H_2 stretching frequency of 5479.11 cm^{-1} . POLYRATE, with this input, yields 5479.18 cm^{-1} . The GAUSSIAN94 electronic structure package yields a slightly different Hessian and a frequency of 5478.64 cm^{-1} .

Notice that the Hessian for H_2 corresponds to the molecule being on the z -axis, whereas the geometry corresponds to the molecule being on the y -axis. For calculating reactant frequencies this does not matter, but such an inconsistency would matter along the reaction path (see Section 9.B). It is worthwhile to point out that commonly-available electronic structure packages may, by default, re-orient a molecule before computing its Hessian.

POLYRATE I/O FILES

FILE NAME

ch5fu30tr3.dat
ch5fu30tr3.fu30
ch5fu30tr3.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu30 input data file
poly.fu6 long output file

14.C.23. Test run ch5fu30tr4

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ TST reaction rate with information read from fu30 and frequencies input directly

This is a conventional TST test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ [NOCVT and TST]. Electronic structure information (energies and frequencies of the reactants, the products, and the saddle point) at the QCISD/6-311G(d, p) level is read from file fu30 [POTENTIAL = *unit30*]. (The $V_{\text{MEP}}(s)$ has been scaled by a factor of 0.86 to be consistent with Ref. 48 in Section 20. The scaling was done by hand; however, the VSCALE keyword could have been used instead of hand scaling) At this level of electronic structure theory, CH_3 is planar; therefore the symmetry number of the forward reaction is 4 [SIGMAF=4]. Harmonic frequencies of reactants and products are input in atomic units and geometries are input in unscaled Cartesian coordinates in bohrs. The rate constants are calculated at twelve temperatures [TEMP].

POLYRATE I/O FILES

FILE NAME

ch5fu30tr4.dat
ch5fu30tr4.fu30
ch5fu30tr4.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu30 input data file
poly.fu6 long output file

14.C.24. Test run ch5fu30tr5

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ TST and CVT reaction rates with information along the reaction path read from fu30

This is a sample run that illustrates the use of electronic structure input file data from unit fu30 [POTENTIAL = *unit30*] in the calculation of rate constants for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$. Electronic structure information read from file fu30 (energies, geometries, and Hessians of reactants, saddle point, products and a series of 60 points along the reaction path) was calculated at the UHF/STO-3G level using GAMESS for generating the reaction path (ES1 integrator, mass-weighted Cartesian coordinates, step size = 0.005 amu^{1/2} a₀) and GAUSSIAN94 for calculating the unscaled forces and Hessians (run with geometry input in Cartesian coordinates and the NOSYMM keyword on) at the generalized TST structures every 0.05 amu^{1/2} bohr along the reaction path. The strategy for producing the electronic structure data was carefully selected in order to avoid the reorientation of geometries along the reaction path though the different steps of the process. At this level of electronic structure theory, CH₃ is nonplanar; therefore the symmetry number of the forward reaction is 2 [SIGMAF=2]. Geometries and forces are input in unscaled Cartesian coordinates in atomic units [LOPT(3) = 100 in unit fu30]. Hessians are also input in unscaled Cartesian coordinates in bohr using the packed format as obtained from the GAUSSIAN94 formatted checkpoint file [LOPT(4) = 99]. All normal modes are treated harmonically and TST, CVT, and ICVT rate constants [CVT and ICVT] are calculated at four temperatures [TEMP], using both MEPSAG and CD-SCSAG tunneling transmission factors [ZCT, SCT]. Curvature components are calculated using the derivative of the gradient with respect to *s* obtained from a quadratic fit of the gradient at three points on the basic input grid [LOPT(6) = 2] which is also called the sparse input grid. Note that the results obtained with this test run differ from those previously reported in Ref. 20 in Section 21. Several errors were detected in Ref. 20 (see Section 22.B in this manual for the corresponding errata).

POLYRATE I/O FILES

FILE NAME

ch5fu30tr5.dat
ch5fu30tr5.fu30
ch5fu30tr5.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu30 input data file
poly.fu6 long output file

14.C.25. Test run ch5fu30tr6

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ TST and CVT reaction rates with information along the reaction path read from fu30. Frequencies evaluated in redundant internal coordinates.

This is a sample run that illustrates the use of electronic structure input file data from unit fu30 [POTENTIAL = *unit30*] in the calculation of rate constants for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$. Electronic structure information calculated at the UHF/STO-3G level following the procedure explained in test run ch5fu30tr5 (energies, geometries, and Hessians of reactants, products, saddle point, and a series of 50 points along the reaction path) is read from file fu30. At this level of electronic structure theory, CH_3 is nonplanar; therefore the symmetry number of the forward reaction is 2 [SIGMAF=2]. Geometries and forces are input in unscaled Cartesian coordinates in atomic units [LOPT(3) = 100 in unit fu30]. Hessians are also input in unscaled Cartesian coordinates in atomic units using the packed format as obtained from the GAUSSIAN94 formatted checkpoint file [LOPT(4) = 99]. All normal modes are treated harmonically in redundant internal coordinates [COORD = *curv3*], and TST, CVT, and ICVT rate constants [CVT and ICVT] are calculated at four temperatures [TEMP], using both MEPSAG and CD-SCSAG tunneling transmission factors [ZCT, SCT].

The ‘Cartesian Forces’ from the GAUSSIAN94 formatted checkpoint file are actually Cartesian Gradients; use of them as forces will introduce errors in the transformation from rectilinear to curvilinear internal coordinates. Also, the gradients (or forces) must be un-normalized in order to carry out the transformation correctly.

POLYRATE I/O FILES

FILE NAME

ch5fu30tr6.dat
ch5fu30tr6.fu30
ch5fu30tr6.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu30 input data file
poly.fu6 long output file

14.C.26. Test run ch5fu31tr1

$\text{CH}_4 + \text{H} \rightarrow \text{CH}_3 + \text{H}_2$ IVTST-M-56/6 calculation using fu31

This sample run performs a IVTST-M-56/6 calculation using an fu31 input file. The curvature components are interpolated, the vibrational analysis is carried out in redundant internal coordinates, and ICVT rate constants are calculated. The information in the fu31 input file was obtained from the J1 potential energy surface by means of the WRITEFU31 keyword. Only the forward rate constant were calculated.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME
ch5fu31tr1.dat	poly.fu5 input data file
ch5fu31tr1.fu31	poly.fu31 input data file
ch5fu31tr1.fu6	poly.fu6 long output file
ch5fu31tr1.fu15	poly.fu15 summary output file

14.C.27. Test run ch5fu40tr1

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ TST reaction rate with frequencies read from fu40 directly

This is a conventional TST test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ [NOCVT and TST]. Electronic structure information calculated at the UHF/STO-3G level (energies, and frequencies of reactants, saddle point, and products) is read from file fu40 [POTENTIAL = *unit40*]. At this level of electronic structure theory, CH_3 is nonplanar; therefore the symmetry number of the forward reaction is 2 [SIGMAF=2]. Harmonic frequencies of reactants, products, and saddle point are input in atomic units [FREQSOURCE = *read* and FREQUIT = *au* in unit fu40] and geometries are input in unscaled Cartesian coordinates in bohrs [GEOMUNIT = *bohr* in unit fu40]. The rate constants are calculated at four temperatures [TEMP].

POLYRATE I/O FILES

FILE NAME

ch5fu40rt1.dat
ch5fu40tr1.fu40
ch5fu40tr1.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu40 input data file
poly.fu6 long output file

14.C.28. Test run ch5fu40tr2

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ TST reaction rate with information read from fu40 and full Hessian matrix for frequency input

This is a conventional TST test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ [NOCVT and TST]. Electronic structure information calculated at the UHF/STO-3G level (energies and full matrix format Hessians [HESSFORM = *full*] of reactants, saddle point, and products) is read from file fu40 [POTENTIAL = *unit40*]. At this level of electronic structure theory, CH_3 is nonplanar; therefore the symmetry number of the forward reaction is 2 [SIGMAF=2]. All reactants and products are treated harmonically. Hessians and geometries are input in unscaled Cartesian coordinates in atomic units [HESSUNIT = *bohr* and GEOMUNIT = *bohr*, respectively, in unit fu40]. The rate constants are calculated at four temperatures [TEMP].

POLYRATE I/O FILES

FILE NAME

ch5fu40tr2.dat
ch5fu40tr2.fu40
ch5fi40tr2.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu40 input data file
poly.fu6 long output file

14.C.29. Test run ch5fu40tr3

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ TST reaction rate with information read from fu40 and Hessian matrix input in packed form

This is a conventional TST test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ [NOCVT and TST]. Electronic structure information calculated at the UHF/STO-3G level (energies and Hessians of reactants, saddle point, and products in packed format [HESSFORM = *packed*]) is read from file fu40 [POTENTIAL = *unit40*]. At this level of electronic structure theory, CH_3 is nonplanar; therefore the symmetry number of the forward reaction is 2 [SIGMAF=2]. All reactants and products are treated harmonically (anharmonicity is not supported for TST-only calculations in POLYRATE-9.4.1). Hessians and geometries are input in unscaled Cartesian coordinates in bohrs [HESSUNIT = *bohr* and GEOMUNIT = *bohr*, respectively, in unit fu40]. The rate constants are calculated at four temperatures [TEMP].

POLYRATE I/O FILES

FILE NAME

ch5fu40tr3.dat
ch5fu40tr3.fu40
ch5fu40tr3.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu40 input data file
poly.fu6 long output file

14.C.30. Test run ch5fu40tr4

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ TST reaction rate with information read from fu40 and frequencies input directly.

This is a conventional TST test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ [NOCVT and TST]. Electronic structure information (energies and frequencies of reactants, saddle point, and products) at the QCISD/6-311G(d, p) level is read from file fu40 [POTENTIAL = *unit40*]. (The $V_{\text{MEP}}(s)$ has been scaled by a factor of 0.86 to be consistent with Ref. 48 in Section 20. The scaling was done by hand; however, the VSCALE keyword could have been used instead of hand scaling) At this level of electronic structure theory, CH_3 is planar; therefore the symmetry number of the forward reaction is 4 [SIGMAF=4]. Harmonic frequencies of reactants and products are input in wavenumbers [FREQSOURCE = *read*, FREQUUNIT = *waven*]. The rate constants are calculated at twelve temperatures [TEMP].

POLYRATE I/O FILES

FILE NAME

ch5fu40tr4.dat
ch5fu40tr4.fu40
ch5fu40tr4.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu40 input data file
poly.fu6 long output file

14.C.31. Test run ch5fu40tr5

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ TST and CVT reaction rates with information along the reaction path read from fu40

This is a sample run that illustrates the use of electronic structure input file data from unit fu40 [POTENTIAL = *unit40*] in the calculation of rate constants for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$. Electronic structure information read from file fu40 (energies, geometries, and Hessians of reactants, products, saddle point, and a series of 60 points along the reaction path) was calculated at the UHF/STO-3G level using GAMESS for generating the reaction path (ES1 integrator, mass-weighted Cartesian coordinates, step size = 0.005 amu^{1/2} a₀) and GAUSSIAN94 for calculating the unscaled forces and Hessians (run with geometry input in Cartesian coordinates and the NOSYMM keyword on) at the generalized TST structure every 0.05 amu^{1/2} bohr along the reaction path. The strategy for producing the electronic structure data was carefully selected in order to avoid the reorientation of geometries along the reaction path through the different steps of the process. At this level of electronic structure theory, CH₃ is nonplanar; therefore the symmetry number of the forward reaction is 2 [SIGMAF=2]. Geometries and forces are input in unscaled Cartesian coordinates in atomic units [GEOMUNIT = *bohr* and GRADUNIT = *bohr*, respectively, in unit fu40]. Hessians are also input in unscaled Cartesian coordinates in atomic units using the packed format as obtained from the GAUSSIAN94 formatted checkpoint file [HESSUNIT = *bohr* and HESSFORM = *packed*]. All normal modes are treated harmonically and TST, CVT, and ICVT rate constants [CVT and ICVT] are calculated at four temperatures [TEMP], using both MEPSAG and CD-SCSAG tunneling transmission factors [ZCT, SCT]. Curvature components are calculated using the derivative of the gradient with respect to *s* obtained from a quadratic fit of the gradient at three points on the basic input grid [CURVATURE = *compute*, GRADDER = *noextra*], which is also called the sparse input grid. Note that the results obtained with this test run differ from those previously reported in Ref. 20 in Section 21. Several errors were detected in Ref. 20 (see Section 22.B in this manual for the corresponding errata).

POLYRATE I/O FILES

FILE NAME

ch5fu40tr5.dat
ch5fu40tr5.fu40
ch5fu40tr5.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu40 input data file
poly.fu6 long output file

14.C.32. Test run ch5fu40tr6

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ TST and CVT reaction rates with information along the reaction path read from fu40. Frequencies are evaluated with redundant internal coordinates.

This is a sample run that illustrates the use of electronic structure input file data from unit fu40 [POTENTIAL = *unit40*] in the calculation of rate constants for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$. Electronic structure information calculated at the UHF/STO-3G level following the procedure explained in test run ch5fu40tr5 (energies, geometries, and Hessians of reactants, saddle point, products and a series of 50 points along the reaction path) is read from file fu40. At this level of electronic structure theory, CH_3 is nonplanar; therefore the symmetry number of the forward reaction is 2 [SIGMAF=2]. Geometries and forces are input in unscaled Cartesian coordinates in atomic units [GEOMUNIT = *bohr* and GRADUNIT = *bohr*, respectively, in unit fu40]. Hessians are also input in unscaled Cartesian coordinates in atomic units using the packed format as obtained from the GAUSSIAN94 formatted checkpoint file [HESSUNIT = *bohr* and HESSFORM = *packed*]. All normal modes are treated harmonically in redundant internal coordinates [*curv3*] and TST, CVT and ICVT rate constants [CVT and ICVT] are calculated at four temperatures [TEMP], using both MEPSAG and CD-SCSAG tunneling transmission factors [ZCT, SCT].

The quantities labeled as “Cartesian forces” in the GAUSSIAN94 formatted checkpoint file are actually Cartesian gradients, and using them as forces will introduce errors in the transformation from rectilinear to curvilinear internal coordinates. Also, the gradients (or forces) must be un-normalized in order to carry out the transformation correctly.

POLYRATE I/O FILES

FILE NAME

ch5fu40tr6.dat
ch5fu40tr6.fu40
ch5fu40tr6.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu40 input data file
poly.fu6 long output file

14.C.33. Test run ch5fu50tr1

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ zero-order IVTST reaction rate with information read from fu50 and frequencies input directly.

This is a conventional TST test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ [NOCVT and TST]. Electronic structure information calculated at the UHF/STO-3G level is read from file fu50 [IC in unit fu5 and 0IVTST in unit fu50]. All reactants and products are treated harmonically and the frequencies are input in cm^{-1} [RCFREQA, PCFREQA, SADFREQA, and FREQIMAG]. A rate constants are calculated at four temperatures [TEMP]. The similar calculation can be performed with the fu29 option. Both the ch5fu29tr1 and ch5fu29tr2 test runs can be used for checking the output from different input methods.

POLYRATE I/O FILES

FILE NAME

ch5fu50tr1.dat
ch5fu50tr1.fu50
ch5fu50tr1.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu50 input data file
poly.fu6 long output file

14.C.34. Test run ch5fu50tr2

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ VTST-IOC dual-level reaction rate calculation with information for the higher level read from fu50 and J1 analytic surface as the lower level.

This is a VTST-IOC test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ using an analytic surface as the lower level [IC and POTENTIAL = *hooks*]. The potential energy is not corrected in the nonadiabatic region in the LCT calculation [LCTCOR = *nocor*]. Electronic structure information calculated at the UHF/STO-3G level is read from file fu50 and the lower-level information is obtained from the J1 surface. All the parameters are the same as for test run ch5j1tr1.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5fu50tr2.dat	poly.fu5	input data file
ch5fu50tr2.fu50	poly.fu50	input data file
ch5fu50tr2.fu6	poly.fu6	long output file
ch5fu50tr2.fu14	poly.fu14	summary output file
ch5fu50tr2.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat	UNIT = 4
	FILE NAME = potch5.dat

14.C.35. Test run ch5fu50tr3

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ VTST-IOC dual-level reaction rate calculation with information for the higher level read from fu50 and J1 analytic surface as the lower level.

This is a VTST-IOC test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ using an analytic surface as the lower level [IC and POTENTIAL = *hooks*]. The potential energy is corrected in the nonadiabatic region in the LCT calculation based on a quadratic correction [LCTCOR = *quadlin*]. Electronic structure information calculated at the UHF/STO-3G level is read from file fu50 and the lower-level information is obtained from the J1 surface. All the parameters are the same as for test run ch5j1tr1.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5fu50tr3.dat	poly.fu5	input data file
ch5fu50tr3.fu50	poly.fu50	input data file
ch5fu50tr3.fu6	poly.fu6	long output file
ch5fu50tr3.fu14	poly.fu14	summary output file
ch5fu50tr3.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat	UNIT = 4
	FILE NAME = potch5.dat

14.C.36. Test run ch5fu51tr1

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ VTST-ISPE calculation with MP2/6-31G** and J1

This is a VTST-ISPE test run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ using an analytic surface as the lower-level [dl and potential = hooks]. Electronic structure information calculated at the MP2/6-31G** level is read in from file fu51 and the lower-level information is obtained from the J1 surface. Two extra points plus the stationary points used for the VTST-ISPE calculation were obtained from single-point energy calculations at the MP2/6-31G** level with the configurations obtained from the lower-level surface. All the parameters for the lower-level are the same as for the test run ch5j1tr1.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5fu51tr1.dat	poly.fu5	input data file
ch5fu51tr1.fu51	poly.fu51	VTST-ISPE input file
ch5fu51tr1.fu6	poly.fu6	long output file
ch5fu51tr1.fu14	poly.fu14	summary output file
ch5fu51tr1.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch5j1.dat UNIT=4
 FILE NAME=potch5.dat

14.C.37. Test run ch5icfu30tr1

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ VTST-ICL-DECKART calculation using file fu30 for the lower level information and rectilinear coordinates

This sample run uses the *unit30* [POTENTIAL = *unit30*] and the IOC [ICOPT: *icl*, *deckart*] options to invoke a dual-level direct dynamics calculation (variational transition state theory with interpolated optimized corrections) on the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$. The lower-level reaction path file is constructed at the UHF/STO-3G level, and the higher-level interpolated optimized corrections are calculated from the QCISD(T)/6-311G(d, p) level results. (The $V_{\text{MEP}}(s)$ has been scaled by a factor of 0.86 to be consistent with Ref. 48 in Section 20. The scaling was done by hand; however, the VSCALE keyword could have been used instead of hand scaling) All the modes are treated harmonically. Cartesian coordinates are used [COORD = *cart*]. Both small-curvature and zero-curvature tunneling calculations are carried out [ZCT, SCT]. The interpolated optimized correction is based on the logarithm of ratios [ICOPT: *icl*] and an Eckart fit is used to the energies at 3 stationary points (i.e. reactants, products, and saddle point) [ICOPT: *deckart*]. At the higher level of electronic structure theory employed, CH_3 is planar; therefore the symmetry number of the forward reaction is 4 [SIGMAF=4].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5icfu30tr1.dat	poly.fu5	input data file
ch5icfu30tr1.fu30	poly.fu30	input data file
ch5icfu30tr1.fu50	poly.fu50	input data file
ch5icfu30tr1.fu6	poly.fu6	output file

14.C.38. Test run ch5icfu30tr2

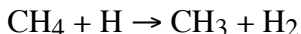
$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ VTST-IOC calculation using file fu30 for the lower level information and curvilinear coordinates

This sample run uses the *unit30* option [POTENTIAL = *unit30*] and the IOC [ICOPT: *ica, seckart*] options to invoke a dual-level direct dynamics calculation with interpolated optimized corrections on the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$. The lower-level reaction path file is constructed at the UHF/STO-3G level, and the higher-level interpolated optimized corrections are calculated from the QCISD(T)/6-311G(d, p) level results. (The $V_{\text{MEP}}(s)$ has been scaled by a factor of 0.86 to be consistent with Ref. 48 in Section 20. The scaling was done by hand; however, the VSCALE keyword could have been used instead of hand scaling) The original ICA interpolated correction method is used for the IC method. At the higher level of electronic structure theory employed, CH_3 is planar; therefore the symmetry number of the forward reaction is 4 [SIGMAF = 4].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5icfu30tr2.dat	poly.fu5	input data file
ch5icfu30tr2.fu30	poly.fu30	input data file
ch5icfu30tr2.fu50	poly.fu50	input data file
ch5icfu30tr2.fu6	poly.fu6	output file

14.C.39. Test run ch5icfu31tr1



VTST-IOC calculation using IVTST-M-56/6 data for the lower level of calculation and using the fu31 input file

This sample test run uses the *unit31* and the IOC options to perform a dual level direct dynamics calculation with interpolated optimized corrections. The lower level is an IVTST-M-6/56 calculation based on the information from a fu31 input file. The higher-level interpolated optimized corrections are calculated at the QCISD(T)/6-311G(d, p) level with $V_{\text{MEP}}(s)$ scaled by a factor of 0.86, as taken from Ref. 48 in Section 20. The ICA method is used for the correction of the frequencies, and the SECKART option is used for the energy correction.

POLYRATE I/O FILES

FILE NAME

ch5fu31tr1.dat
ch5fu31tr1.fu31
ch5fu31tr1.fu50
ch5fu31tr1.fu6
ch5fu31tr1.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu31 input data file
poly.fu50 input data file
poly.fu6 long output file
poly.fu15 summary output file

14.C.40. Test run ch5icfu40tr1

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ VTST-ICL-DECKART calculation using file fu40 for the lower level information and rectilinear coordinates

This sample run uses the *unit40* [POTENTIAL = *unit40*] and the IC [ICOPT: *icl, deckart*] options to invoke a dual-level direct dynamics calculation (variational transition state theory with interpolated optimized corrections) on the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$. The lower-level reaction path file is constructed at the UHF/STO-3G level, and the higher-level interpolated optimized corrections are calculated from the QCISD(T)/6-311G(d, p) level results with $V_{\text{MEP}}(s)$ scaled by a factor of 0.86 taken from Ref. 48 in Section 20. All the modes are treated harmonically. Cartesian coordinates are used [COORD = *cart*]. Both small-curvature and zero-curvature tunneling calculations are carried out [ZCT, SCT]. The interpolated correction is based on the logarithm of ratios [ICOPT: *icl*]. At the higher level of electronic structure theory employed, CH_3 is planar; therefore the symmetry number of the forward reaction is 4 [SIGMAF = 4].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5icfu40tr1.dat	poly.fu5	input data file
ch5icfu40tr1.fu40	poly.fu40	input data file
ch5icfu40tr1.fu50	poly.fu50	input data file
ch5icfu40tr1.fu6	poly.fu6	output file

14.C.41. Test run ch5icfu40tr2

$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ VTST-IOC calculation using file fu40 for the lower level information

This sample run uses the *unit40* [POTENTIAL = *unit40*] and the IOC [ICOPT: *ica, seckart*] options to invoke a dual-level direct dynamics calculation with interpolated optimized corrections on the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$. The lower-level reaction path file is constructed at the UHF/STO-3G level, and the higher-level interpolated optimized corrections are calculated from the QCISD(T)/6-311G(d, p) level results with $V_{\text{MEP}}(s)$ scaled by a factor of 0.86 taken from Ref. 48 in Section 20. The original ICA interpolated correction procedure is used for the IOC method. At the higher level of electronic structure theory employed, CH_3 is planar; therefore the symmetry number of the forward reaction is 4 [SIGMAF = 4].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch5icfu40tr2.dat	poly.fu5	input data file
ch5icfu40tr2.fu40	poly.fu40	input data file
ch5icfu40tr2.fu50	poly.fu50	input data file
ch5icfu40tr2.fu6	poly.fu6	output file

14.C.42. Test run `clhbrtr1`

$\text{Cl} + \text{HBr} \rightarrow \text{HCl} + \text{Br}$ with LCG3 calculations for all energetically accessible excited states in the LCG3 exit arrangement, and with the zero of energy of the reactants determined by the program

This sample run is for the reaction $\text{Cl} + \text{HBr} \rightarrow \text{HCl} + \text{Br}$ with an extended LEPS surface. All modes are treated using the default harmonic approximation. TST and CVT rate constants are computed [TST, CVT]. The tunneling calculations of ground-state transmission coefficients are performed using the MEPSAG, CD-SCSAG, and LCG3 methods [ZCT, SCT, LCTOPT]. The MEP is followed with the ESD integrator [RPM = *esd*] with a step size of 0.00001 bohr [SSTEP], and a generalized normal mode analysis is performed every 0.01 bohr [INH = 1000]. For the LCG3 calculation, the highest energetically open vibrational state in the LCG3 exit arrangement (HCl + Br) is determined and then all open vibrational states are included in the tunneling calculations [ALLEXCIT]. (As always, for LCG3 calculations only the ground state is considered for the LCG3 entrance arrangement, which is the reactant arrangement of the exoergic direction.) Also, the theta integrals, the vibrationally adiabatic potential, the vibrational period, and the sine of the angle between the MEP and the tunneling path at NGS0 points between each pair of adjacent energy quadrature points are obtained by 2-point interpolation [LCTOPT: *interpolate*]. This is done because the adiabatic potential of the first vibrational excited state is not monotonic on the reactant side and, therefore, unphysical results are obtained using interpolation with an order higher than 2. The curvature components are calculated using a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate [CURV = *dgrad*]. The zero of energy of the reactants is calculated from the energies of the individual molecular reactant species; in other words, the default value of EZERO = *calculate* is used. Note that the degeneracies and energies of the $^2\text{P}_{3/2}$ and $^2\text{P}_{1/2}$ states of Cl and Br are used in the calculation of the reactant and product electronic partition functions. However, for the transition state only the ground electronic state needs to be considered, because all excited electronic states are too high in energy to have a significant effect.

The results differ from those in Ref. 10 in Sect. 21 because Ref. 10 included anharmonicity and treated the bend in curvilinear internal coordinates, whereas the present run is harmonic and treats all vibrations in mass-scaled Cartesians. The treatment in Ref. 10 in Sect. 21 is more reliable.

POLYRATE I/O FILES

FILE NAME

clhbrtr1.dat
clhbrtr1.fu6
clhbrtr1.fu14
clhbrtr1.fu15

POLYRATE FILE NAME

poly.fu5	input data file
poly.fu6	long output file
poly.fu14	summary output file
poly.fu15	summary output file

14.C.43. Test run clhbrtr2

Cl + HBr \rightarrow HCl + Br with LCG3 calculations in which $n = 3$ is the highest excited state that is included in the LCG3 exit arrangement, and with the zero of energy of the reactants specified in the input

This sample run is similar to the clhbrtr1 test run. All vibrational modes are treated using the default harmonic approximation. TST and CVT rate constants are computed [TST, CVT]. The tunneling calculations of ground-state transmission coefficients are performed using the MEPSAG, CD-SCSAG, and LCG3 methods [ZCT, SCT, LCTOPT]. For the LCG3 calculation, the $n = 0, 1, 2,$ and 3 vibrational states in the LCG3 exit arrangement (HCl + Br) are included in the tunneling calculation [EXCITED = 3]. The highest energetically open vibrational state N_p^{\max} is determined [Ref. 30 in Sect. 21], and the LCG3 tunneling calculation includes the $n = 0$ to $\min[3, N_p^{\max}]$ vibrational states in the LCG3 exit arrangement. (For LCG3 calculations, only the ground state is considered for the LCG3 entrance arrangement, which is the reactant arrangement of the exoergic direction.) The MEP is followed by the ESD method [RPM = *esd*] with a step size of 0.00001 bohr [SSTEP] and a generalized normal mode analysis is performed every 0.01 bohr [INH = 1000]. Just as for test run clhbrtr1, we use 2-point interpolation [LCTOPT: *interpolate*] for the various quantities needed in the LCG3 calculations. The curvature components are calculated using a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate [CURV = *dgrad*]. The energy of the reactants, which is taken as the overall zero of energy, is read from unit fu5 [EZERO = *read*]. The results should be the same as those obtained in the clhbrtr1 test run.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
clhbrtr2.dat	poly.fu5	input data file
clhbrtr2.fu6	poly.fu6	long output file
clhbrtr2.fu14	poly.fu14	summary output file
clhbrtr2.fu15	poly.fu15	summary output file

14.C.44. Test run clhbrtr3

$\text{Cl} + \text{HBr} \rightarrow \text{HCl} + \text{Br}$ with LCG4 calculation for all the energetically accessible excited states in the LCG4 exit rearrangement, and with the zero of energy determined by the program

This sample is similar to the clhbrtr1 run. All vibrational modes are treated within the harmonic approximation. The only difference with test run clhbrtr1 is that large curvature transmission coefficients are obtained using the LCG4 method. At each tunneling energy a number of points along the straight path is $\text{NGTHETA} = 140$. The LCG4 tunneling probabilities are evaluated at $\text{NGAMP} = 40$ tunneling paths.

POLYRATE I/O FILES

FILE NAME

clhbrtr3.dat
clhbrtr3.fu6
clhbrtr3.fu14
clhbrtr3.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu6 long output file
poly.fu14 summary output file
poly.fu15 summary output file

14.C.45. Test run clhbrtr4

Cl + HBr \rightarrow HCl + Br with interpolated LCG4 calculation for all the energetically accessible excited states in the LCG4 exit rearrangement, and with the zero of energy determined by the program

This sample is similar to the previous clhbrtr3 test run. All vibrational modes are treated within the harmonic approximation. The difference with the previous run is that an interpolation is carried out, at each tunneling energy, by a splines-under-tension procedure. In this case only 3 points were considered at each tunneling energy [keywords ILCT and NILCGIT = 3]. Two of the three points are located at the beginning of the nonadiabatic region and the third point is located at half the distance between the two limiting points. This procedure still makes use of the keyword NGTHETA = 140 but the needed values are produced by the spline interpolation procedure. As in the previous run the calculation takes into account all the possible open vibrational states of the exit channel.

POLYRATE I/O FILES

FILE NAME

clhbrtr4.dat
clhbrtr4.fu6
clhbrtr4.fu14
clhbrtr4.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu6 long output file
poly.fu14 summary output file
poly.fu15 summary output file

14.C.46. Test run clhbr1tr5

Cl + HBr HCl + Br with interpolated LCG4 calculation with all the energetically accessible excited states in the LCG4 exit rearrangement. Restart from test run clhbrtr3.

This sample is the similar to the previous test run. The only differences are that this test run makes use of the ILCG2D interpolation option and that a restart file is used to build the 2D rectangular grid. In this case a quite rough grid of 5 5 is specified by LCTGRD.

POLYRATE I/O FILES

FILE NAME

chbrtr5.dat
chbrtr5.fu6
chbrtr5.fu14
chbrtr5.fu15
chbrtr5.fu48

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu6 long output file
poly.fu14 summary output file
poly.fu15 summary output file
poly.fu48 LCT restart file

14.C.47. Test run cmctr1



This sample run for the reaction $\text{Cl}^- + \text{CH}_3\text{Cl} \rightarrow \text{ClCH}_3 + \text{Cl}^-$ uses the default harmonic approximation for all vibrational modes. The MEP is calculated by the ESD integrator [RPM = *esd*] with a step size of 0.005 bohr [SSTEP], and a generalized normal mode analysis is performed every 0.025 bohr [INH = 5]. Both CD-SCSAG and MEPSAG tunneling probabilities are calculated [ZCT, SCT].

This test run and the next one (cmctr2) form a pair. This one must be run before the next one to find the s values of the wells that correspond to complexes on the reactant and product sides of the MEP. The interpolation in test run cmctr2 on both sides of the MEP is based on these wells, and thus we must first find the s values at these wells. A relatively long range of the MEP ($s = -7.1$ bohr to $s = 7.1$ bohr) is calculated [SRANGE] and we find that the s values at the wells are approximately 7.0 bohr and -7.0 bohr. We also learn from this run that in order for the tunneling calculation to converge, the MEP should be calculated at least from $s = -1.5$ bohr to $s = 1.5$ bohr. The saddle point frequencies match the frequencies of the complexes in descending order (except for the one corresponding to the reaction coordinate). We then use the information above together with some ab initio data to make the input data file for the VTST-IOC calculation in the cmctr2 test run.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
cmctr1.dat	poly.fu5	input data file
cmctr1.fu6	poly.fu6	long output file
cmctr1.fu14	poly.fu14	summary output file
cmctr1.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potcmc1.dat	UNIT = 4
	FILE NAME = potcmc1.dat
potcmc2.dat	UNIT = 7
	FILE NAME = potcmc2.dat

14.C.48. Test run cmctr2

$\text{Cl}^- + \text{CH}_3\text{Cl}^+ \rightarrow \text{ClCH}_3 + \text{Cl}^-$ harmonic approximation and VTST-IOC calculation

This sample run is similar to the cmctr1 test run except we perform a VTST-IOC calculation in addition to the VTST calculation [ICOPT] .

This reaction has two reactants and two products, but the interpolation on both sides of the saddle point is based on wells that correspond to complexes. We set the complex corresponding to the well on the product side to be the product so we can get its properties, and thus we set MEPTYPEP to *well* and MEPTYPE to *one*. The correct energy, frequency, and moment-of-inertia-tensor determinant information at reactants, products, saddle point, and the ion-molecule complexes are obtained from the calculations of Tucker and Truhlar [Ref. 34, Sect. 20]. The FREQMAT keyword is used, and the frequency matching information and the *s* value for the ion-molecule complex are obtained from the cmctr1 test run. The range parameters for the unit fu50 input are obtained by the utility programs findl.f and findb.f (see Subsection 15.D) by using additional points on the MEP that have an energy halfway between the energy of the saddle point and the energy of reactant- or product-side stationary point.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
cmctr2.dat	poly.fu5	input data file
cmcic.dat	poly.fu50	input data file
cmctr2.fu6	poly.fu6	long output file
cmctr2.fu14	poly.fu14	summary output file
cmctr2.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potcmc1.dat	UNIT = 4
	FILE NAME = potcmc1.dat
potcmc2.dat	UNIT = 7
	FILE NAME = potcmc2.dat

14.C.49. Test run cmctr3



This sample test run for the reaction $\text{Cl}^- + \text{CH}_3\text{Cl}^+ \rightarrow \text{ClCH}_3 + \text{Cl}^-$ uses the default harmonic approximation for all vibrational modes. The MEP is followed by the Euler steepest-descent integrator [ESD] with a step size of 0.005 bohr [SSTEP], and a generalized normal mode analysis is performed every 5 gradient steps [INH]. Both CD-SCSAG and MEPSAG tunneling probabilities are calculated [ZCT, SCT].

This test run corrects the energies along the MEP by the VTST-ISPE method. The energies of two extra points plus the stationary points were evaluated at the MP2/6-31G** level with the configuration taken from the potential energy surface. The MEP is calculated from $s = -1.5$ bohr to $s = 1.5$ bohr.

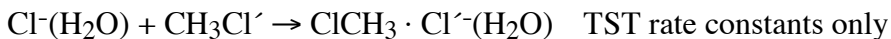
POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
cmctr3.dat	poly.fu5	input data file
cmctr3.fu31	poly.fu51	VTST-ISPE input file
cmctr3.fu6	poly.fu6	long output file
cmctr3.fu14	poly.fu14	summary output file
cmctr3.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potcmc1.dat	UNIT = 4 FILE NAME = potcmc1.dat
potcmc2.dat	UNIT = 7 FILE NAME = potcmc2.dat

14.C.50. Test run cwmctr1



In this sample run the reactant, product, and saddle point properties for the reaction $\text{Cl}^-(\text{H}_2\text{O}) + \text{CH}_3\text{Cl} \rightarrow \text{ClCH}_3 \cdot \text{Cl}^-(\text{H}_2\text{O})$, where $\text{Cl}^-(\text{H}_2\text{O})$ represents a chloride ion in the presence of a water molecule, are calculated, and only forward TST rate constants [TST, FORWARDK, NOCVT] are determined. No transmission coefficients are calculated nor is the minimum energy path determined (RATE and TUNNEL sections are not used). All normal modes and generalized normal modes are treated using the default harmonic approximation.

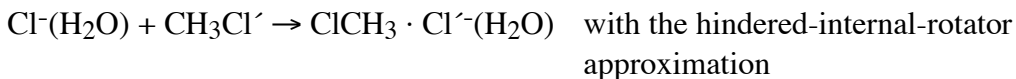
POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
cwmctr1.dat	poly.fu5	input data file
cwmctr1.fu6	poly.fu6	long output file
cwmctr1.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILES

potcwmc3b.dat	UNIT = 4	FILE NAME = potcwmc3b.dat
potcwmcvib.dat	UNIT = 7	FILE NAME =
potcwmcvib.dat		

14.C.51. Test run cwmctr2



In this sample run for the same reaction as in the cwmctr1 test run, the torsional mode of the generalized transition state is treated with the hindered-internal-rotator approximation [VANHAR, TOR]; all other modes are treated harmonically (default). The ESD integrator [RPM = *esd*] with a step size of 0.001 bohr [SSTEP] is used to follow the MEP, and a generalized normal mode analysis is performed every 0.01 bohr [INH = 10]. The curvature components are calculated using a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate [CURV = *dgrad*]. A restart file containing reaction-path information is written to unit fu1 [RESTART: WRITEFU1]. Rate constants are computed using CVT only [CVT]. The tunneling corrections are computed using the MEPSAG and CD-SCSAG methods [ZCT, SCT]. The rate constants are calculated for the same six temperatures as in the cwmctr1 test run [TEMP].

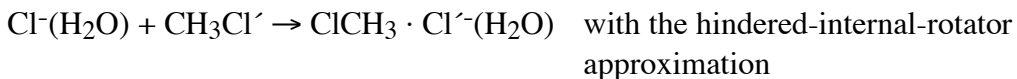
POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
cwmctr2.dat	poly.fu5	input data file
cwmctr2.fu6	poly.fu6	long output file
cwmctr2.fu1	poly.fu1	reaction path info.
cwmctr2.fu14	poly.fu14	summary output file
cwmctr2.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILES

potcwmc3b.dat	UNIT = 4	FILE NAME = potcwmc3b.dat
potcwmcvib.dat	UNIT = 7	FILE NAME = potcwmcvib.dat

14.C.52. Test run cwmctr3



In this sample run for the same reaction as in the cwmctr1 test run, the torsional mode of the generalized transition state is treated with the hindered-internal-rotator approximation with Pitzer's algorithm for the moment of inertia [VANHAR, TOR] and with $P = 3$ and $\sigma_1 = 3$; all other modes are treated harmonically (which is the default). The ESD integrator [RPM = *esd*] with a step size of 0.001 bohr [SSTEP] is used to follow the MEP, and a generalized normal mode analysis is performed every 0.01 bohr [INH = 10]. The curvature components are calculated using a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate [CURV = *dgrad*]. A restart file containing reaction-path information is written to unit fu1 [RESTART: WRITEFU1]. Rate constants are computed using CVT only [CVT]. The tunneling corrections are computed using the MEPSAG and CD-SCSAG methods [ZCT, SCT]. The rate constants are calculated for the same six temperatures as in the cwmctr1 test run [TEMP].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
cwmctr3.dat	poly.fu5	input data file
cwmctr3.fu6	poly.fu6	long output file
cwmctr3.fu1	poly.fu1	reaction path info.
cwmctr3.fu14	poly.fu14	summary output file
cwmctr3.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILES

potcwmc3b.dat	UNIT = 4	FILE NAME = potcwmc3b.dat
potcwmcvib.dat	UNIT = 7	FILE NAME = potcwmcvib.dat

14.C.53. Test run h3tr1

$\text{H}_2 + \text{H}' \rightarrow \text{HH}' + \text{H}$ with WKB and Morse IIIa approximations and Page-McIver integrator

This sample run uses the Page-McIver method [RPM = *pagem*] for calculating points along the MEP of the reaction $\text{H}_2 + \text{H}' \rightarrow \text{HH}' + \text{H}$ with a step size of 0.01 bohr [SSTEP], evaluates the Hessian matrix and performs a generalized normal mode analysis at every calculated point [INH = 1]. Note that the generalized transition state is restricted to linear symmetry [SPECIES = *lints*]. Two different methods are used to treat the vibrational modes [VANHAR]; vibrational modes of the reactant and product molecules and the highest frequency mode of the generalized transition state are treated by the Morse IIIa approximation [MORSE, MORMODEL = *morseiia*], while the two lowest vibrational modes of the generalized transition state are treated by the uniform WKB method applied to a quadratic-quartic fit to the potential along the normal mode coordinate [QQSEMI]. Extrapolation to both reactant and product limits is performed [EXFIRST, EXSECOND]. Both CVT [CVT] and ICVT [ICVT] rate constants are calculated only in the forward direction [FORWARDK] using both MEPSAG and CD-SCSAG tunneling corrections [ZCT, SCT], which are computed with two equal segments in the Boltzmann and theta integrals [NSEGBOLTZ = 2, NSEGTHETA = 2].

POLYRATE I/O FILES

FILE NAME

h3tr1.dat
h3tr1.fu6
h3tr1.fu14
h3tr1.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu6 long output file
poly.fu14 summary output file
poly.fu15 summary output file

14.C.54. Test run h3tr2

$\text{H}_2 + \text{H}' \rightarrow \text{HH}' + \text{H}$ with harmonic approximation and Page-McIver integrator

This sample run uses the Page-McIver method [RPM = *pagem*] for calculating points along the MEP of the reaction $\text{H}_2 + \text{H}' \rightarrow \text{HH}' + \text{H}$ with a step size of 0.005 bohr [SSTEP], evaluates the Hessian matrix and performs a generalized normal mode analysis at every 0.01 bohr [INH = 2]. The generalized transition states are restricted to linear symmetry [SPECIES = *lints*]. All vibrational modes are treated by the default harmonic approximation. Both CVT [CVT] and ICVT [ICVT] rate constants are calculated only in the forward direction [FORWARDK], using MEPSAG, CD-SCSAG, and LCG3 transmission coefficients [ZCT, SCT, LCTOPT], in which only the ground state is included in the LCG3 exit arrangement [NOALLEXCIT]. (We expect on physical grounds that only the ground state needs to be considered in this case, so this option is chosen to simplify the calculation.) The overall translations and rotations are projected out at stationary points; this is not necessary at stationary points, but it does improve the stability of the calculations.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
h3tr2.dat	poly.fu5	input data file
h3tr2.fu6	poly.fu6	long output file
h3tr2.fu14	poly.fu14	summary output file
h3tr2.fu15	poly.fu15	summary output file

14.C.55. Test run h3tr3

$H_2 + H' \rightarrow HH' + H$ with harmonic approximation, Page-McIver integrator, and symmetric reaction path option

This sample run is the same as test run h3tr2 except that the symmetric reaction-path option [SYMMETRY] is used, and LCG3 transmission coefficients are not calculated, i.e., only MEPSAG and CD-SCSAG transmission coefficients are calculated [ZCT, SCT].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
h3tr3.dat	poly.fu5	input data file
h3tr3.fu6	poly.fu6	long output file
h3tr3.fu14	poly.fu14	summary output file
h3tr3.fu15	poly.fu15	summary output file

14.C.56. Test run h3tr4

$\text{H}_2 + \text{H}' \rightarrow \text{HH}' + \text{H}$ with WKB and Morse IIIa approximations, Page-McIver integrator, and symmetric reaction path option

This sample run is the same as test run h3tr1 except that the symmetric reaction-path option is used [SYMMETRY].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
h3tr4.dat	poly.fu5	input data file
h3tr4.fu6	poly.fu6	long output file
h3tr4.fu14	poly.fu14	summary output file
h3tr4.fu15	poly.fu15	summary output file

14.C.57. Test run ho2tr1

OH + O → HO₂ with Morse I approximation

In this sample run, CVT, μ VT, and US rate constants [CVT, MUVT, MUVTOPT, US] are calculated at two temperatures [TEMP] for the reaction OH + O → HO₂. (The values of the conventional transition state theory rate constants, not computed in this sample calculation because there is no saddle point ([STATUS = 2] in START section and NOSADDLE in the PATH section), are set to 1.0 in the unit fu15 output.) The first step off the starting point is followed along the negative of the gradient direction [FIRSTSTEP = *gradi*]. All vibrational modes are treated by the Morse I approximation [MORSE, MORMODEL = *morsei*] The MEP is followed from the input initial geometry of the reactants by the ESD method [RPM = *esd*] with a step size of 0.0001 bohr [SSTEP], and a generalized normal mode analysis is performed every 0.031 bohr [INH = *310*]. Both CD-SCSAG and MEPSAG tunneling corrections are calculated [ZCT, SCT]. Note that the reverse rate constants are scaled by a factor of 1×10^{15} [REVKEXP = 15]. Thirty quadrature points were used in calculating the μ VT rate constants. Note that the degeneracies and energies of the $^2\Pi_{3/2}$ and $^2\Pi_{1/2}$ states of OH and 3P_2 , 3P_1 , and 3P_0 states of O are used in the calculation of the electronic partition functions. The results may be compared to those in Table V of Ref. 4 in Sect. 21.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ho2tr1.dat	poly.fu5	input data file
ho2tr1.fu6	poly.fu6	long output file
ho2tr1.fu14	poly.fu14	summary output file
ho2tr1.fu15	poly.fu15	summary output file

14.C.58. Test run ho2tr2

OH + O → HO₂ with Morse I approximation using curvilinear coordinates

In this sample run, CVT, μ VT, and US rate constants [CVT, MUVT, MUVTOPT, US] are calculated at two temperatures [TEMP] for the reaction OH + O → HO₂ using curvilinear coordinates (O-H and O-O stretches and an H-O-O bend). (The values of the conventional transition state theory rate constants, not computed in this sample calculation because there is no saddle point [STATUS = 2 in START section and NOSADDLE in the PATH section], are set to 1.0 in the unit fu15 output.) The MEP is followed from the input initial geometry of the reactants by the ESD method [RPM = *esd*] with a step size of 0.0001 bohr [SSTEP], and a generalized normal mode analysis is performed every 0.031 bohr [INH = 310]. Both CD-SCSAG and MEPSAG tunneling corrections are calculated [ZCT, SCT]. Note that the reverse rate constants are scaled by a factor of 1×10^{15} [REVKEXP = 15].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ho2tr2.dat	poly.fu5	input data file
ho2tr2.fu6	poly.fu6	long output file
ho2tr2.fu14	poly.fu14	summary output file
ho2tr2.fu15	poly.fu15	summary output file

14.C.59. Test run ho2tr3

OH + O → HO₂ with WKB approximation for one mode

This sample run for the reaction OH + O → HO₂ uses a combination of methods for the treatment of the vibrational modes [VANHAR]. All modes of the reactant and product species are treated by the default harmonic approximation. The lowest frequency mode of the generalized transition state is also treated harmonically, while the highest frequency mode is treated by the primitive WKB method applied to a quadratic-quartic fit of the potential along the normal mode coordinate [QQWKB]. (The values of the conventional transition state theory rate constants, not computed in this sample calculation because there is no saddle point [STATUS = 2 in START section and NOSADDLE in the PATH section], are set to 1.0 in the unit fu15 output). Points along the MEP are calculated with the ESD integrator [RPM = *esd*] with a step size of 0.0001 bohr [SSTEP]. A generalized normal mode analysis, together with a Hessian matrix evaluation, is performed every 0.155 bohr [INH = *1550*]. A generalized normal mode analysis is also performed at eight special save points [SSPECIAL]. CVT [CVT] and ICVT [ICVT] rate constants are calculated at one temperature [TEMP], using both MEPSAG and CD-SCSAG tunneling corrections [ZCT, SCT]. Sixth-order Lagrangian interpolation is used to obtain the values of the effective mass for CD-SCSAG tunneling [SCTOPT: *lagrange*]. The reverse rate constants are scaled by a factor of 1×10^{15} [REVEXP = 15]. Also, restart information is written to unit fu1 [RESTART: WRITEFU1].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ho2tr3.dat	poly.fu5	input data file
ho2tr3.fu6	poly.fu6	long output file
ho2tr3.fu1	poly.fu1	reaction path info.
ho2tr3.fu14	poly.fu14	summary output file
ho2tr3.fu15	poly.fu15	summary output file

14.C.60. Test run ho2tr4

$O_2 + H \rightarrow HO_2$ with Morse I approximation

This sample run for the reaction $O_2 + H \rightarrow HO_2$ uses the Morse I approximation for all vibrational modes [MORSE, MORMODEL = *morsei*]. The MEP is calculated by the ESD integrator [RPM = *esd*] with a step size of 0.0002 bohr [SSTEP], and a generalized normal mode analysis is performed every 0.01 bohr [INH = 50]. Both CD-SCSAG and MEPSAG tunneling probabilities are calculated [ZCT, SCT].

This test run forms a set with the ho2tr5 test run. Since the MEP is finite on the product side of the saddle point and the next test run involves a VTST-IOC calculation, before running it we must find the s value at the product. Thus, in this run we calculate a relatively long range of the MEP ($s = -2.5$ bohr to $s = 2.5$ bohr) [SRANGE] and find that the s value at the product is approximately 2.20 bohr. We also learn from this test run that in order for the tunneling calculation to converge, the MEP should be calculated at least from $s = -1.8$ bohr to $s = 0.8$ bohr. Finally we learn that the highest saddle point frequency becomes the reactant frequency at $s = -\infty$, and the second and third highest saddle point frequencies become the product frequencies at $s = 2.20$ bohr. The lowest product frequency corresponds to the reaction coordinate. We then use the information above together with some ab initio data to make the input data file for the VTST-IOC calculation in test run ho2tr5.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ho2tr4.dat	poly.fu5	input data file
ho2tr4.fu6	poly.fu6	long output file
ho2tr4.fu14	poly.fu14	summary output file
ho2tr4.fu15	poly.fu15	summary output file

14.C.61. Test run ho2tr5

$O_2 + H \rightarrow HO_2$ with Morse I approximation and VTST-IOC calculation

This sample run is similar to the ho2tr4 test run except we perform a VTST-IOC calculation in addition to the VTST calculation [ICOPT].

This reaction has two reactants and one product and a shallow well on the reactant side of MEP. In this test run we neglect the well and set the MEPTYPEP to *two* and MEPTYPEP to *one*. The accurate energy and frequency information and the geometry for calculating the determinant of the moment of inertia tensor at the reactants and product are obtained from *J. Chem. Phys.* **74**, 3482 (1981). The corrected barrier height is set to 1.0 kcal/mol and the corrected saddle point frequencies are obtained by interpolating the correction at reactants and product. Frequency matching information is input by the FREQMAT keyword. We use the DETMI keyword to input the corrected determinant of the moment of inertia tensor, and we use UNITMI to select atomic units input. The frequency-matching information and the *s* value for the product are obtained from the test run ho2tr4. The range parameters in the unit fu50 input are determined by the utility programs findl.f and findb.f (see Subsection 15.D) by using additional points on the MEP that have energies halfway between the energy of the saddle point and the energies of the reactant- and product-side stationary points.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ho2tr5.dat	poly.fu5	input data file
ho2ic.dat	poly.fu50	input data file
ho2tr5.fu6	poly.fu6	long output file
ho2tr5.fu14	poly.fu14	summary output file
ho2tr5.fu15	poly.fu15	summary output file

14.C.62. Test run nh3tr1

NH₃ → NH₃ (inversion) with harmonic approximation and VTST-IOC calculation

This sample run is for the unimolecular reaction corresponding to the unimolecular inversion of ammonia between two pyramidal configurations, using the default harmonic approximation for all vibrational modes. The MEP is calculated by the ESD method [RPM = *esd*] between $s = 1.18$ bohr and $s = -1.18$ bohr [SRANGE] with a step size of 0.0001 bohr [SSTEP]. A generalized normal mode analysis is performed every 0.01 bohr [INH = *100*]. MEPSAG and CD-SCSAG tunneling probabilities are calculated [ZCT, SCT], and they can be used to study the tunneling splitting in this reaction. Rate constants are not calculated [NOTST, NOCVT].

The VTST-IOC option is selected [ICOPT] and the accurate energy and frequency information and the geometry for calculating the determinant of the moment of inertia tensor are obtained from *J. Chem. Phys.* **93**, 6570 (1990). The range parameters in the unit fu50 input are determined by the utility programs findl.f and findb.f (see Subsection 15.D) by using additional points on the MEP that have an energy halfway between the energy of the saddle point and the reactant/product energy.

POLYRATE I/O FILES

FILE NAME

nh3tr1.dat
nh3ic.dat
nh3tr1.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu50 input data file
poly.fu6 long output file

14.C.63. Test run nh3tr2

NH₃ → NH₃ (inversion) with harmonic approximation VTST calculation

This sample run is for the unimolecular reaction corresponding to the unimolecular inversion of ammonia between two pyramidal configurations, using the default harmonic approximation for all vibrational modes. The MEP is calculated by the ESD method [RPM = *esd*] between $s = 1.18$ bohr and $s = -1.18$ bohr [SRANGE] with a step size of 0.0001 bohr [SSTEP]. A generalized normal mode analysis is performed every 0.01 bohr [INH = 100]. The TST and CVT rate constants are calculated. The partition functions and the internal free energy of reactant are printed out with PRPART = *rpt*. The GTLOG keyword is used so the generalized free energy of activation is calculated using logarithms of the individual partition functions.

This is a somewhat artificial example (because it is a degenerate rearrangement and because ordinary pressures are too low to stabilize the system in on or another well), it is used for testing the code.

POLYRATE I/O FILES

FILE NAME

nh3tr2.dat
nh3tr2.fu6
nh3tr2.fu14
nh3tr2.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu6 input data file
poly.fu14 summary output file
poly.fu15 summary output file

14.C.64. Test run oh3tr1

OH + H₂ → H₂O + H with harmonic approximation and using the VRP(ESD) algorithm

This sample run is for the OH + H₂ → H₂O + H reaction. The zero of energy of the reactants is calculated from the energies of the individual molecular reactant species; in other words, the default value of EZERO = *calculate* is used. All normal modes and generalized normal modes are treated using the default harmonic approximation. The VRP(ESD) algorithm [RPM = *vrpe*] with a step size of 0.001 bohr [SSTEP] is used to follow the MEP, while the initial step from the saddle point is based on a cubic expansion of the potential energy surface [FIRSTSTEP = *cubic*]. The first five steps from the saddle point geometry in each direction are of size 0.0001 bohr [SFIRST]. The Hessian matrix is recalculated and a generalized normal mode analysis is performed every 0.01 bohr [INH = 10].

MEPSAG and CD-SCSAG tunneling corrections [ZCT, SCT] are computed, and both CVT [CVT] and ICVT [ICVT] rate constants are calculated at eight temperatures [TEMP]. The derivative of the gradient with respect to the reaction coordinate is found by a quadratic fit [CURV = *dgrad*]. In addition, in this test run as well as in test runs oh3tr2–oh3tr6, extrapolation is used for the first direction from the saddle point (i.e., toward reactants) [EXFIRST] .

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3tr1.dat	poly.fu5	input data file
oh3tr1.fu6	poly.fu6	long output file
oh3tr1.fu14	poly.fu14	summary output file
oh3tr1.fu15	poly.fu15	summary output file

14.C.65. Test run oh3tr2

OH + H₂ → H₂O + H with Morse I-quadratic-quartic approximation

This sample run for the OH + H₂ → H₂O + H reaction uses the Morse I and quadratic-quartic approximations [MORSEQQ, MORMODEL = *morsei*] for the normal and generalized normal modes, i.e., modes with nonzero cubic anharmonicity are treated by the Morse I model, while modes with zero cubic anharmonicity are approximated as quadratic-quartic oscillators. Both the CVT [CVT] and ICVT [ICVT] methods are used. The rate constants are calculated at twelve temperatures [TEMP] and the activation energies are computed for three temperature pairs [EACT]. The ESD integrator [RPM = *esd*] with a step size of 0.0001 bohr [SSTEP] is used to follow the MEP, and a generalized normal mode analysis is performed every 0.01 bohr [INH = 100]. Both MEPSAG and CD-SCSAG tunneling corrections [ZCT, SCT] are computed. Note that the derivative of the gradient with respect to the reaction coordinate is computed by the one-sided difference method [CURV = *oneside*] and the free energy of activation profile is calculated over a different range of reaction coordinate for each temperature [GTEMP]. The results can be compared to those in Table IV of Ref. 1 in Sect. 21.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3tr2.dat	poly.fu5	input data file
oh3tr2.fu6	poly.fu6	long output file
oh3tr2.fu14	poly.fu14	summary output file
oh3tr2.fu15	poly.fu15	summary output file

14.C.66. Test run oh3tr3

OH ($\nu = 0$) + H₂ → H₂O + H with WKB and harmonic approximations

This sample run for the OH ($\nu = 0$) + H₂ → H₂O + H reaction uses the WKB approximation for the ground-state energy of all normal modes [WKB], and the default harmonic approximation for all excited-state energies. The overall translations and rotations are not projected out at stationary points [NOPROJECT]. The MEP is followed with the Euler stabilization (ES1) integrator [RPM = *es1*], with a step size of 0.001 bohr [SSTEP], and the initial step from the saddle point is based on a cubic expansion of the energy surface [FIRSTSTEP = *cubic*]. A value different from the default value for the stabilization criterion (magnitude of the vector difference between successive gradients) is used; it is set equal to 0.001 [ES1OPT: *diffd*]. A generalized normal mode analysis is performed every 0.02 bohr [INH = *20*]. CVT [CVT] and ICVT [ICVT] rate constants are calculated at eight temperatures [TEMP]. Both MEPSAG and CD-SCSAG tunneling corrections are calculated [ZCT, SCTOPT], and the curvature components are found by using a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate [CURV = *dgrad*]. Sixth-order Lagrangian interpolation is used to obtain the values of the effective mass for the CD-SCSAG tunneling calculation [SCTOPT: LAGRANGE].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3tr3.dat	poly.fu5	input data file
oh3tr3.fu6	poly.fu6	long output file
oh3tr3.fu14	poly.fu14	summary output file
oh3tr3.fu15	poly.fu15	summary output file

14.C.67. Test run oh3tr4

Adiabatic OH ($v = 0$) + H₂($v = 1$) → H₂O + H with Morse I quadratic-quartic approximation

This sample run is an example of the calculation of adiabatic state-selected rate constants [STATE = *adiab*] for the reaction OH ($v = 0$) + H₂($v = 1$) → H₂O + H. The highest-frequency normal mode of the generalized transition state is restricted to the first excited state, while the second highest frequency mode is restricted to the ground state. All other normal modes of the generalized transition state are treated thermally. For the reactants, the stretching mode of OH is restricted to the ground state, and the stretching mode of H₂ is restricted to the first excited state. Finally, for the product H₂O, the highest normal mode is restricted to the excited state, the second highest mode is restricted to the ground state, and the lowest mode is treated thermally. The Page-McIver local-quadratic approximation integrator [RPM = *pagem*] with a step size of 0.001 bohr [SSTEP] is used to follow the MEP, the Hessian matrix is recalculated and a generalized normal mode analysis is performed every 0.01 bohr [INH = 10]. All other options and input are the same as for test run oh3tr2 except that only the forward rate constants are calculated [FORWARDK].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3tr4.dat	poly.fu5	input data file
oh3tr4.fu6	poly.fu6	long output file
oh3tr4.fu14	poly.fu14	summary output file
oh3tr4.fu15	poly.fu15	summary output file

14.C.68. Test run oh3tr5

Diabatic OH + H₂($\nu = 1$) → H₂O + H with Morse I quadratic-quartic approximation

This sample run is an example of the calculation of diabatic state-selected rate constants [STATE = *diab*] for the reaction OH + H₂($\nu = 1$) → H₂O + H.

The adiabatic/diabatic switch occurs at $s = -0.31$ bohr. For $s < -0.31$ bohr, the highest and second-highest frequency normal modes of the generalized transition state are restricted to the first excited and ground state, respectively. For $s > -0.31$ bohr, the highest-frequency normal mode is restricted to the ground state, while the second-highest normal mode is restricted to the first excited state. The lowest three normal modes of the GTS are all treated thermally for all values of s . For the reactants, the OH stretch vibration is restricted to the ground state, while the H₂ vibration is restricted to the first excited state. The highest and second-highest frequency normal modes of the product H₂O are restricted to the ground state and first excited state, respectively, while the lowest normal mode is treated thermally. Other options are the same as those for test run oh3tr2, except that only forward rate constants are calculated [FORWARDK] and the free energies are calculated over the same range of s for all temperatures [GSPEC]. The results may be compared to those in Table I of Ref. 3 in Sect. 21.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3tr5.dat	poly.fu5	input data file
oh3tr5.fu6	poly.fu6	long output file
oh3tr5.fu14	poly.fu14	summary output file
oh3tr5.fu15	poly.fu15	summary output file

14.C.69. Test run oh3tr6

OH + H₂ → H₂O + H with harmonic approximation and the ES1 integrator

This sample run is for the OH + H₂ → H₂O + H reaction. All normal modes and generalized normal modes are treated using the default harmonic approximation. The ES1* integrator [RPM = *es1*, ES1OPT: DELTA2 = SSTEP and DIFFD = 0.01] with a step size of 0.01 bohr [SSTEP] is used to follow the MEP, while the initial step from the saddle point is based on a cubic expansion of the potential energy surface [FIRSTSTEP = *cubic*]. The Hessian matrix is recalculated and a generalized normal mode analysis is performed every 0.01 bohr [INH = 1].

MEPSAG and CD-SCSAG tunneling corrections [ZCT, SCT] are computed, and both CVT [CVT] and ICVT [ICVT] rate constants are calculated at eight temperatures [TEMP]. The derivative of the gradient with respect to the reaction coordinate is found by a quadratic fit [CURV = *dgrad*].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3tr6.dat	poly.fu5	input data file
oh3tr6.fu6	poly.fu6	long output file
oh3tr6.fu14	poly.fu14	summary output file
oh3tr6.fu15	poly.fu15	summary output file

14.C.70. Test run oh3tr7

$\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$ with MEPSAG and CD-SCSAG calculations at two temperatures (one of the test cases in Ref. 27 in Sect. 21), WRITEFU30 option

This sample run for the $\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$ reaction is taken from one of the test cases in Ref. 27 in Sect. 21. (However, Ref. 27 used the obsolete SCSAG method whereas the current code uses CD-SCSAG.) The MEP is calculated from $s = -1.0$ bohr to $s = 0.3$ bohr [SRANGE] using the Page-McIver method [RPM = *pagem*] with the cubic starting algorithm [FIRSTSTEP = *cubic*]. The gradient and save step sizes are both set to 0.062 bohr [SSTEP, INH = 1]. The CVT/MEPSAG and CVT/CD-SCSAG rate constants are calculated [CVT, ZCT, SCT] at 200K and 300K [TEMP]. All the rate constants are within 15% of the converged values obtained in the test run oh3tr1. Also, a file is written out in unit fu30 format as an electronic structure input file with the WRITEFU30 option. The points are not in the required ascending order. Therefore, this generated fu30 file needs to be reordered by the user before using the file.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3tr7.dat	poly.fu5	input data file
oh3tr7.fu6	poly.fu6	long output file
oh3tr7.fu14	poly.fu14	summary output file
oh3tr7.fu15	poly.fu15	summary output file
tr7old.fu30	poly.fu30	old fu30 file
oh3tr7.fu30	poly.fu30	reordered fu30 file

14.C.71. Test run oh3tr8

OH + H₂ → H₂O + H with MEPSAG and CD-SCSAG calculations at two temperatures, one of the test cases in Ref. 27 in Sect. 21, using curvilinear internal coordinates

This sample run for the OH + H₂ → H₂O + H reaction is taken from one of the test cases in Ref. 27 in Sect. 21. (However, Ref. 27 used the obsolete SCSAG method, whereas the current code uses the CD-SCSAG method that has replaced it. Other changes are that here curvilinear internal coordinates [COORD = *curv2*] are used here to calculate bound-state vibrational frequencies and generalized normal modes, and that the run uses SPECSTOP keyword. The curvilinear internal coordinates used for the O-H + H'-H'' → H-O-H' + H'' hydrogen atom transfer reaction include the O-H, H'-H'', and O-H' stretches, the H-O-H' and H''-H'-O bends, and the H''-H'-O-H dihedral [INTDEF]. The MEP is calculated from $s = -1.0$ bohr to $s = 1.0$ bohr [SRANGE] or until the V_a^G curve goes below 95% from its value at the saddle point to its value at the reactants [SPECSTOP]. The MEP is determined using the Page-McIver method [RPM = *pagem*] with the cubic starting algorithm [FIRSTSTEP = *cubic*]. The gradient and save step sizes are both set to 0.062 bohr [SSTEP, INH = 1]. The CVT/MEPSAG and CVT/CD-SCSAG rate constants are calculated [CVT, ZCT, SCT] at 200K and 300K [TEMP].

POLYRATE I/O FILES

FILE NAME

oh3tr8.dat
oh3tr8.fu6
oh3tr8.fu14
oh3tr8.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu6 long output file
poly.fu14 summary output file
poly.fu15 summary output file

14.C.72. Test run oh3tr9

OH + H₂ → OH₂ + H VTST calculation including a well on the reactant side and writing a fu31 file

This sample test run writes the fu31 input file from the information calculated using an analytical potential energy surface. The run does not require any fu31 input file and the resulting fu31 file is ready to be used without any change. In addition, a well that is 1.28 kcal/mol more stable than the reactants is optimized, and its frequencies are calculated. The data is written to the fu31 file.

The vibrational analysis is carried out in curvilinear coordinates, and debugging information on this analysis is printed out to the fu6 output file.

POLYRATE I/O FILES

FILE NAME

ch5fu31tr1.dat
ch5fu31tr1.fu31
ch5fu31tr1.fu50
ch5fu31tr1.fu6
ch5fu31tr1.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu31 input data file
poly.fu50 input data file
poly.fu6 long output file
poly.fu15 summary output file

14.C.73. Test run oh3tr10

OH + H₂ → H₂O + H with harmonic approximation and an ESD/RODS reaction path

This sample run uses the WRITEFU31 keyword to generate an fu31 file, and it uses the RODS keyword to make up for deficiencies in the path. All normal modes and generalized normal modes are treated using the default harmonic approximation. The ESD integrator [RPM = *esd*] with a step size of 0.05 bohr [SSTEP] is used to follow the MEP, and the RODS algorithm [RODS] is used to correct the energies and frequencies along the path. The Hessian matrix is recalculated and a generalized normal mode analysis is performed every 0.05 bohr [INH = 1].

MEPSAG tunneling corrections [ZCT] are computed, and CVT [CVT] rate constants are calculated at eight temperatures [TEMP]. The derivative of the gradient with respect to the reaction coordinate is found by a quadratic fit [CURV = *dgrad*].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3tr10.dat	poly.fu5	input data file
oh3tr10.fu6	poly.fu6	long output file
oh3tr10.fu14	poly.fu14	summary output file
oh3tr10.fu15	poly.fu15	summary output file
oh3tr10.fu31	poly.fu31	input data file

14.C.74. Test run oh3fu30tr1

OH + H₂ → H₂O + H using electronic structure input file data from file fu30 and curvilinear internal coordinates

This sample run demonstrates the use of electronic structure input file data in the calculation of conventional TST rate constants [POTENTIAL = *unit30*] for the reaction OH + H₂ → H₂O + H. Hessians are read in packed format in mass-scaled coordinates [LOPT(4) = -2 in unit fu30 input]. Note that even though the reaction path is not required for a conventional TST rate constant calculation, the PATH section must be included in the unit fu5 input in order to pass to the program the value of the factor used in scaling the coordinates [SCALEMASS in the unit fu5 input], and the data for at least two points along the reaction path have to be input from unit fu30. The curvilinear internal coordinates [COORD = *curv2*] used for the O-H + H'-H'' → H-O-H' + H'' hydrogen atom transfer reaction include the O-H, H'-H'', and O-H' stretches, the H-O-H' and H''-H'-O bends, and the H''-H'-O-H dihedral [INTDEF]. All normal modes are treated harmonically [HARMONIC in the unit fu5 input].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3fu30tr1.dat	poly.fu5	input data file
oh3fu30tr1.fu30	poly.fu30	input data file
oh3fu30tr1.fu6	poly.fu6	long output file
oh3fu30tr1.fu14	poly.fu14	summary output file
oh3fu30tr1.fu15	poly.fu15	summary output file

14.C.75. Test run oh3fu30tr2

OH + H₂ → H₂O + H using electronic structure input file data from file fu30 and curvilinear internal coordinates

This sample run demonstrates the use of electronic structure input file data in the calculation of rate constants [POTENTIAL = *unit30*] for the reaction OH + H₂ → H₂O + H. This type of calculation requires the unit fu30 input file. The normal mode and generalized normal mode frequencies are obtained at each save point on the MEP by first transforming the Cartesian Hessian matrix into curvilinear internal coordinate space, projecting out the gradient direction, and then diagonalizing the projected Hessian using the Wilson **GF** method. The curvilinear internal coordinates [COORD = *curv2*] used for the O-H + H'-H'' → H-O-H' + H'' hydrogen atom transfer reaction include the O-H, H'-H'', and O-H' stretches, the H-O-H' and H''-H'-O bends, and the H''-H'-O-H dihedral [INTDEF]. The curvature components are computed using a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate [LOPT(6) = 4]. All normal modes and generalized normal modes are treated harmonically [HARMONIC in the unit fu5 input], and only TST and CVT rate constants are calculated [CVT] at four temperatures [TEMP], using both MEPSAG and CD-SCSAG tunneling corrections [ZCT, SCT].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3fu30tr2.dat	poly.fu5	input data file
oh3fu30tr2.fu30	poly.fu30	electronic structure input file data
oh3fu30tr2.fu6	poly.fu6	long output file
oh3fu30tr2.fu14	poly.fu14	summary output file
oh3fu30tr2.fu15	poly.fu15	summary output file

14.C.76. Test run oh3fu30tr3

OH + H₂ → H₂O + H direct dynamics calculation using the electronic structure data read from file fu30, rectilinear coordinates, and IVTST-M-4/4 interpolation.

This sample run demonstrates the use of electronic structure input file data read from unit fu30 in the calculation of rate constants [POTENTIAL = *unit30*] for the reaction OH + H₂ → H₂O + H. This type of calculation requires the unit fu30 input file. The interpolation is done using IVTST-M-4/4 [LOPT(2) = 500]. Note that the normal mode and generalized normal mode frequencies are obtained by diagonalization of the matrix of the second derivatives of the potential with respect to the mass-scaled Cartesian coordinates, which is read in packed form from unit fu30 [LOPT(4) = -2 in the unit fu30 input], and no anharmonic data are included [LOPT(5) = 0]. All normal modes are treated harmonically [HARMONIC in the unit fu5 input], and only TST and CVT rate constants are calculated [CVT] at four temperatures [TEMP], using both MEPSAG and CD-SCSAG tunneling corrections [ZCT, SCT].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3fu30tr3.dat	poly.fu5	input data file
oh3fu30tr3.fu30	poly.fu30	input data file
oh3fu30tr3.fu6	poly.fu6	long output file
oh3fu30tr3.fu14	poly.fu14	summary output file
oh3fu30tr3.fu15	poly.fu15	summary output file

14.C.77. Test run oh3fu30tr4

$\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$ using electronic structure input file data from file fu30 for generating the next point along the reaction path (reactants valley).

This sample run demonstrates the use of the NEXTPT keyword for generating the next point along the reaction path in conjunction with the use of electronic structure input from file fu30 [POTENTIAL = *unit30*] for the reaction $\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$. Note that the print option LOPT(1) and IFITV are set equal to 0 in file fu30 because nonzero values for these variables are incompatible with NEXTPT. Electronic structure information is included for two points on the reaction path (one on each side of the saddle point), with the first one having a positive s and the second one having a negative s (thus the points are not in increasing order of s , but this is permitted when NEXTPT is on). Since the last point has a negative s , IDIRECT is set equal to -1. The next point is generated on the reactants side using a step size of $0.0005 \text{ amu}^{1/2} \text{ bohr}$ [SSTEP].

POLYRATE I/O FILES

FILE NAME

oh3fu30tr4.dat
oh3fu30tr4.fu30
oh3fu30tr4.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu30 input data file
poly.fu6 long output file

14.C.78. Test run oh3fu30tr5

$\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$ using electronic structure input file data from file fu30 for generating the next point along the reaction path (products valley).

This sample run demonstrates the use of the NEXTPT keyword for generating the next point along the reaction path in conjunction with the use of electronic structure input from file fu30 [POTENTIAL = *unit30*] for the reaction $\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$. Note that the print option LOPT(1) and IFITV are set equal to 0 in file fu30 because nonzero values for these variables are incompatible with NEXTPT. Electronic structure information is included for two points on the reaction path (one on each side of the saddle point), with the last one being in the products valley. Since the last point has a positive *s*, IDIRECT is +1. The next point is generated on the products side using a step size of 0.0005 amu^{1/2} bohr [SSTEP].

POLYRATE I/O FILES

FILE NAME

oh3fu30tr5.dat
oh3fu30tr5.fu30
oh3fu30tr5.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu30 input data file
poly.fu6 long output file

14.C.79. Test run oh3fu30tr6

$\text{OH} + \text{H}_2 \rightarrow \text{OH}_2 + \text{H}$ fu30 calculation including a well on the reactant side

The oh3fu30tr6 test run is a sample run that uses file fu30 and includes a well on the reactant side in the calculation. The reaction path is interpolated by means of third order Lagrangian interpolation using the points along the reaction path including the reactant well. The file fu30 was created with the writefu30 option, with only two changes being necessary: to input of the value of the reaction coordinate, s , at the reactants well, and to reorder of the points.

POLYRATE I/O FILES

FILE NAME

oh3fu30tr6.dat
oh3fu30tr6.fu30
oh3fu30tr6.fu6
oh3fu30tr6.fu14
oh3fu30tr6.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu30 input data file
poly.fu6 long output file
poly.fu14 summary output file
poly.fu15 summary output file

14.C.80. Test run oh3fu31tr1

OH + H₂ → OH₂ + H IVTST-M calculation including a well on the reactant side

This sample test run uses the *unit31* option to perform an IVTST-M calculation including a well on the reactant side. This is an IVTST-M-100/2002 calculation based on an fu31 input file identical to the one written in the oh3tr9 test run. The value for the step-size SINCW is set to 0.001 bohr, which is the step-size used in the generation of the fu31 file.

Note that this run cannot be done without interpolating to the well, since the energies given in the fu31 input file for some of the points on the reactant side are below the energy of the infinitely separated reactants. Since the interpolation to two separated products is based on a Eckart potential, which does not allow the presence of wells, the algorithm would fail. Thus, the well is required in the present IVTST-M calculation. If the information given on the reactant side of the reaction path were to be reduced to the point where no energy lower than the reactant energy appears in the fu31 file, the interpolation could be carried out using the reactant information.

The calculated reaction path is extended up to $s = -1.70 a_0$ (using a scaling mass of 1.802023 a.u.). Since the well is estimated to be located at $s = -1.73 a_0$, extending the range of values of s further toward reactants could cause numerical errors, since the reaction path has not been interpolated beyond the well. In general, whenever a finite range of s is used in a IVTST-M calculation (i.e., reactions with only one reactant or only one product, or reactions with wells), the user should check that the range of s given in the fu5 file does not go beyond the values of s estimated for the stationary point (or points).

POLYRATE I/O FILES

FILE NAME

oh3fu31tr1.dat
oh3fu31tr1.fu31
oh3fu31tr1.fu6
oh3fu31tr1.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu31 input data file
poly.fu6 long output file
poly.fu15 summary output file

14.C.81. Test run oh3fu31tr2

OH + DH \rightarrow HOD + H direct dynamics calculation of an isotopically substituted reaction using electronic structure data for the isotopically unsubstituted reaction read from file fu31, using ESD(H)/RODS, and using IVTST-M.

This sample run demonstrates how to use the OH + H₂ \rightarrow H₂O + H reaction path along with the ESD(H)/RODS [RODS] and IVTST-M algorithms and file fu31 [POTENTIAL = *unit31*] to obtain the conventional TST, CVT, CVT/ZCT, and CVT/SCT rate constants for the isotopically substituted reaction OH + DH \rightarrow HOD + H. The ESD(H)/RODS method is described in Ref. 50 in Sect. 21. The fu31 input file used in this run was written using the WRITEFU31 option in a run on OH + H₂ (not included in the test suite). The KIES option must be included in the GENERAL section of file fu31 so that the correct value of *s* is calculated for the isotopically substituted reaction.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3fu31tr2.dat	poly.fu5	input data file
oh3fu31tr2.fu31	poly.fu31	input data file
oh3fu31tr2.fu6	poly.fu6	long output file
oh3fu31tr2.fu14	poly.fu14	summary output file
oh3fu31tr2.fu15	poly.fu15	summary output file

14.C.82. Test run oh3fu40tr1

$\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$ using electronic structure input file data from file fu40 and curvilinear internal coordinates

This sample run demonstrates the use of electronic structure input file data in the calculation of conventional TST rate constants [$\text{POTENTIAL} = \text{unit40}$] for the reaction $\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$. Hessians are read in mass-scaled coordinates [$\text{HESSFORM} = \text{msbohr}$, $\text{HESSUNIT} = \text{msbohr}$ in file fu40]. Note that even though the reaction path is not required for a conventional TST rate constant calculation, the PATH section must be included in the unit fu5 input in order to pass to the program the value of the factor used in scaling the coordinates [SCALEMASS in the unit fu5 input] and at least two points along the reaction path have to be input from unit fu40. The curvilinear internal coordinates [$\text{COORD} = \text{curv2}$] used for the $\text{O-H} + \text{H}'\text{-H}'' \rightarrow \text{H-O-H}' + \text{H}''$ hydrogen atom transfer reaction include the O-H, H'-H'', and O-H' stretches, the H-O-H' and H''-H'-O bends, and the H''-H'-O-H dihedral [INTDEF]. All normal modes are treated harmonically [HARMONIC in the unit fu5 input].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3fu40tr1.dat	poly.fu5	input data file
oh3fu40tr1.fu40	poly.fu40	input data file
oh3fu40tr1.fu6	poly.fu6	long output file
oh3fu40tr1.fu14	poly.fu14	summary output file
oh3fu40tr1.fu15	poly.fu15	summary output file

14.C.83. Test run oh3fu40tr2

OH + H₂ → H₂O + H using electronic structure input file data from file fu40 and curvilinear internal coordinates

This sample run demonstrates the use of electronic structure input file data in the calculation of rate constants [POTENTIAL = *unit40*] for the reaction OH + H₂ → H₂O + H. This type of calculation requires the unit fu40 input file. The normal mode and generalized normal mode frequencies are obtained at each save point on the MEP by first transforming the Cartesian Hessian matrix into curvilinear internal coordinate space, projecting out the gradient direction, and then diagonalizing the projected Hessian using the Wilson **GF** method. The curvilinear internal coordinates [COORD = *curv2*] used for the O-H + H'-H'' → H-O-H' + H'' hydrogen atom transfer reaction include the O-H, H'-H'', and O-H' stretches, the H-O-H' and H''-H'-O bends, and the H''-H'-O-H dihedral [INTDEF]. The curvature components are computed using a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate [CURVATURE = *compute* and GRADDER = *noextra* in the unit fu40 input]. All normal modes and generalized normal modes are treated harmonically [HARMONIC in the unit fu5 input], and only TST and CVT rate constants are calculated [CVT] at four temperatures [TEMP], using both MEPSAG and CD-SCSAG tunneling corrections [ZCT, SCT].

POLYRATE I/O FILES

FILE NAME

oh3fu40tr2.dat
oh3fu40tr2.fu40
oh3fu40tr2.fu6
oh3fu40tr2.fu14
oh3fu40tr2.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu40 input data file
poly.fu6 long output file
poly.fu14 summary output file
poly.fu15 summary output file

14.C.84. Test run oh3fu40tr3

OH + H₂ → H₂O + H direct dynamics calculation using the electronic structure data read from file fu40, using rectilinear coordinates, and IVTST-M-4/4.

This sample run demonstrates the use of electronic structure input file data read from unit fu40 in the calculation of rate constants [POTENTIAL = *unit40*] for the reaction OH + H₂ → H₂O + H. This type of calculation requires the unit fu40 input file. The interpolation is calculated using IVTST-M-4/4 [MAXLPTS = 500]. Note that the normal mode and generalized normal mode frequencies are obtained by diagonalization of the matrix of the second derivatives of the potential with respect to the mass-scaled Cartesian coordinates, which is read in packed form from unit fu40 [HESSFORM = *packed*, HESSUNIT = *msbohr*], and no anharmonic data are included [ANHARM = *none*]. The curvature components are computed using a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate [CURVATURE = *compute*, GRADDER = *noextra*]. All normal modes are treated harmonically [HARMONIC in the unit fu5 input], and only TST and CVT rate constants are calculated [CVT] at four temperatures [TEMP], using both MEPSAG and CD-SCSAG tunneling corrections [ZCT, SCT].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
oh3fu40tr3.dat	poly.fu5	input data file
oh3fu40tr3.fu40	poly.fu40	input data file
oh3fu40tr3.fu6	poly.fu6	long output file
oh3fu40tr3.fu14	poly.fu14	summary output file
oh3fu40tr3.fu15	poly.fu15	summary output file

14.C.85. Test run oh3fu40tr4

OH + H₂ → H₂O + H using electronic structure input file data from file fu40 for generating the next point along the reaction path (reactants valley).

This sample run demonstrates the use of the NEXTPT keyword for generating the next point along the reaction path in conjunction with the use of electronic structure input from file fu40 [POTENTIAL = *u n i t 4 0*] for the reaction OH + H₂ → H₂O + H. Electronic structure information is included for two points on the reaction path (one on each side of the saddle point), with the first one having a positive *s* and the second one having a negative *s* (thus the points are not in increasing order of *s*, but this is permitted when NEXTPT is on). Since the last point has a negative *s*, IDIRECT is set equal to -1. The next point is generated on the reactants side using a step size of 0.0005 amu^{1/2} bohr [SSTEP].

POLYRATE I/O FILES

FILE NAME

oh3fu40tr4.dat
oh3fu40tr4.fu40
oh3fu40tr4.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu40 input data file
poly.fu6 long output file

14.C.86. Test run oh3fu40tr5

OH + H₂ → H₂O + H using electronic structure input file data from file fu40 for generating the next point along the reaction path (products valley)

This sample run demonstrates the use of the NEXTPT keyword for generating the next point along the reaction path in conjunction with the use of electronic structure input from file fu40 [POTENTIAL = *unit40*] for the reaction OH + H₂ → H₂O + H. Electronic structure information is included for two points on the reaction path (one on each side of the saddle point), with the last one being in the products valley. Since the last point has a positive *s*, IDIRECT is +1. The next point is generated on the products side using a step size of 0.0005 amu^{1/2} bohr [SSTEP].

POLYRATE I/O FILES

FILE NAME

oh3fu40tr5.dat
oh3fu40tr5.fu40
oh3fu40tr5.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu40 input data file
poly.fu6 long output file

14.C.87. Test run oh3fu40tr6

$\text{OH} + \text{H}_2 \rightarrow \text{OH}_2 + \text{H}$ fu40 calculation including a well on the reactant side

The oh3fu40tr6 test run is a sample run that uses file fu40 and includes as the left-hand edge of the interpolation interval a well on the reactant side in the calculation. The reaction path is interpolated by means of third order Lagrangian interpolation using the points along the reaction path including the reactants well.

POLYRATE I/O FILES

FILE NAME

oh3fu40tr6.dat
oh3fu40tr6.fu40
oh3fu40tr6.fu6
oh3fu40tr6.fu14
oh3fu40tr6.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu40 input data file
poly.fu6 long output file
poly.fu14 summary output file
poly.fu15 summary output file

14.C.88. Test run hnitrl

H(a)/Ni(100) → H(a)/Ni(100) with harmonic approximation and ESD integrator

This sample run is for the surface diffusion reaction H(a)/Ni(100) → H(a)/Ni(100), where both reactant and product are adsorbed species [PHASE = *solid*] on a fixed Ni lattice. In this particular calculation none of the Ni atoms are allowed to move. All vibrational modes are treated by the default harmonic approximation. Points along the MEP are calculated by the Euler steepest-descents (ESD) integrator [RPM = *esd*] with a step size of 0.0001 bohr [SSTEP], and a generalized normal mode analysis is performed every 0.02 bohr [INH = 200]. CVT rate constants [CVT] are calculated. In addition, MEPSAG and CD-SCSAG transmission coefficients are included [ZCT, SCT]. The reaction path curvature components are calculated using a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate [CURV = *dgrad*]. Note that the electronic partition functions of the transition state and of the reactant are assumed to cancel.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
hnitrl.dat	poly.fu5	input data file
hnitrl.fu6	poly.fu6	long output file
hnitrl.fu14	poly.fu14	summary output file
hnitrl.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILE

pothni.dat UNIT = 4
 FILE NAME = pothni.dat

14.C.89. Test run h2nitr1

$\text{H}_2(\text{a})/\text{Ni}(100) \rightarrow 2\text{H}(\text{a})/\text{Ni}(100)$ with harmonic approximation and ESD integrator

This sample run is for the unimolecular gas-solid-interface reaction $\text{H}_2(\text{a})/\text{Ni}(100) \rightarrow 2\text{H}(\text{a})/\text{Ni}(100)$ where both reactant and products are adsorbed species [PHASE = *solid*] on a fixed Ni lattice consisting of 817 Ni atoms. All vibrational modes are treated by the default harmonic approximation. Points along the MEP are calculated by the ESD integrator [RPM = *esd*] with a step size of 0.001 bohr [SSTEP], and a generalized normal mode analysis is performed every 0.01 bohr [INH = *10*]. CVT rate constants are calculated [CVT] at 8 temperatures [TEMP]. In addition, MEPSAG, CD-SCSAG, and LCG3 transmission coefficients are included [ZCT, SCT, LCTOPT], and only the ground state is involved in the LCG3 exit arrangement for the LCG3 tunneling calculation [NOALLEXCIT].

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
h2nitr1.dat	poly.fu5	input data file
h2nitr1.fu6	poly.fu6	long output file
h2nitr1.fu14	poly.fu14	summary output file
h2nitr1.fu15	poly.fu15	summary output file

POTENTIAL ENERGY SURFACE DATA FILES

poth2niads.dat	UNIT = 4	FILE NAME = poth2niads.dat
poth2nilat.dat	UNIT = 7	FILE NAME = poth2nilat.dat

14.C.90. Test run ch4ohtr1

OH + CH₄ → H₂O + CH₃ zero-order IVTST calculation

This sample run uses the ICOPT keyword [ICOPT: *zero*] for a zero-order IVTST (*i.e.*, IVTST-0) calculation of the reaction OH + CH₄ → H₂O + CH₃ at 19 temperatures [TEMP]. The classical potential energy curve and the adiabatic ground-state energy curve are approximated by Eckart functions from $s = -2.0$ bohr to $s = 2.0$ bohr [SRANGE]. The Hessian grid spacing is set to 0.02 bohr [INH = 20]. The ab initio data taken from *J. Chem. Phys.* **99**, 1013 (1993) are input by means of the unit fu50 file. The ZCT keyword is used since MEPSAG is the only semiclassical tunneling method available for the zero-order IVTST calculation. The lowest-frequency vibrational mode of the generalized transition state is treated as a hindered rotor [VANHAR, TOR], while all of the other modes are treated by the default harmonic approximation.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch4ohtr1.dat	poly.fu5	input data file
ch4ohic.dat	poly.fu50	input data file
ch4ohtr1.fu6	poly.fu6	long output file
ch4ohtr1.fu14	poly.fu14	summary output file
ch4ohtr1.fu15	poly.fu15	summary output file

14.C.91. Test run ch4ohtr2

OH + CH₄ → H₂O + CH₃ zero-order IVTST calculation

This sample run uses the IVTST0 keyword for a zero-order IVTST (*i.e.*, IVTST-0) calculation of the reaction OH + CH₄ → H₂O + CH₃ at 19 temperatures [TEMP]. The classical potential energy curve and the adiabatic ground-state energy curve are approximated by Eckart functions from $s = -2.0$ bohr to $s = 2.0$ bohr [SRANGE]. The Hessian grid spacing is set to 0.02 bohr [INH = 20]. The ab initio data taken from *J. Chem. Phys.* **99**, 1013 (1993) are input by means of the unit fu29 file. The MEPSAG method [ZCT] is the only semiclassical tunneling method available for the zero-order IVTST calculation. All vibrational modes are treated by the default harmonic approximation.

POLYRATE I/O FILES

FILE NAME

ch4ohtr2.dat
ch4ohtr2.fu29
ch4ohtr1.fu6

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu29 input data file
poly.fu6 output file

14.C.92. Test run ch4cltr1

$\text{CH}_4 + \text{Cl} \rightarrow \text{HCl} + \text{CH}_3$ zero-order IVTST calculation

This sample run uses the IVTST0 keyword for a zero-order IVTST (*i.e.*, IVTST-0) calculation of the reaction $\text{CH}_4 + \text{Cl} \rightarrow \text{HCl} + \text{CH}_3$ at 4 temperatures [TEMP]. The classical potential energy curve and the adiabatic ground-state energy curve are approximated by Eckart functions from $s = -1.0$ bohr to $s = 1.5$ bohr [SRANGE]. The Hessian grid spacing is set to 0.02 bohr [INH = 4]. The *ab initio* data are input by means of the unit fu29 file. The MEPSAG method is the only semiclassical tunneling method available for the zero-order IVTST calculation. All vibrational modes are treated by the default harmonic approximation. The $^2\text{P}_{3/2}$ and $^3\text{P}_{1/2}$ states of Cl are used in the calculation of the electronic partition functions [ELEC]. [This test run reproduces the results of one of the test cases in Ref. 26 in Sect. 21.]

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch4cltr1.dat	poly.fu5	input data file
ch4cltr1.fu29	poly.fu29	input data file
ch4cltr1.fu6	poly.fu6	output file

14.C.93. Test run ch4cltr2

$\text{CH}_4 + \text{Cl} \rightarrow \text{HCl} + \text{CH}_3$ first-order IVTST calculation

This sample run uses the IVTST1 keyword for a first-order IVTST (*i.e.*, IVTST-1) calculation of the reaction $\text{CH}_4 + \text{Cl} \rightarrow \text{HCl} + \text{CH}_3$ at 4 temperatures [TEMP]. The classical potential energy curve and the adiabatic ground-state energy curve are approximated by Eckart functions from $s = -1.0$ bohr to $s = 1.5$ bohr [SRANGE]. The Hessian grid spacing is set to 0.02 bohr [INH = 4]. The *ab initio* data are input by means of the unit fu29 file. Both MEPSAG and SCSAG semiclassical tunneling methods [ZCT, SCT] are used, since they are available for the first-order IVTST calculation. All vibrational modes are treated by the default harmonic approximation. The $^2\text{P}_{3/2}$ and $^3\text{P}_{1/2}$ states of Cl are used in the calculation of the electronic partition functions[ELEC]. [This test run reproduces the results of one of the test cases in Ref. 26 in Sect. 21.]

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ch4cltr2.dat	poly.fu5	input data file
ch4cltr2.fu29	poly.fu29	input data file
ch4cltr2.fu6	poly.fu6	output file

14.C.94. Test run ch4otr1

CH₄ + O → CH₃ + OH CUS calculation using harmonic redundant curvilinear coordinates

This sample run calculates canonical unified statistical model (CUS) rate constants including MEPSAG, CD-SCSAG, LCG3, COMT, and μOMT tunneling, based on the information from an analytical potential energy surface. The reaction has two different electronic states, but since they are very close in energy, they are assumed to be represented by the same surface. Thus, the total rate constant should be multiplied by a factor of 2 in order to take into account this electronic degeneracy. This is done by using a factor of 6 for the saddle point electronic degeneracy instead of its value of 3.

The $V_a^G(s)$ curve has two maxima near the saddle point. In order to estimate the recrossing effects due to these maxima a CUS calculation is performed.

POLYRATE I/O FILES

FILE NAME

ch4otr1.dat
ch4otr1.fu6
ch4otr1.fu14
ch4otr1.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu6 long output file
poly.fu14 summary output file
poly.fu15 summary output file

POTENTIAL ENERGY SURFACE DATA FILE

potch4o.dat

UNIT = 4

FILE NAME = potch4o.dat

14.C.95. Test run ohcltr1

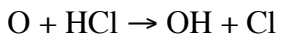
O + HCl → OH + Cl ICVT calculation using Morse 1 anharmonicity and internal coordinates

This sample run calculates improved canonical variational theory (ICVT) rate constants including MEPSAG and CD-SCSAG tunneling, based on the information from an analytical potential energy surface. The saddle point and generalized transition states are nonlinear [SPECIES = *nonlints*]. Internal coordinates are defined [INTDEF keyword] as the OH (1-2) and HCl (2-3) bond lengths and OHCl (1-2-3) bond angle. Vibrational modes are treated by the Morse I approximation [MORSE, MORMODEL = *morsei*], with a dissociation energy [DEMIN] of 0.16968172 E_h.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ohcltr1.dat	poly.fu5	input data file
ohcltr1.fu6	poly.fu6	long output file
ohcltr1.fu14	poly.fu14	summary output file
ohcltr1.fu15	poly.fu15	summary output file

14.C.96. Test run ohcltr2



ICVT calculation using WKB-on-the-true-potential for ground-state energies and Morse 1 anharmonicity for excited state energies.

This sample run is the same as ohcltr1, except that the ground-state vibrational energy levels are calculated using the WKB method on the true potential. All excited state energy levels are computed using the Morse I approximation [MORSE, MORMODEL = *morsei*], with a dissociation energy [DEMIN] of 0.16968172 E_h .

POLYRATE I/O FILES

FILE NAME

ohcltr2.dat
ohcltr2.fu6
ohcltr2.fu14
ohcltr2.fu15

POLYRATE FILE NAME

poly.fu5 input data file
poly.fu6 long output file
poly.fu14 summary output file
poly.fu15 summary output file

14.C.97. Test run ohcltr3

O + HCl($v=1$) \rightarrow OH + Cl Adiabatic excited-state calculation using WKB-on-the-true-potential for the $v=1$ stretch energy and for the ground-state bend energy and Morse 1 anharmonicity for excited statebend energies.

This sample run is the same as ohcltr2, except that the reaction is for excited-state stretch vibration. The stretch vibration is treated as adiabatic along the full reaction path [STATE adiab] with quantum number $v=1$ specified using the STATEOPT keyword. The excited-state energy level for the stretch vibration and the ground-state vibrational energy level for the bend vibration are calculated using the WKB method on the true potential. Bend excited state energy levels are computed using the Morse I approximation [MORSE, MORMODEL = *morsei*], with a dissociation energy [DEMIN] of $0.16968172 E_h$. The ICVT calculation is limited to $-1.5 a_0 < s < 1.0 a_0$.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ohcltr3.dat	poly.fu5	input data file
ohcltr3.fu6	poly.fu6	long output file
ohcltr3.fu14	poly.fu14	summary output file
ohcltr3.fu15	poly.fu15	summary output file

14.C.98. Test run ohcltr4

O + HCl($v=1$) \rightarrow OH + Cl Partial-reaction-path (PRP) adiabatic excited-state calculation using WKB-on-the-true-potential for stretch energy and for the ground-state bend energy and Morse 1 anharmonicity for excited state bend energies.

This sample run is the same as ohcltr3, except that stretch vibration is treated as adiabatic in the $v=1$ state along the reaction path only up to $s = -0.22 a_0$ and then in the ground state for $s > 0.22 a_0$. This is specified using the STATE diab keyword and specifying transition state mode 2 in the first excited state before the diabatic switch and in the ground excited state after the diabatic switch. Thermal treatment of the bend vibration before and after the switch is explicitly indicated in the input. The energy level for the stretch vibration and the ground-state vibrational energy level for the bend vibration are calculated using the WKB method on the true potential. Bend excited state energy levels are computed using the Morse I approximation [MORSE, MORMODEL = *morsei*], with a dissociation energy [DEMIN] of $0.16968172 E_h$. The ICVT calculation is limited to $-1.5 a_0 < s < 1.0 a_0$.

POLYRATE I/O FILES

FILE NAME	POLYRATE FILE NAME	
ohcltr4.dat	poly.fu5	input data file
ohcltr4.fu6	poly.fu6	long output file
ohcltr4.fu14	poly.fu14	summary output file
ohcltr4.fu15	poly.fu15	summary output file

14.D. Summary of test runs

This section contains a summary of the test runs which make up the POLYRATE test suite.

Analytic potential

ch5j1tr1	ch5j1tr2	ch5j1tr3	ch5j1tr4	ch5j1tr5	ch5j1tr6
ch5j1tr7	ch5j1tr8	ch5j1tr9	ch5j1tr10	ch5j1tr11	
ch5j2tr1	ch5j2tr2	ch5j2tr3	clhbrtr1	clhbrtr2	clhbrtr3
clhbrtr4	cmctr1	cwmctr1	cwmctr2	cwmctr3	h3tr1
h3tr2	h3tr3	h3tr4	ho2tr1	ho2tr2	ho2tr3
ho2tr4	oh3tr1	oh3tr2	oh3tr3	oh3tr4	oh3tr5
oh3tr6	oh3tr7	oh3tr8	oh3tr9	oh3tr10	hnitr1
h2nitr1	ch4otr1	cmctr3	ch5fu51tr1	nh3tr2	ohcltr1
ohcltr2	ohcltr3	ohcltr4			

fu30

ch5fu30tr1	ch5fu30tr2	ch5fu30tr3	ch5fu30tr4	ch5fu30tr5	ch5fu30tr6
oh3fu30tr1	oh3fu30tr2	oh3fu30tr3	oh3fu30tr4	oh3fu30tr5	oh3fu30tr6

fu31

ch5fu31tr1	oh3fu31tr2
ohfu31tr1	

fu40

ch5fu40tr1	ch5fu40tr2	ch5fu40tr3	ch5fu40tr4	ch5fu40tr5	ch5fu40tr6
oh3fu40tr1	oh3fu40tr2	oh3fu40tr3	oh3fu40tr4	oh3fu40tr5	oh3fu40tr6

IVTST-0 with fu29

ch5j2itr1	ch5fu29tr1	ch5fu29tr2	ch4ohtr2	ch4cltr1
-----------	------------	------------	----------	----------

IVTST-0 with fu50

ch5fu50tr1	ch4ohtr1
------------	----------

IVTST-1 with fu29

ch5j2itr2	ch4cltr2
-----------	----------

VTST-IOC with analytic potential and fu50

ch5fu50tr2 ch5fu50tr3 cmctr2 ho2tr5 nh3tr1

VTST-IOC with fu30, fu31, or fu40 and fu50

ch5icfu30tr1 ch5icfu30tr2 ch5icfu31tr1 ch5icfu40tr1 ch5icfu40tr2

VTST-ISPE with fu30, fu31, or fu40 and fu51

ch5fu51tr1 cmctr3

15. DESCRIPTION OF FILES IN RELEASE OF VERSION 9.4.1

The source code is partitioned into twenty-four FORTRAN files, seven INCLUDE files, and twelve date-time subroutines. Note that a full POLYRATE source consists of everything in the twenty-four FORTRAN files, common.inc, percon.inc, a param.inc file (such as one of the param[1, 2, 3, 4].inc files), and a dattim routine and possibly dateclock, as described in Section 13. A description of each POLYRATE subprogram is given in Subsection 15.A.1. Subsection 15.A.2 briefly describes the INCLUDE files and the subprograms SETUP and SURF, which must be supplied by the user for runs that do not use any of the potentials provided for the test run suite. (Dummy versions that can be used for runs with $POTENTIAL = unit30, unit31, \text{ or } unit40$ are in dumpot.f.) The input and output files for the sample runs are explained in Subsection 14.C. The Unix C shell scripts are explained in Subsection 13.A and Section 16. The file polydoc84 is this manual for the current version. A list of all the files in the distribution package can be found in Subsection 13.A.

The files polymq.f and polysz.f contain subprogram MXLNEQ for solving and inverting sets of simultaneous linear equations and EISPACK subroutines for determining matrix eigenvalues and eigenvectors. In the distributed version the machine precision variable MACHEP is set equal to 1.0×10^{38} in the MXLNEQ subprogram and to 2^{-46} in the EISPACK subprograms. We found these settings to be adequate for all machines on which we tested the code.

If you received the POLYRATE distribution in a tar format, then the files will automatically be distributed into the default POLYRATE directories as described in Section 13, except that this manual may be in a separate file (not tarred with everything else).

15.A. FORTRAN files and subprograms, INCLUDE files, and PES subroutines

Subsection 15.A.1 contains an alphabetical list and brief descriptions of all the subprograms that make up the POLYRATE program, Subsection 15.A.2 describes the INCLUDE files and the user-supplied potential energy subprograms, and Subsection 15.A.3 describes the FORTRAN files.

15.A.1. POLYRATE subprograms

In many cases a more detailed description and some relevant references are given as comments at the beginning of the FORTRAN code for each subprogram. Note that subroutines may be divided into two groups, POLYRATE-proper routines and hooks routines. Routines that start with a Y or have HOOK in the name are always part of the hooks.

ABSORD

Order the lowest eigenvalues and eigenvectors according to absolute values.

ACALC

Dummy routine in place of the corresponding ACESRATE routine.

ACESET

Dummy routine in place of the corresponding ACESRATE routine.

AITKEN

Performs a Lagrangian interpolation of the effective reduced mass from the reaction-path Hessian grid.

AITKF2

Checks the Aitken interpolation for "bad" values, i.e., interpolants that do not lie within the original range of the data.

AITKNF

Performs Aitken interpolation.

ALFCT

Computes the logarithmic values of factorials of even integers used in the calculation of the Bernoulli numbers.

ANCOEF

Computes the anharmonicity coefficients x_{ij} and the constant term (E_0) from dimensionless normal coordinate force constants evaluated previously by subprogram GRADDR, and then calculates the vibrational ground-state energy and fundamentals from the anharmonicity coefficients and the constant term.

ANGL

Function that calculates the angle i-j-k.

ANGLCS

Function that calculates the sine and cosine of the valence angle between two difference Cartesian vectors.

ANGLV

Calculates the angle between two vectors.

ANHARM
 Computes anharmonicities for bound normal modes. The anharmonicity option is selected by the anharmonicity keywords.

ANHDER
 Computes the 3rd and 4th derivatives along the generalized normal modes.

ARMUEF
 Calculates the argument of the exponential in equation 19 of Ref IVTST-M for the interpolation of the effective mass for CD-SCSAG tunneling.

ARMUEF1
 Calculates μ^{CDSC} for the extrapolated points in IVTST-M with the hooks.

ARRSM
 Calculates the moment of inertia for the points along the reaction path given in fu31 and reorders them according to the value of s .

ARRSV
 Reorders the value of $V_{\text{MEP}}(s)$ given as input in fu31 according to s .

ASYMALP
 Calculates the direction cosines for an asymmetric gyrator.

ASYMCAL
 Prints the moment of inertia of a group of asymmetric gyrators.

ASYMENT
 Calculates the moment of inertia of an asymmetric gyrator.

ASYMNZD
 Sets up the calculation of moment of inertia of an asymmetric gyrator.

ASYMRMI
 Calculates the moment of inertia of an asymmetric gyrator.

BANGLE1
 Calculates the Wilson B-matrix ($\partial R/\partial X$) for bending.

BCALC
 Calculates the curvature components and the total curvature for the small-curvature semiclassical adiabatic tunneling calculations.

BCALCP
 Calculates Page-McIver curvature factors B_{mF} .

BCALC0
 Calculates the small-curvature effective mass at the saddle point by linear interpolation.

BTORSN
 Calculates the Wilson B-matrix ($\partial R/\partial X$) for a torsion.

- BMAT**
Calculates the Wilson B-matrix, $\partial R/\partial X$, where R are the curvilinear internal coordinate vectors and X are the Cartesian coordinate vectors.
- BMAT1**
Forms the Wilson B-matrix row corresponding to a linear stretch coordinate between atom j and atom i . The input is in the form of difference Cartesians.
- BMAT2**
Forms the rows of the Wilson B-matrix corresponding to a bent triatomic.
- BMAT5**
Forms the Wilson B-matrix rows corresponding to degenerate linear bends. The input is in the form of two difference Cartesians defining the linear triatomic. The definitions of the bends are those of Hoy et al. (*Mol. Phys.* **24**, 1265 (1972)).
- BMAT5X**
Forms the Wilson B-matrix rows corresponding to degenerate linear bends. The input is in the form of two difference Cartesians defining the linear triatomic by first determines a transformation matrix to rotate the coordinate system such that the molecule lies along the Z axis.
- BIMAT**
Calculates the generalized inverse of the Wilson B-matrix.
- BTEN1**
Forms the Wilson B-matrix 2nd derivative elements for a bond-length curvilinear internal coordinate in terms of its first derivatives and magnitudes of difference Cartesians.
- BTEN2**
Computes the bent triatomic 2nd derivative Wilson B-matrix elements.
- BTEN5**
Defines the 2nd derivative Wilson B-matrix elements for the degenerate linear bend in terms of the difference Cartesian vectors that define its bonds.
- BTEN5X**
Defines the 2nd derivative Wilson B-matrix elements for the degenerate linear bend in terms of the difference Cartesian vectors that define its bonds so that the molecular frame has the linear bend along the Z axis.
- BTENR**
Transforms the Wilson B-matrix elements from difference Cartesians to Cartesians.
- BTENS**
Calculates the tensor $(X/R \cdot R^2/X^2 \cdot \partial X/\partial R)$
- BOLTZ**
Performs the Boltzmann average of the tunneling probability by a numerical integration with respect to E .

BRENT
Implements Brent's line minimization method. See Ref. 14 in Sect. 20.

BRNULI
Computes Bernoulli numbers used in the calculation of the WKB phase integral for a double-well potential.

CALCS
Computes the minimum distance between two geometries along the reaction path measured as a straight line between them.

CALCUB
Calculates the coefficients for the cubic polynomial used in the IVTST-M interpolation of $V_{\text{MEP}}(s)$ for unimolecular reactions.

CASE
Function that takes a string of 80 characters and converts the upper case letters in the string to lower case letters.

CENTER
Translates the center of mass of the system to the origin. Also calculates the determinant of the moment of inertia tensor, which is equal to the product of the three principal moments of inertia, since they are the eigenvalues of this matrix.

CENTRI
Determines the relative index assignments for a linear triatomic system (i-j-k). Atom i is placed in the center and the j-k assignments are made arbitrarily.

CENTRAL
Determines which atom is the central atom for a bent triatomic by using bond lengths, and calculates the bond angle.

CFLOAT
Function that takes a character string that contains a floating point number and converts it to a floating point number as defined by standard FORTRAN conventions.

CHKFRE
Checks if any of the vibrational frequencies are zero or imaginary.

CHKUP
Retrieves 3rd-order dimensionless normal coordinate force constants needed for calculating the vibrational ground-state energy and fundamentals for simple perturbation theory calculations.

CLASS
Finds the recursion relation for various classes of orthogonal polynomials used to find the nodes in Gaussian-type quadratures.

CMS
Calculates the center of mass of a subset of the system.

COG
Defines a cut-off gaussian correction function.

COHBT1
Defines a cut-off hyperbolic tangent correction function.

COHBT2
Defines a cut-off hyperbolic tangent correction function.

COLSHF
Performs shifting of Hessian grid data in the storage arrays.

CONOUT
Prints out the curvilinear internal coordinates.

CONV40
Converts the geometries, gradients, and Hessians to atomic units.

CORTRM
Calculates Coriolis constants.

COSINE
Modified cosine function.

CRVVEC
Computes the curvature vector when unit40 input is used.

CUBST
Evaluates quantities necessary for the cubic-starting algorithm.

CUBIC
Finds the real root of $ax^3 + bx^2 + cx + d = 0$ closest to a given x value.

CUBIC2
Finds all roots (real and complex) of $ax^3 + bx^2 + cx + d = 0$.

CUSSPL
Spline fits the $\Delta G^{GT,0}(s)$ curve and finds the maxima and minima along the curve for a CUS calculation

CVCOOR
Driver for the curvilinear internal coordinate bound-state analysis of non-linear polyatomics.

DAXPY
BLAS routine calculates $c\mathbf{x} + \mathbf{y}$.

DATTIM
Calls system-dependent subroutines DATE and TIME and writes the current date and time.

DEFAULT
Sets the defaults for all the parameters and variables used in POLYRATE. The defaults for each different section of input is set in a separate subprogram for clarity.

DEFENG
Sets the defaults for keywords used in the ENERGETICS section.

DEFGEN
Sets the defaults for keywords used in the GENERAL section.

DEFLT40
Determines if the needed fu40 keywords were read for the IOPth species. Default values are assigned when necessary.

DEFOPT
Sets the defaults for keywords used in the OPTIMIZATION section.

DEFPAT
Sets the defaults for keywords used in the PATH section.

DEFRAT
Sets the defaults for keywords used in the RATE section.

DEFSEC
Sets the defaults for keywords used in the SECOND section.

DEFSTA
Sets the defaults for keywords used in the all the stationary point sections.

DEFTUN
Sets the defaults for keywords used in the TUNNEL section.

DERIV2
Calculates the Hessian matrix by using a quadratic fit to the potential energy surface. Only the potential is used, not its first derivatives.

DERV
Calculates the curvature vector related to third derivatives for IVTST.

DERV24
Computes the second derivatives of the potential by fitting the analytic first derivatives to a fourth-order polynomial around the desired location and then taking the derivative of this polynomial.

DERVAG
Evaluates the first derivative of an n -dimensional point by the two-point central difference method.

DFGN31
Set defaults for a fu31 file.

DGAMMA
Evaluates the gamma function of order $n \leq 3$.

DGEDI
Computes the determinant and inverse of a matrix using the factors computed by DGEFA.

DGEFA
Factors a double precision matrix by Gaussian elimination.

DIATOM
Computes the equilibrium properties for a diatomic reactant or product treated as a Morse oscillator.

DIGISTR
Function that transform a three digits integer into a string with 3 characters.

DIST
Function that calculates an interatomic distance.

DPMPAR
A service routine for HYBRD1 that computes machine precision.

DOGLEG
A service routine for HYBRD1 that determines the convex combination x of the Gauss-Newton and scaled gradient directions that minimizes $(ax - b)$ in the least-squares sense, subject to the restriction that the Euclidean norm of dx be at most delta.

DOPNM
Calculates the steepest-descent path and the generalized normal modes along the MEP.

DOREPR
Calculates the reactant and product properties.

DOREST
Prepares a restart calculation (i.e., reads data from unit fu1, etc.).

DORODS
Subroutine used to setup a RODS or VRP(ESD) calculation.

DORPH
Prepares the potential energy routines and sets up the MEP interpolation information for an electronic structure input file calculation.

DOSAFR
Calculates the saddle point frequencies.

DOSAGE
Calculates the saddle point geometry.

DDOT
BLAS routine computes the dot product of two vectors.

DOTPRD
Computes the dot product between two vectors.

DPR123
Computes the dot product between two vectors. First vector points from atom 2 to atom 1; second vector points from atom 2 to atom 3.

DPSWAP
Swaps two double precision numbers.

DSCAL
BLAS routine scales a vector by a constant.

DSWAP
BLAS routine interchanges two vectors.

EBND
Computes the energy of a level of a quadratic-quartic potential by the perturbation-variation method of Truhlar.

ECKARS
Interpolates the classical potential curve (IOP = 1) or the ground-state vibrationally adiabatic energy curve (IOP = 2).

ECKART
Defines an Eckart correction function.

ECKRT
Interpolates frequencies with an Eckart or a hyperbolic tangent function.

EHOOK
The driver for the energy routines in the modular version of POLYRATE.

EFFBATH
Computes the solvent contribution for the nonequilibrium solvation approximation.

EFMAIN
The driver for the eigenvector following algorithm.

EFPHES
Prints the Hessian matrix in eigenvector following algorithm.

EFORMD
Forms geometry step by eigenvector followint algorithm.

EFOVLP
Determines the overlap of the geometry steps in the eigenvector following algorithm.

ELLIP
Computes elliptic integrals of the first and second kinds.

ELRPH
Evaluates generalized normal mode frequencies along the MEP when electronic-structure input is used [POTENTIAL = *unit30* or *unit40*].

ENDRODS
Subroutine used to finish a RODS or VRP(ESD) calculation.

ENDSLP
TSPACK subroutine that estimates the first derivative at the edges of the spline-fit interval to be used as the two extra conditions of the global spline under tension.

ENORM
Calculates the norm of a vector \mathbf{x} .

ENROUT
Writes reaction energetics information to FORTRAN unit fu6.

EPARTI
Function that computes the electronic partition function for IVTST calculations.

EPART
Computes the electronic partition function.

ERF
Calculates the error function.

EVIB
Calculates vibrational energy levels.

EWKB
Uses a Chebyshev quadrature to compute the primitive WKB approximation to the energy of a vibrational level of a quadratic-quartic potential.

EXPND
Expands triangular matrix \mathbf{A} to square matrix \mathbf{B} .

EXTRAP
Provides simple exponential extrapolations to the asymptotic values of quantities used in tunneling calculations.

EXTRP40
Sets the values for extrapolation when using unit fu40 input.

EXTRP
Fits a straight line to two points and carries out interpolations or extrapolations.

F1DI
Evaluates the function VALVAG for LINMNR.

F1DIM
Computes the one-dimensional energy for use in line minimization subroutines.

FACORD

Orders the force constants scaling factors according to the actual sequence of the curvilinear internal coordinate when FCSCALE is used.

FCINPT

Closes all the input files used in READ5.

FCMEP

Closes all the input and output files used in the MEP calculation.

FCN

Calculates an Eckart or a hyperbolic tangent function to be passed into the non-linear equation solver.

FCRATE

Closes all the input and output files used in the rate constant and the tunneling calculations.

FCSCL

Scales diagonal elements of the force constant matrix in curvilinear internal coordinates by corresponding scaling factors when FCSCALE is used.

FDIAG

Diagonalizes the force constant matrix and, optionally, prints out generalized normal mode information.

FDIAG2

Diagonalizes the Cartesian force constant matrix for Hessians from unit 29 input and, optionally, prints out generalized normal mode information.

FDJAC1

A service routine for HYBRD1 that computes a forward-difference approximation to the $n \times n$ Jacobian matrix associated with a specified problem of n functions in n variables. If the Jacobian has a banded form, then function evaluations are saved by only approximating the nonzero terms.

FICLSE

Closes a file given the status with which to close the file and the FORTRAN unit number to which the file is linked.

FINDL

Calculates the value for the RPL keyword in a VTST-IC input file (fu50) from a lower-level calculation.

FINDLV

Calculate the range parameter for an Eckart function.

FINOUT

Computes and prints out the final set of rate constants. Also obtains activation energies and bottleneck properties and analyzes the ratio of transition state theory results to those of canonical variational transition state theory.

FIOPEN
 Opens all files needed for a given calculation except for the main input and the main output files (i.e., the files linked to FORTRAN units fu5 and fu6).

FITMAX
 Finds the maximum in the adiabatic potential curve and the generalized free energy of activation curve using 3- and 5-point fits.

FITMX
 Finds the maximum of function f with a 5-point fit.

FIVFT
 Computes a quartic fit through five points for IVTST.

FIVPT
 Computes a quartic fit of $F = ax^4 + bx^3 + cx^2 + dx + e$ to five points.

FORMA
 Forms the generalized inverse of the B-matrix by the Pulay prescription.

FORMF
 Form the product $\mathbf{A}^T \mathbf{F} \mathbf{A}$, where \mathbf{A} is the A-matrix and \mathbf{F} is the Cartesian force constant matrix.

FORMF2
 Form the second term of curvilinear internal force constant matrix.

FORMG
 Form the G matrix.

FORMBT
 Forms one xyz-band of the transformation matrix from Cartesian coordinates to difference Cartesians.

FRPRMN
 Conjugate gradients minimization subroutine.

FRSORT
 Sorts a set of frequencies into ascending or descending order.

GAUSQD
 Calculates the weights and abscissa for N -point Gauss-Legendre quadrature.

GAUSSQ
 Computes the nodes and weights for Gaussian-type quadrature.

GBSLVE
 For a symmetric tridiagonal $n \times n$ matrix \mathbf{A} , this subroutine finds the component n of the n -dimensional vector \mathbf{v} such that $\mathbf{A} \mathbf{v} = \mathbf{c}$, where \mathbf{c} is a vector of all zeros except for 1 in the n -th position.

GBTQL2
Finds the eigenvalues and the first components of the eigenvectors of an asymmetric tridiagonal matrix by the implicit QL method.

GEMM
Performs fast matrix multiplication. From the BLAS library.

GENSEC40
Reads and processes the GEN40 section of the unit fu40 input.

GETFRQ
Obtains vibrational frequencies in cm^{-1} and generalized eigenvectors.

GFDIAG
This routine effectively diagonalizes the unsymmetric **GF**-matrix, formed by product of **G** and **F**. The method is given by Macintosh and Michaelian (*Can. J. Spectrosc.* **24**, 65 (1979)). Only symmetric matrices are diagonalized here. the **GF** product is never explicitly formed.

GFDIA2
This routine effectively diagonalizes the unsymmetric, rectangular **GF** matrix by generalized inverse **G** matrix method. The method is given in P. Pulay,; G. Fogarasi, *J. Chem. Phys.* **1992**, 96, 2856.

GHOOK
The driver for the first derivative routines in the modular version of POLYRATE.

GIVTST
Driver for zero- and first-order interpolated VTST (the global approach) as described in *J. Chem. Phys.* **95**, 8875 (1991).

GRAD
Evaluates derivatives along normal mode (or generalized normal mode) directions.

GRADDR
Calculates 3rd- and 4th-order dimensionless normal coordinate force constants.

GSETA
Dummy routine in place of the corresponding ACESRATE routine.

GSETM
Geometry optimization calculations needed when POLYRATE is used with MOPAC.

GSETP
Geometry optimization calculations needed when POLYRATE is used with an analytic surface.

HBT
Calculates the hyperbolic tangent at s_{MEP} .

HEADR
Prints out the program header.

HHOOK
The driver for the second derivative routines in the modular version of POLYRATE.

HINDRT
Calculates the moment of inertia for internal rotation.

HINDRT1
Calculates the moment of inertia for internal rotation of an asymmetric gyrotor.

HQSC
Semiclassical solution of a harmonic-quartic potential.

HRPART
Calculates the partition functions using the hindered-internal-rotator approximation.

HRSET
Setup calculation of hinder rotor partition function.

HVAL
TSPACK subroutine that evaluates a Hermite interpolatory tension spline at a point.

HYBRD
A service routine for HYBRD1.

HYBRD1
Finds a zero of a system of n nonlinear functions in n variables by a modification of the Powell hybrid method.

ICFDRP
Driver of the curvilinear internal coordinate bound-state analysis for reactants and products.

ICFDIAG
Driver for the curvilinear internal coordinate bound-state analysis of triatomics.

ICINT
Function that takes a character string which is really an integer and converts it to an integer. Error checking that is not possible using ichar is done.

ICODAL
Function that translates a set of alphanumeric characters into an integer code.

ICOUTRP
Prints out the curvilinear internal coordinates for reactants and products.

ICSAVE
Copies frequencies and eigenvectors for bound modes from curvilinear internal coordinate work spaces to the locations used by Cartesian coordinate procedures.

ICSVRP
Copies the bound-state analysis results from the curvilinear internal coordinate work spaces to the locations used by Cartesian coordinate procedures.

IDAMAX
BLAS routine finds the index of element having maximum absolute value.

IMOM40
Computes the moment of inertia.

INTCOR
Defines the curvilinear internal coordinates for reactants and products by splitting the global connectivities into subsets.

INTERSC
Finds out the intersection of two integer arrays.

INTF31
Subroutine that reorders the information on stationary points taken from fu31 and puts it on the correspondent arrays.

INTRVL
TSPACK function that locates the interval in the input data for a spline fit that contains a certain point.

INVRT
Inverts a one-dimensional array of size N .

INVRT2
Inverts a two-dimensional array of size N_1, N_2 .

INVRT3
Inverts a three-dimensional array of size N_1, N_2, N_3 .

INERTA
Calculates the moment of inertia for a linear molecule or a diatomic, or the product of the moments of inertia for a non-linear molecule in au.

INITZE
Initializes some of the variables and arrays.

INTAB
Writes out a table of all the input parameters.

INTABS
Writes out a table of all the input parameters associated with the stationary point sections.

INTEGR
Integration driver for the PATH subroutine.

INTEUL
Performs Euler steepest-descents integration.

INTFNC
Calculates and returns the negative of the normalized gradient for the path integration.

INTPM
Page-McIver reaction path algorithm..

INTPM2
Computes $\partial s / \partial t$ for the Page-McIver method.

INTRPL
Calls POLFIT to do polynomial interpolation for a given quantity.

IRCX
Intrinsic-reaction-coordinate pathfinder, using the method of Schmidt, Gordon, and Dupuis.

IRFTG0
Computes the information necessary for dealing with the frequencies, Hessians, and curvature vectors for unit fu40 input.

IRPHZ
Does interpolations for the zero-order IVTST calculations.

IRPH
Does interpolations for the first-order IVTST calculations.

ISWAP
Swaps two integers.

IVTMA
Interpolates the exponent for calculating μ^{CDSC} by IVTST-M through the hooks.

IVTMD
Calculates the extrapolated points with IVTST-M through the hooks.

IVTMH
Sets up the parameters required for IVTST-M through the hooks.

IVTST0
Calculates the classical (i.e., Born-Oppenheimer) potential energy curve and the adiabatic ground-state potential energy curve along the reaction path in a zero-order VTST-IOC calculation.

JACSCF
Computes all eigenvalues and eigenvectors by the Jacobi method.

KAPA

Calculates transmission coefficients for IVTST calculations.

KAPPAS
Calculates the transmission coefficients using the uniform semiclassical WKB method (MEPSAG) and the small curvature method (SCSAG).

KAPVA
Calculates the tunneling corrections for a set of temperatures.

KG1
Nodes and weights for Kronrod quadrature.

LCG34
Calculates the tunneling probabilities $P^{\text{LCG34}}(E)$ by the LCG34 method.

LCGIT
Builds the grid for the 1D spline under tension.

LCGNAD
Defines the limit between adiabatic and nonadiabatic regions for the LCT3 and LCT4 methods.

LCGRD
Builds a 2D rectangular grid from selected points of the PES.

LCGSRT
Sorts the uniformized LCG3 or LCG4 probabilities.

LCGTHE
Calculates the primitive tunneling amplitude $T_{\text{tun}}(\tilde{s}_0, n_p)$ by integration over the straight-line LCG3 or LCG4 tunneling path using both Gauss-Legendre quadrature and the trapezoidal rule.

LCGTR
Transforms a point of the PES to a point of the 2D rectangular grid.

LCNORM
Calculates the generalized normal mode coordinates corresponding to the scaled displacements along the LCG3 or LCG4 tunneling path. These coordinates are needed for eq. (14) of Ref. 20 in Sect. 21.

LCNWTN
Performs a Newton root search to determine the s value corresponding to a point along the LCG3 or LCG4 tunneling path, using eq. (22) of Ref. 20 in Sect. 21.

LCPATH
Calculates the mass-scaled Cartesian coordinates of the LCG3 or LCG4 tunneling path as given in eq. (8) of Ref. 20 in Sect. 21.

LCPROB

Calculates the uniformized LCG3 probabilities from the primitive values $P_{prim}^{LCG3}(E)$ using eq. (30) of Ref. 20 in Sect. 21.

LCPROJ

Calculates the projections of the generalized normal mode coordinates onto the LCG3 path using eq. (14) of Ref. 20 in Sect. 21.

LCSET

Sets up the LCG3 tunneling path and calculates the angle between the MEP and the tunneling path [eq. (12) of Ref. 20 in Sect. 21] along with the vibrational period of eq. (17) of Ref. 20 in Sect. 21.

LCSINT

Performs the numerical integration over tunneling termini along the reaction path as given by eq. (9) of Ref. 20 in Sect. 21.

LCSTX

Sets up the tunneling path for excited final states and determines the highest excited state involved in LCG3 calculations.

LCTP

Determines the ending terminus of the tunneling path for a given excited state of the LCG3 exit arrangement and a given initiating terminus \tilde{x}_0 using the resonance condition.

LCTV4

Searches for a LCT4 nonadiabatic region within a LCG3 adiabatic region.

LCVCOR

Calculates the effective potential for the nonadiabatic region in an LCG3 calculation as given in eq. (28) of Ref. 20 in Sect. 21.

LCVIB

Calculates the vibrational zero-point energies of eq. (5) of Ref. 20 in Sect. 21.

LCVMOD

Calculates the zero-point energy and potential for a given normal mode in an LCG3 calculation.

LCXINT

Calculates the integrand in the phase integral along the adiabatic segments of an LCG3 tunneling path.

LCXTAU

Calculates the vibrational period, τ , for LCG3 calculations as given by eqs. (17) and (18) of Ref. 20 of Sect. 21.

LCZCRT

Determines which region (adiabatic or nonadiabatic) a given point on an LCG3 tunneling path is in by using eqs. (23) and (24) of Ref. 20 in Sect. 21.

LIN
Solves a set of simultaneous linear equations.

LIN2
Solves two simultaneous linear equations in the matrix form $\mathbf{Ax} = \mathbf{b}$.

LININT
Does the linear interpolation.

LINMNR
Finds the minimum along a given direction.

LINMN
One-dimensional minimization search.

LOCATE
Determines the indices on the grid of reaction path coordinates corresponding to a point on the LCG3 tunneling path determined by LCNWTN.

LOCS
Called by AITKF2 to locate the s value about which to center the interpolation grid. The grid is centered about the closest point on the Hessian grid.

LOPT40
Translates the GEN40 input keywords to LOPT values.

LSAME
Service routine for GEMM.

LUDCMP
Performs an LU decomposition using an algorithm from W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes* (Cambridge University Press, Cambridge, 1986), p. 31.

LUBKSB
Service routine for LUDCMP.

MAIN
Driver for POLYRATE. The main input and output files (i.e., those files linked to FORTRAN units fu5 and fu6) are opened in this subprogram.

MATINV
Performs matrix inversion using LU decomposition routines.

MATX
A general-purpose matrix multiplication routine.

MEPINV
Stores the information saved as functions of the reaction coordinate in order of increasing s by inverting the order of storage.

MEPOUT
Prints MEP information to the long output file, i.e., to unit fu6.

MEPSRT
Does a bubble sort of the MEP information in order of increasing s .

MNBRAK
Finds an interval which contains a minimum.

MOPEG
Dummy routine to disable the keyword MOPAC in POLYRATE.

MOPHES
Dummy routine to disable the keyword MOPAC in POLYRATE.

MOPOPT
Dummy routine to disable the keyword MOPAC in POLYRATE.

MOPSET
Dummy routine to disable the keyword MOPAC in POLYRATE.

MTINV2
Performs matrix inversion using singular value decomposition routines

MUCDSC
Computes the centrifugal-dominant small-curvature effective mass.

MUSC
Computes the SCSAG effective mass.

MUSCCD
Computes the CD-SCSAG effective mass.

MXLNEQ
Solves a set of simultaneous linear equations in the matrix form $\mathbf{Ax} = \mathbf{b}$, and evaluates the determinant of \mathbf{A} .

NEARWC
computes the nearest reactant or product endpoint used in frequency interpolation.

NEWT
Does a Newton-Raphson search for a stationary point.

NEXTPT
Computes the next point on the MEP.

NORMOD
Computes generalized normal mode frequencies and directions.

NOROUT
Prints out generalized normal mode information.

NOST31
Processes the information about non-stationary points along the reaction path.

NSTATE

Calculates the total number of accessible states in LCG3 calculations that include contributions from tunneling into excited states.

NUMFRQ

Calculates the number of internal vibrations.

OBTSRT

Compares two elements and stores the bigger one.

OHOOK

The driver for the geometry optimization routines in the modular version of POLYRATE.

OPENFI

Opens a file given the unit number, the status, and the file name.

OPT50

Checks for any conflicts in the options for the VTST-IOC calculations.

OPT50C

Checks for consistency between the VTST-IOC options chosen in the unit fu50 input.

OPTION

Checks for consistency between the options chosen in the unit fu5 input and writes out information about the options selected.

ORDS31

Reorders the information stored on the array save31 containing all the information given as an input in fu31.

OVRLP

Computes overlaps between old and new normal mode eigenvectors and warns if a crossing has occurred.

PATH

Calculates the reaction path.

PHID

Computes the phi function needed for uniform semiclassical quantization of a double-well potential.

PHSINT

Computes the phase integral for the WKB method.

POLINT

Uses Neville's algorithm to interpolate a given quantity.

POLYAT
Calculates the equilibrium properties of a linear or a nonlinear polyatomic reactant or product.

PRCORD
Prints the bond lengths, bond angles, and dihedral angle along the reaction path.

PREGPT
Spline fits s , V , and I^{-1} for IVTST-M calculations carried out through the hooks.

PREP
The driver for all the preparation routines in the modular version of POLYRATE.

PREPJ
The driver for all the preparation routines needed by REACT1, REACT2, PROD1, PROD2, START, and PATH. See Sect. 8.C.

PRMEP
Prints extra MEP information to units fu25-fu27.

PRNTBT
Prints the B-matrix stored in lower triangular form in clusters of 5 rows.

PRNTFC
Prints a lower triangular matrix (in clusters of 5 rows).

PROJTI
Forms the projection operator and calculates the projected Cartesian force constant matrix for IVTST calculations.

PROJCT
Calculates the projected force constant matrix and normalized gradient.

PROJF
Forms the projection operator and projects out the gradient from the curvilinear internal force constant matrix.

PRSQ
Prints out a square matrix.

PSAG
Locates points (translational turning points) where the energy is equal to the adiabatic potential and then computes the imaginary-action integral.

PSATP
Finds turning points on the adiabatic barrier.

PSATP2
Finds turning points on the adiabatic barrier.

PSATX
Finds turning points on the adiabatic barrier.

PSATX2

Finds turning points on the adiabatic barrier.

PTORS

Function that calculates the torsion angle i-j-k-l.

PTQVIB

Calculates vibrational partition functions by simple perturbation theory using the ground-state energy and fundamentals. [See Ref. 23 in Sect. 21 for the equations without Coriolis terms in E_0 and see Ref. 35 in Sect. 20 for the Coriolis contributions to E_0 .]

PTSEC40

Reads and processes the POINT40 sections in the unit fu40 input.

PYC1

TSPACK subroutine that employs a three-point quadratic interpolation method to compute local derivative estimates associated with a set of data points. The derivatives are chosen so that the Hermite interpolatory tension spline fulfils the user selected end conditions.

PYC1P

TSPACK subroutine that employs a three-point quadratic interpolation method to compute local derivative estimates associated with a set of data points. The derivatives are chosen so that the Hermite interpolatory tension spline fulfils the defaulted end conditions.

PYC2

TSPACK subroutine that solves a linear system for a set of first derivatives associated with a Hermite interpolatory tension spline. The derivatives are chosen so that the Hermite interpolatory tension spline fulfils the user selected end conditions.

PYC2P

TSPACK subroutine that solves a linear system for a set of first derivatives associated with a Hermite interpolatory tension spline. The derivatives are chosen so that the Hermite interpolatory tension spline fulfils the defaulted end conditions.

PYCOEF

TSPACK subroutine that computes the coefficients of the derivatives in the symmetric diagonally dominant tridiagonal system associated with the C-2 derivative estimation procedure for a Hermite interpolatory tension spline.

QFORM
Computes the **QR** factorization.

QQPOT
Fits the quadratic-quartic potential to two points along the generalized normal coordinate using two given distances.

QRFAC
This is a service routine for **HYBRD1** that uses Householder transformations with column pivoting (optional) to compute a QR factorization of the $m \times n$ matrix **A**.

QRSENE
Calculates the quantized reactant eigenstates for tunneling.

QSLVE
Computes a quadratic fit to the function values at two points and the derivative at one point, and uses the fit to perform an interpolation.

QTQVIB
Calculates the quantized reactant-state partition functions.

QUADFT
Computes a quadratic fit to three points.

QUADR
Solves a quadratic equation.

QUADTW
For a function $y(x) = A + B(x-C)^2$, finds B and C from the given information.

RBASIS
Dummy routine in place of the corresponding **ACESRATE** routine.

RBATH
Reads the keyword options for the **BATH** keyword in **PATH** section.

R1UPDT
A service routine for **HYBRD1**.

R1MPYQ
A service routine for **HYBRD1**.

RACES
Reads the **ACES** namelist options.

RALL31
Reads all the data from **fu31** and store it in the **save31** array.

RATE
Computes the generalized free energy of activation curves and rate constants.

RANALY
Reads the temperatures at which a detailed analysis is desired.

RATOMS
Reads the atomic symbols and optionally the masses.

RCNST
Reads the coordinates that will be held constant in any optimization of a stationary point geometry.

RD40EXTR
Reads the EXTRAP40 section of the file fu40 input.

RD40GEN
Reads the GEN40 section of the file fu40 input.

RD40RP
Reads the reactants and products sections of the file fu40 input.

RDETIL
Reads the LCG3 output details information.

RDIATM
Reads the diatomic options for stationary points.

RDIFFU
Reads the diffusion coefficients or coupling constants for the nonequilibrium solvation calculation.

RDHESS
Reads a Hessian matrix from file fu40.

RDLN
Reads a line from an input file without conversion.

REACT
Computes the geometry and properties of a reactant or product.

READ5
Reads input from the file linked to FORTRAN unit fu5.

READIC
Reads the VTST-IOC correction data from the file linked to FORTRAN unit fu50.

READINT
Reads INTDEF keyword information to get the connectivities.

READLN
Reads a line from an input file and converts all characters to uppercase.

REARG1
Rearranges a set of frequencies to an order that is input by the user.

RECS31
Recalculates all the values of s given in fu31 as distances along straight lines between two consecutive geometries.

REDGEO
Determines the geometry of the reactant, product, or saddle point, i.e., linear or nonlinear, and the number of degrees of freedom.

REDMU
Evaluates the effective mass for a given value of s for IVTST as described in *J. Chem. Phys.* **95**, 8875(1991).

REFOPT
Reads the options for eigenvector following algorithm.

RELEC
Reads the electronic degeneracies for stationary points.

RENERG
Reads the ENERGETICS section of the file fu5 input.

REPFL
Replaces the low-frequency mode along the path when the IVTST0FREQ keyword is used.

RES1
Reads the variables for the Euler-with-stabilization (ES1) integrator.

RESTOR
Reads information from the file linked to FORTRAN unit fu1 for a restart rate constant calculation.

RESTRT
Writes information to the file linked to FORTRAN unit fu1 for a restart rate constant calculation.

REXTRP
Reads the EXTRAP section of the file fu5 input.

REZERO
Reads the input for the zero of energy.

REZERU
Reads the unit for the zero of energy.

RFCFAC
Reads FCSCALE keyword information to get the force constant scaling factors when FCSCALE is used.

RFREQ
Reads frequencies from unit fu29 for IVTST calculations.

RGEM31

Reads the options for geometry input in fu31.

RGEN
Reads general information for IVTST calculations from unit fu29.

RGEN31
Reads the options in the general section of an fu31 input file.

RGENER
Reads the GENERAL section of the file fu5 input.

RGEOM
Reads the initial geometry for reactants or products.

RGRD
Read gradient from unit fu29 for IVTST calculations.

RGRD31
Reads the options for gradients input in fu31.

RGSAD
Reads the initial geometry for the saddle point or starting geometry.

RGSPEC
Reads the range of s for which the generalized free energy curve is to be computed.

RGSZM
Reads the initial geometry for the saddle point or starting geometry using z-matrix form.

RGTEMP
Reads the range of s values for each temperature for which the generalized free energy curve is to be computed.

RGZMAT
Reads the initial geometry for the saddle point or starting geometry using z-matrix form. This is used only when the geometry optimization is done using ACES.

RHESS
Reads Hessians from unit fu29 for IVTST calculations.

RHES31
Reads a Hessian matrix from fu31.

RHESSC
Reads HESSCAL options from unit fu5.

RHFSEC
Checks whether frequencies or Hessians are to be read.

RHSS31
Reads the options for Hessian input in fu31.

RICOPT
Reads the VTST-IOC options.

RLCT
Reads the information for large-curvature tunneling calculations.

RLINE
Utility program for reading a line of input and parsing off extra spaces and comments.

RLOWFR
Reads the information given by the LOWFREQ keyword.

RMUVT
Reads μ VT options.

ROPT
Reads the OPTIMIZATION section of the file fu5 input.

ROPT31
Reads the general section of an incomplete fu31 file and sets the options for writing it.

ROPTOP
Reads the options for the OPTIMIN and OPTTS keywords.

RPART
Computes the classical rotational partition function.

RPARTI
Computes the classical rotational partition function for IVTST calculations.

RPATH
Reads the PATH section of the file fu5 input.

RPH31
Sets up the defaults and routes for reading an fu31 file.

RPH40
Sets up the electronic structure information from file fu40.

RPHAIT
Performs NINT-point Lagrangian interpolation of an asymptotic function used in the electronic structure input file option (i.e., the RPH formulation).

RPHB
Calculates the curvature component of mode k .

RPHDXN
Converts from unweighted to mass-scaled gradient and normalizes the gradient vector.

RPHEXP
Evaluates asymptotic exponential functional forms.

RPHFIT
Evaluates parameters of the asymptotic exponential fits.

RPHFT2
Computes 2-point fits to 3-parameter asymptotic exponential functions with the range parameter fixed.

RPHFT3
Computes 3-point fits to 3-parameter asymptotic exponential functions.

RPHG1
Function used in the root search for the range parameter in an asymptotic functional form.

RPHG2
Function used in the root search for the range parameter in an asymptotic functional form.

RPHG5
Function used in the root search for the range parameter in an asymptotic functional form.

RPHINT
Interpolates RPH information.

RPHMEF
Computes the effective tunneling mass for CD-SCSAG from the interpolated value of the exponent in equation 19 in ref. IVTST-M.

RPNT31
Reads the information from each point in the fu31 file.

RPHNWT
Performs a Newton search for the zero of one of the functions $G(C)$ defined in Subsection 9.D.1.

RPHRD1
Reads electronic structure input information and sets up tables used for interpolation.

RPHRD2
Reads electronic structure input information.

RPHSET
Sets up electronic structure input interpolation information.

RPHTRF
Transforms the matrix of second derivatives with respect to unweighted coordinates to second derivatives with respect to mass-scaled coordinates.

RPHTRX
Converts mass-scaled to Cartesian coordinates or Cartesian to mass-scaled coordinates.

RPHWRT
Writes reaction path Hamiltonian interpolation information.

RPOTET
Reads the potential type.

RPOTGE
Reads the potential type to be used for optimizing the geometry.

RPRFL
Replaces the low-frequency mode of the stationary points when the IVTST0FREQ keyword is used.

RPRMEP
Reads the parameters for printing the extra MEP information.

RPRVIB
Reads the range of s between which extra vibrational mode information and Q_{vib} are printed.

RPSEC40
Reads and processes a reactant or product section of the file fu40 input.

RQCOM
Calculates the distance between the centers of mass of two reacting species.

RQRST
Reads the information for the quantized-reactant-state tunneling calculation.

RQUAD
Reads the quadrature information for the tunneling calculations.

RRANGE
Reads the range of s values for which the reaction path is to be computed.

RRATE
Reads the RATE section of the file fu5 input.

RRPM
Reads the value of the RPM keyword of the file fu5.

RRSTRT
Reads the restart options.

RSCT
Reads the options for small curvature tunneling.

RSECND
Reads SECOND section of the file fu5 input.

RSFRST
Reads the variables for a special step size for the first n steps.

RSP
EISPACK diagonalization routine. Finds the eigenvalues and eigenvectors of a real symmetric packed matrix.

RSPDRV
Accepts unpacked arrays the way EISPACK subroutine RS does, packs them for RSPP, and calls RSPP to compute eigenvalues and eigenvectors.

RSPECL
Reads special values of s at which a normal mode analysis is to be done.

RSPP
Diagonalizes a real symmetric matrix in packed form.

RSSTOP
Reads the options of the SPECSTOP keyword.

RST
Diagonalizes a real symmetric matrix using RSPP.

RSTAT
Reads all the stationary point sections of the file fu5 input.

RSTATE
Reads the options used with a state-selected rate constant calculation.

RTEMP
Reads the temperatures at which the rate constant is to be calculated.

RTITLE
Reads the title.

RTOR
Reads the TOR keyword options.

RTOROPT
Reads the TOROPT keyword options.

RTPJAC
Prints the MEP and large curvature representative tunneling termini in mass-scaled Jacobian coordinates.

RTUNNL
Reads the tunneling parameters.

RUNM31
Reads the options for the IVTST-M interpolation on a unimolecular side (or side with a well) on the reaction path.

RVAR6
Reads keyword values in a restart calculation.

RVARY
Reads the variable anharmonicity options desired for reactants, products and or the saddle point.

RVPERT
Reads the options for using simple perturbation theory for the vibrational partition functions.

RVRANG
Reads the anharmonicity options desired for the reaction path for specific regions of the reaction coordinate *s*.

RWKB
Reads the WKB anharmonicity options.

RWORD
Parses a character string to find the next word.

RZVAR
Reads the z-matrix variables for the geometry optimization of the reactants, products and the saddle point.

SADDLE
Computes the saddle point geometry (with subroutine NEWT) and other saddle point properties.

SADENG
Computes the saddle point energetics.

SADSEC40
Reads and processes the SADDLE40 section of the file fu40 input.

SECCEN
Calculates second derivatives by the method of central differences.

SECCEP
Calculates second derivatives by the method of central differences; only computes part of the Hessian matrix.

SENOUT
Writes saddle point energetics information to unit fu6.

SETANH
Sets variables associated with the variable anharmonicity option.

SETLGS
Converts the variables and flags set by evaluating the keywords to the corresponding LGS flags.

SETUP

Called once to initialize the potential, as discussed in Sect. 8.C.1. Ordinarily, this is supplied by the user. The distribution, however, contains a dummy version for use as described in Section 8.C, as well as sample versions for the test runs.

SETVAR

Converts the variables and flags set by evaluating the keywords to the corresponding POLYRATE variables.

SGND3V

Stores the sign of the third derivative in an array for use in computing the Morse turning point.

SHFTSAD

Shifts saddle point information when using unit fu40 input.

SIGS

TSPACK subroutine that determines the value of the tension factors for a spline under tension fit.

SIMPSN

Calculates the phase integral by using Simpson's rule in the quantized-reactant-state tunneling calculation.

SINE

Modified sine function.

SNHCSH

TSPACK subroutine that computes approximate values for modified hyperbolic functions.

SORTG

Sorts frequencies in descending order.

SORT31

Sorts an nx2 array according to the values on the first column.

SPL1B1

Converts the spline fit data supplied by SPL1D1 to four arrays containing the coefficients for local cubic fits.

SPL1B2

Computes the spline interpolation using the cubic polynomial coefficients supplied by SPL1B1.

SPL1D1

Computes the second derivative array for spline interpolations.

SPL2D

Calls the 2D spline under tension.

- SPL31**
Subroutine that interfaces POLYRATE and TSPACK. This subroutine is used when the information is given by means of fu31, which allows for a different number of data for each property.
- SPLNB**
Calculates the value of the curvature components at any point along the reaction path by using spline under tension fitting.
- SPLNF**
Calculates the value of the frequencies at any point along the reaction path by using spline under tension fitting.
- SPLNM**
Calculates the value of the moment of inertia at any point along the reaction path by using spline under tension fitting.
- SPLNMF**
Calculates the value of the effective mass for CD-SCSAG tunneling at any point along the reaction path by using spline under tension fitting.
- SPLNV**
Calculates the value of the energy along the reaction path, $V_{\text{MEP}}(s)$ at any point along the reaction path by using spline under tension fitting.
- SPLM31**
Calculates the value of the moment of inertia at any point along the reaction path by using spline under tension fitting. This subroutine is used when the input is given in fu31.
- SPLRMI**
Calculates the value of the moment of inertia at any point along the reaction path by using spline under tension fitting.
- SPLSPV**
Spline fits ΔV in the ISPE method. This subroutine is used with fu51.
- SPLV31**
Calculates the value of the energy along the reaction path, $V_{\text{MEP}}(s)$, at any point along the reaction path by using spline under tension fitting. This subroutine is used when the input is given in fu31.
- SSAVE**
Saves the reaction path results calculated in subroutine PATH.
- SSX**
Computes the distance between two Cartesian geometries as a straight line in mass-scaled coordinates.
- STAT31**
Calculates stationary point properties from the information read in fu5 and fu31.

STAUV
Calculates the correction term used in computing the improved generalized transition-state partition function from the generalized transition-state partition function.

STLGSJ
Converts the stationary-point-dependent variables and flags set by evaluating the keywords to the corresponding LGS flags.

STOR
TSPACK subroutine that determines floating point number characteristics.

STVARJ
Converts the stationary-point-dependent variables and flags set by evaluating the keywords to the corresponding POLYRATE variables.

SURF
Evaluates the potential, as discussed in Subsection 8.C.1. Ordinarily this is supplied by the user, but the POLYRATE distribution provides a dummy version for use as discussed in Section 8.C, as well as sample versions for the test runs.

SVBKSB
This routine, taken from Numerical Recipes, is for the backward substitution of SV decomposition.

SVBCMP
This routine, taken from Numerical Recipes, is for the singular value decomposition procedure.

TABLE
Writes a summary of selected forward rate constants to the file linked to FORTRAN unit fu15.

TAUV
Calculates the cutoff partition function for improved canonical variational theory.

THETA2
Computes the phase integral for the semiclassical eigenvalues of a single-well potential.

THETA3
Computes the phase integral for the semiclassical eigenvalues of a double-well potential.

TITLE
Reads and writes the title of the job.

TP
Locates the turning points at a given energy for a normal mode vibration.

TPCDSC
Calculates the centrifugal-dominant small-curvature turning point.

TQL2
 Computes the eigenvalues and eigenvectors of a symmetric tridiagonal matrix by the QL method [H. Bowdler, R. S. Martin, C. H. Reinsch, and J. H. Wilkinson, *Num. Math* **11** (1968) 293].

TQLRAT
 Finds the eigenvalues of a symmetric tridiagonal matrix by the rational QL method [C. H. Reinsch, *Comm. ACM* **16** (1973) 689].

TRANLF
 Converts Cartesian to mass-weighted coordinates or mass-weighted to Cartesian coordinates store in lower triangular form.

TRANS
 Converts Cartesian to mass-scaled coordinates or mass-scaled to Cartesian coordinates. Also converts derivatives of the potential energy function between these coordinate systems.

TRANS2
 Converts Cartesian to mass-scaled coordinates (IOP = 1) or mass-scaled to Cartesian coordinates (IOP = 2) used in IVTST calculations.

TRANG
 Transforms gradient in Cartesians to curvilinear internal coordinates.

TRANFC
 Transforms Hessian in Cartesians to curvilinear internal coordinates.

TRANSF
 Converts Cartesian to mass-weighted coordinates (IOP = 1) or mass-weighted to Cartesian coordinates (IOP = 2).

TRBAK3
 Forms the eigenvectors of a real symmetric matrix by back transformation of the eigenvectors of the similar symmetric tridiagonal matrix.

TRED3
 Reduces a real symmetric matrix to a symmetric tridiagonal matrix.

TREPT
 Computes a parabolic fit, $F = ax^2 + bx + c$, to three points.

TRLFRP
 Stores the force constant matrix in lower triangle form.

TSC
 Locates points where $E = V_a^{AG}$ and calculates the theta integral.

TSOUT
 Computes and writes the final set of conventional transition state theory rate constants. Also obtains activation energies and equilibrium constants.

TSPSI

TSPACK subroutine that computes a set of parameter values which define a Hermite interpolatory tension spline.

TSRATE

Calculates the free energy of activation at the saddle point, individual reactant and product partition functions, and conventional transition state theory rate constants in the harmonic approximation. This routine is only used when the NOCVT keyword is used in the RATE section. Some further discussion is provided in Section 6.L of this manual.

TURNPT

Computes the concave-side turning point for the zero-point level in the calculation of the curvature component.

UPCSE

Converts a line of characters to uppercase.

UPCAS

Converts lower case characters to upper case in the given string.

UPDHR

Updates the arrays FMIHR and FMIHTS in the VTST-IOC calculations.

UPDMI

Updates the arrays FMOM and FMITS in the VTST-IOC calculations.

VALVAG

Calculates the value of V_a^G for a given projected direction, given V, X, DX, and F.

VECCON

Normalizes the normal mode vectors and converts them to mass-scaled Cartesians.

VECMAG

Obtains the magnitudes of a set of vectors.

VICLCG

Calculates the classical energy corrections in the non-adiabatic region for the LCT method in the VTST-IOC calculations.

VIVT

Computes the Eckart potential and its derivatives.

VPART

Computes the vibrational partition function.

VPARTI

Computes the vibrational partition function using the harmonic approximation.

VSPLIN
Generates spline fits to the vibrationally adiabatic potential curve and the effective reduced mass.

VSPLI2
Generates spline fits to the vibrationally adiabatic potential curve and the effective reduced mass for IVTST calculations.

VTMUSN
Calculates the microcanonical variational sum of states in the calculation of the microcanonical variational rate constant.

WTITLE
Writes the title of the run.

WKBENE
Calculates the energy eigenstates with the WKB approximation.

WKBPOT
Computes the energy levels of a quadratic-quartic function using a semiclassical method with parabolic connection formulae.

WKBVIB
Driver for the calculation of WKB vibrational energies based on the true (unfitted) potential for a given mode.

WRTHOK
Writes stationary point information to file fu61 and, if necessary, to file fu62.

WRTWEL
Writes the information about the wells to file fu6.

WVAR6
Writes the keyword values for a restart calculation.

XDOTP
Calculates the dot product of two vectors.

XERBLA
Error handler for the BLAS routines.

XPROD
Forms the cross product of vectors x and y, placing the result in vector z. All vectors are of dimension three.

YDER24
Computes the second derivatives of the potential by fitting the analytic first derivatives to a fourth-order polynomial around the desired location and then taking the derivative of this polynomial. This is the hooks version of DERV24.

YDERV2
Calculates the Hessian matrix by using a quadratic fit to the potential energy surface. Only the potential is used, not its first derivatives. This is the hooks version of DERIV2.

YNEWT
Does a Newton-Raphson search for a stationary point. This is the hooks version of NEWT.

YSECEN
Calculates second derivatives by the method of central differences. This is the hooks version of SECCEN.

YSECEP
Calculates second derivatives by the method of central differences; only computes part of the Hessian matrix. This is the hooks version of SECCEP.

YTRANS
Converts Cartesian coordinates to mass-scaled coordinates or converts mass-scaled coordinates to Cartesian coordinates. Also converts derivatives of the potential energy function between these coordinate systems. This is the hooks version of TRANS.

ZEROPT
Computes the zero-point energy.

ZERP
Function that computes the zero-point energy.

ZIGAMA
Calculates the incomplete Gamma function of order 3/2.

ZOC3P
Prepares the correction information at 3 points for the VTST-IOC calculations.

ZOCAST
Checks the accuracy of the interpolation in the VTST-IOC calculations.

ZOCFRE
Calculates the corrections for the frequencies for VTST-IOC calculations.

ZOCPAR
Calculates the parameters for the VTST-IOC calculations.

ZOCPRN
Prints some of the information involved in VTST-IOC calculations.

ZOCUPD
Updates the arrays WER, FMOM, CDSCMU, VCLAS, WETS, and VADIB and the variables EPRD and WSTAR with zero-order corrections for VTST-IOC calculations.

ZOCVCL
Calculates the correction for the energy VMEP in VTST-IOC calculations.

ZRIDDR
Solves a nonlinear equation for calculating the range parameter of an Eckart potential.

ZROUT

Explicitly initializes the work space for matrix with zeros to overcome the FORTRAN save effect.

ZRPTE

Evaluates the zero-point energies.

ZSPMEP

Sets up information for ISPE. This subroutine is used with unit fu51.

15.A.2 User-supplied subprograms and INCLUDE files

INCLUDE files:

common.inc

This INCLUDE file contains all the common blocks used for passing information between the various subprograms.

param.inc

This INCLUDE file contains all the parameters which may be modified; these parameters affect the memory requirements of the POLYRATE code and the FORTRAN I/O unit numbers for the input and output requirements of the code (see Section 5.A.2).

percon.inc

This INCLUDE file contains all the permanent constants for the POLYRATE code. These parameters should not be changed.

User-supplied subprograms:

SETUP

Subprogram supplied by the user that computes any necessary constants for the potential (see Section 8 and examples in the test suite).

SURF

Subprogram supplied by the user that computes the potential and gradient for a set of Cartesian coordinates (see Section 8 and examples in the test suite).

15.A.3 FORTRAN files

acespoly.f

This file contains dummy subroutines used with ACESRATE.

energetics.f
This file contains subprograms for the calculations of properties of the PES. These include subprograms that handle optimizations and numerical second derivatives.

ef.f
This file contains the Eigenvector-Following algorithm for optimizing geometries of the stationary points.

fromblas.f
This file contains all the subroutines from the BLAS math library.

givtst.f
This file contains subprograms for the global version of IVTST calculations.

hdr.f
This file contains subprograms that print out the program header.

hooks.f
This file contains subprograms for interfacing electronic structure calculations with POLYRATE.

intbsv1.f
This file contains subprograms for the transformations of the Cartesian Hessians into a curvilinear coordinate system at non-stationary points along the MEP and for the curvilinear coordinate vibrational analysis for linear and nonlinear triatomics.

intbsv2.f
This file contains subprograms for the transformations of the Cartesian Hessians into a curvilinear coordinate system at non-stationary points along the MEP and for the curvilinear coordinate vibrational analysis for nonlinear polyatomics. The coordinate transformation of the Hessian is based on the transformation ANHTRAX program written by Matt Challacombe and Jerzy Cioslowski.

intbsv3.f
This file contains subprograms for the transformations of Cartesian Hessians into curvilinear coordinate system at reactants and products and for the curvilinear coordinate vibrational analysis for nonlinear polyatomic.

interface.f
This file contains subprograms for parsing POLYRATE keyword input.

ivtstm.f
This file contains subprograms for performing IVTST-M interpolation.

main.f
This file contains the driver for polyatomic VTST calculations.

mopoly.f
This file contains dummy subroutines used with MORATE.

polyag.f This file contains subprograms with names beginning with letters A to G.

polyhl.f This file contains subprograms with names beginning with letters H to L.

polymq.f This file contains subprograms with names beginning with letters M to Q.

polyrr.f This file contains subprograms with names beginning with the letter R.

polysz.f This file contains subprograms with names beginning with letters S to Z.

poly31.f This file contains subprograms that are used when `POTENTIAL = unit31`.

poly40.f This file contains subprograms that are used when `POTENTIAL = unit40`.

rtpjac.f This file contains subprograms that are used to plot the MEP and large curvature representative tunneling termini in mass-scaled Jacobian coordinates.

15.B. Utility program detmi.f

In this version of POLYRATE, one of the VTST-IOC options [DETMI keyword] allows the user to correct the calculated determinants of moment of inertia tensors with accurate data. However, the values of determinants of moment of inertia tensors are not always available in the literature or in the electronic structure calculation output of standard codes. So, in this version of the POLYRATE distribution we provide a small utility program, detmi.f, in the util subdirectory of the polyrate9.4.1 directory. This program reads a molecular geometry in Cartesians (in bohrs or in Angstroms) and the atomic weights (in a.m.u.) of the constituent atoms of the molecule, and calculates the determinant of the moment of inertia tensor of the molecule in all the units supported by the POLYRATE unit fu50 input options [UNITMI keyword]. (See subsection 11.F) The input formats are very simple, and a detailed explanation can be found in the file detmi.doc in the same subdirectory. A job control file detmi.jc and three sample input files, ch4.mi, ch4oh.mi, and oh.mi are also provided.

One can use the compiling and linking commands listed in the Section 13 of this manual to make an executable file `detmi.exe` for any machine. This program is easy to use; however, one has to remember to do the following:

1. Make a data file. The program will read the data from a file called `detmi.dat`. If your data file has a different name, copy, move, or link your file to this name.
2. When finished, the program will generate an output file called `detmi.out`. It is recommended that you move (or copy) this file to another file name before doing any other calculations. Otherwise the new output will overwrite the old `detmi.out` file, and the old information in that file will be lost.
3. There is a simple script `detmi.jc` in the same sub directory. It can do the above file copying steps for you if you want. It takes the input file name as the only argument.

For example, if your data file is called `ch4.mi`, type

```
detmi.jc ch4.mi <Return>
```

to do the calculation. The output will be written to the file `detmi.out` and copied to the file `ch4.out`.

How to make an input data file:

1. In the first line you have to input three numbers in free format.

The first number is the number of atoms in the molecule.

The second number specifies if the geometry of the molecule is linear or nonlinear. Set the number to 1 if it is linear. Set the number to 0 if the geometry is not linear.

The third number specifies the input units for the Cartesians. Set it to 1 if the units you use are Angstroms. Set the number to 0 if you use bohrs.

For example, if the target molecule is methane and your geometry data in Cartesians are in Angstroms, the first line should look like the following:

```
5 0 1
```


2. The second and the third lines are the calculation titles. You can write whatever you want here (less than 80 characters in one line).

For example:

```
Methane: CH4
QCISD/adj-cc-pVTZ geometry
```

3. The next n lines, where n is the first number you entered in the first line, contains the atomic masses and Cartesian coordinate information. There should be 4 numbers on each line. The first number is the atomic mass in a.m.u. of the atom; the second, third, and fourth numbers are the x , y , and z coordinates of the atom.

For example:

```
1.007825    0.0000    0.0000    0.0000
12.00000    1.0848    0.0000    0.0000
1.007825    1.4464    1.0228    0.0000
1.007825    1.4464   -0.5114    0.8857
1.007825    1.4464   -0.5114   -0.8857
```

15.C. Utility program hrotor.f

When some of the vibrational modes of the reactants, products, or GTS are treated as hindered rotations, POLYRATE allows the user to input more accurate values of the moments of inertia of these modes in a VTST-IOC calculation [IC and HRMI keywords]. However, usually the user can only get the geometry of a molecule and the vibrational eigenvectors of these modes from an electronic structure calculation output.

The utility program hrotor.f is designed to take the atomic masses and geometry of the molecule and the vibrational eigenvector of a hindered-rotor mode of the molecule and then to calculate the moment of inertia. It can also calculate the harmonic, free rotor, and hindered rotor partition functions at different temperatures. Detailed explanations can be found in the file hrotor.doc in the polyrate9.4.1/util directory and in Refs. 24, 29, and 30 in Sect. 21. A sample input file, cwmc.rmi, is also provided in the polyrate9.4.1/util directory.

You can use the compiling and linking commands listed in the Section 13 of this POLYRATE manual to make an executable file hrotor.exe for your machine. The program is designed to read data from Unix standard input and write data to Unix standard output. To use the program, type

```
hrotor.exe < input_file_name > output_file_name <Return>
```

For example:

```
hrotor.exe < cwmc.rmi > cwmc.rmi.out <Return>
```

To make a data file follow the steps below:
(Note: All the input data are in free format.)

1. In the first line you have to input 4 numbers.

The first number is the number of atoms in the molecule.

The second number specifies the units of the Cartesian coordinates you are going to input. Set it to 0 if you use bohrs, to 1 if you use Angstroms.

The third number specifies the coordinate system of the vibrational eigenvector you are going to input. Set it to 0 if you use a mass-scaled coordinate system; set it to 1 if you use a non-mass-scaled Cartesian coordinate system.

The fourth number specifies the number of temperatures at which you want to calculate the partition function.

For example, if the target molecule is the saddle point of the reaction in the POLYRATE cwmcctrl test run, your geometry data in Cartesians are in bohrs, the eigenvector is in mass-scaled coordinates, and you want to calculate the partition function at two temperatures, the first line should look like the following:

```
9 0 0 2
```

2. The second and the third lines are the calculation titles. You can write whatever you want here (less than 80 characters in one line).

For example:

```
Saddle point of the reaction in cwmc test run  
The moment of inertia
```

3. If you set the third number in the first line to 1, skip this step. The next line is the scaling mass in a.m.u. of the mass-scaled coordinate system which the input eigenvector is based on.

For example:

```
1.0000
```

4. The next n lines, where n is the first number you entered in the first line, contain the atomic masses of the constituent atoms in a.m.u.

For example:

```
12.00000
34.96885
1.007825
1.007825
1.007825
34.96885
1.007825
1.007825
15.99491
```

5. The next n lines, where n is the first number you entered in the first line, contain the geometry data of the molecule in (unscaled) Cartesian coordinates. There should be 3 numbers on each line, namely the x , y , and z coordinates of the atom.

For example:

```
2.0187 -2.3695 0.1399
1.7442 -2.3695 4.4447
4.0466 -2.3695 0.1399
1.0334 -4.1335 0.1399
1.0334 -0.6054 0.1399
1.7442 -2.3695 -4.1650
-2.3564 -2.3695 1.5703
-2.3564 -2.3695 -1.2906
-3.4890 -2.3695 0.1399
```

6. The next n lines, where n is the first number you entered in the first line, contain the vibrational eigenvector of this hindered-rotor mode. There should be 3 numbers on each line, the x , y , and z components of the eigenvector for the atom corresponding to this line.

For example:

```
0.000 0.567 0.000
0.000 -0.260 0.000
0.000 0.533 0.000
0.308 -0.008 0.000
-0.308 -0.008 0.000
0.000 -0.260 0.000
0.000 -0.116 0.000
0.000 -0.116 0.000
0.000 0.207 0.000
```

7. The next line contains the symmetry and subgroup information. The first number is the symmetry factor of this hindered rotation. The second number is the number of atoms in one of the subgroups. The next m numbers, where m is the second number in this line, are the atom indices of those atoms in that subgroup.

For example:

```
3 4 1 3 4 5
```

8. If you decide to calculate the partition function (i.e., if the fourth number in the first line is greater than 0) you need to input the harmonic frequency in cm^{-1} in a new line and the temperatures in Kelvin in the last line.

For example:

```
163.4794  
600.00 1000.00
```

15.D. Utility programs **findl.f** and **findb.f**

In VTST-IOC calculations one has to input some range parameters for the correction functions, in particular, the range parameter L for the Eckart function and the parameter B for the cut-off gaussian function on the reactant and/or product side. These parameters are usually obtained by fitting a lower-level potential energy curve to the corresponding functions. The utility programs **findl.f** and **findb.f**, which are located in the `polyrate9.4.1/util` subdirectory, are designed to do the fitting and return the optimized values of L and B . The user can use the compiling and linking commands in Section 13 of this POLYRATE manual to make executable files (e.g., `findl.exe` and `findb.exe`). To use these programs, just type the name of the executable file, and you will be prompted to input the energies of 3 stationary points for `findl.exe`, or the energies of 2 stationary points and the s value for one of the stationary points for `findb.exe`. Then, for both programs, you will be prompted to input the energy and s value for another point along the lower-level reaction path. This point is usually chosen to be the point that has an energy halfway between the saddle point and the stationary point on the endothermic side of the MEP for `findl.exe` or the reactant/product side stationary point for `findb.exe`. The programs will then print the optimized value of L or B for the Eckart or cut-off gaussian function, respectively. The user can then use these values for the unit `fu50` input [RANGEIC keyword].

15.E. Utility program **rotate30.f**, **rotate31.f**, and **rotate40.f**

ROTATE x ($x = 30, 31, \text{ or } 40$) is a FORTRAN program that uses as an input file a POLYRATE `poly.fu30`, `poly.fu31`, or `poly.fu40` input file and orients the geometries on that

file in a consistent way along the reaction path. This program is based on the Chen orientation algorithm Ref. 60 of Section 20.

The poly.fu30, poly.fu31, and poly.fu40 input files for POLYRATE are usually constructed from electronic structure calculations carried out by one of the several available packages. These packages usually re-orient the input geometry in a way that allows the program to take advantage of the symmetry of the system, or just to put the molecule in the orientation preferred by the program. As a consequence, the input geometry is oriented in a different way than the output geometry, gradients, and/or Hessians. This results errors in: 1) reaction path following algorithms and 2) computation of reaction path curvature. If we do not calculate the reaction path or the curvature of the reaction path, the properties that POLYRATE uses for the dynamical calculations are rotationally invariant, and the results will not be affected by any rotation of any geometry. However, if we want to calculate the reaction path curvature, we need to calculate the derivatives of the Cartesian components of the gradient with respect to the reaction coordinate s . For doing this, we require that all the points along the reaction path be oriented in the same way. Otherwise, the numerical differentiation will lead to erroneous results.

The ROTATE x program uses a geometry that is input by the user (this is assumed to be the saddle point geometry), and it reorients the first point along each side of the reaction path according to this point. The following points along the reaction path are reoriented using as reference geometry the previous point. This reorientation is done by means of the Chen algorithm, as described in the original paper. However, this algorithm only works for small rotations, so before applying Chen's method the program performs a series of trial rotations in order to obtain the closest orientation to the reference geometry, and next it applies Chen's method on the most similar orientation.

Chen's method is based on the minimization of the distance in mass-scaled Cartesian coordinates of the two geometries. In some cases, this minimization does not converge. When that happens, there is no way to solve the reorientation problem with this program.

The program can also recalculate the values of the reaction coordinate, s , as a straight line between each pair of two consecutive points. This recalculation can be useful, e.g., for calculating kinetic isotope effects.

There are three versions of the program ROTATE_x: ROTATE30, ROTATE31, and ROTATE40, and they are used for reorienting the geometries in poly.fu30, poly.fu31, and poly.fu40 input files, respectively. In order to perform this rotation it is necessary to give the geometry of each point. This is sometimes not given on the poly.fu30 and poly.fu40 input files. Thus, when we supply additional information on the gradients of one or two points on each side of the Hessian points in poly.fu30 or poly.fu40, the geometry of these one or two additional points per Hessian point is not given. This is the reason why only poly.fu30 files with LOPT(6) = 2 will work with ROTATE30, and only poly.fu40 input files with GRADER = NOEXTRA will work with ROTATE 40.

An additional limitation of the program is that it only works with poly.fu30 and poly.fu40 input files that do not include any anharmonic information.

Along with the source code, a test run is included for each program, as well as a sample script for compiling and running the code.

The program requires two input files: (1) one of the POLYRATE input files poly.fu30, poly.fu31, or poly.fu40, and (2) a input file for the ROTATE program. The latter input file is described next.

The first line of the input file consists of seven integers that indicate the number of atoms in each system. The first integer is NATOM, the number of atoms in the saddle point structure (which is assumed to be the same as the number of atoms for the points along the reaction path) followed by six integers indicating the number of atoms in the first reactant species, the second reactant, the first product, the second product, the reactants well, and the products well, respectively. If one or more of these species don't exist, a zero has to be input in its place.

The second line is a list of NATOM + 1 real numbers. The first one indicates the mass used for scaling the Cartesian coordinates (which has to be the same as SCALEMASS in the PATH section of the POLYRATE poly.fu5 input file. The following NATOM real numbers are the masses of each of the atoms present in the points along the reaction path, in the same order as in in these points. All the masses have to be input in amu.

The third line is an integer that indicates if the values of s read from the poly.fu30, poly.fu31, or poly.fu40 input files have to be kept unchanged or if the user wants to put in

their places the values computed by the program. If this integer is set to one, the values of s that are read are discarded, and the calculated values are printed in the output file. Any other value leads the program to preserve the s values that are input in the poly.fu30, poly.fu31, or poly.fu40 input file.

The last information to be input in the ROTATE input file is the geometry in Cartesian coordinates of the saddle point in atomic units. The three Cartesian coordinates for each atom have to be preceded by an integer, which acts as a label. The ordering of the atoms has to be the same as in the points along the reaction path, and the labels have to start with the integer 1 for the first atom and finish with the integer NATOM for the last atom.

The output of the program, poly.fu60, is a file with the format same as poly.fu30, poly.fu31, or poly.fu40, which can be used in a POLYRATE calculation directly without further modification.

16. UNIX C SHELL SCRIPTS

POLYRATE includes many useful scripts for the user. The following subsections discuss the functions of these scripts and explain how to use them.

16.A. Installation script

This script, `configure`, resides in the `polyrate9.4.1` directory. It is used to install the POLYRATE program on your computer. Usually, the first thing you do after untarring the distribution tar file is to run this script. (See Subsection 13.A.) Actually, this script does four things:

1. Makes a hidden file `.poly_path` in your home directory.

The hidden file contains the path to the `polyrate9.4.1` directory in the file system you are using. This lets other scripts in POLYRATE 9.4.1 easily locate the necessary files. If you lose the hidden file for some reason, or if the file system changes after you install POLYRATE, you need to run this script again to generate a new `.poly_path` file.

2. Detects compilers, operating system, and processor type

The script will automatically search for the FORTRAN and C compilers that are correct for the configuration being used.

3. Copies machine-dependent files and puts them into the proper directories with the default file names.

It copies the appropriate date-and-time routine from the `polyrate9.4.1/util` directory to the `polyrate9.4.1/src` directory and names it `dattim.f`.

4. Creates the Makefile

The Makefile is used to compile and link all of the code needed for a job. If any changes are made to the files used to create the executable, the Makefile will automatically update the executable.

See section 13.A.1 for a description of the directory structure following a successful installation.

16.B. Potential energy surface compiling scripts

This script is distributed in version 9.4.1, but it is no longer needed when using Makefile.

There are two potential energy surface subroutine compiling scripts in the polyrate9.4.1/script directory. The script comp_pot.jc is used to compile all the potential energy surface subroutines for the test runs. (See Subsection 13.A.2.)

The script potcomp.jc is designed to compile a single file of potential energy surface subroutines. It is general, and you can use it to compile your own potential subroutines. This script takes two arguments; the first one is the potential file name without extension, and the second one is the number of atoms involved in the potential. For example, suppose you have a set of FORTRAN potential energy subroutines which defines the potential energy surface of a system that consists of one oxygen atom and 3 hydrogen atoms, and the set of subroutines has the file name myoh3.f. Put this file in the polyrate9.4.1/poten directory. You then change to the polyrate9.4.1/script directory and type potcomp.jc myoh3 4 <Return>; the script will compile the file myoh3.f for you and put the object module myoh3.o in the polyrate9.4.1/obj directory. You can, if you wish, copy the script into the polyrate9.4.1/poten directory so you can use it more conveniently. Remember that you must put your potential source code in the polyrate9.4.1/poten directory. It should be noted that since the nh3 potential energy surface subroutine provided in the distribution package has already been written the way that is required by POLYRATE, when compiling this potential energy subroutine with the script potcomp.jc, you should use a number that is not 3 or 4 as the second argument. This is because the nh3.f subprogram already contains the necessary SETUP, SURF, COORD, and CHAIN subroutines.

16.C. Source code compiling script

This script is distributed in version 9.4.1, but it is no longer needed when using Makefile.

The script comp_src.jc in the polyrate9.4.1/script directory is used for compiling the POLYRATE source code. It takes one argument, the parameter INCLUDE file number. You probably used this script when you did the testing. (See Subsection 13.A.2.) Usually, after you compile the source code with a certain parameter INCLUDE file and keep the library file that the script generated, you don't need to recompile the source code unless you need to use a different parameter INCLUDE file. For example, suppose you write your own INCLUDE file named param5.inc, and you want to compile the source code with this INCLUDE file. Put the param5.inc file in the polyrate9.4.1/src directory, change to the polyrate9.4.1/script directory, and type comp_src.jc 5 & <Return>. The script will

compile the source code with the parameter INCLUDE file param5.inc and generate a library file named libpoly5.a in the polyrate9.4.1/obj directory. You can then use the linking script described in the next subsection to link this library file with your potential object module to make an executable file. Remember that you must put the parameter INCLUDE file in the polyrate9.4.1/src directory and always name a new parameter file param#.inc where # can be any integer greater than 4.

16.D. Linking script

This script is distributed in version 9.4.1, but it is no longer needed when using Makefile.

The script link_exe.jc in the polyrate9.4.1/script directory is used for linking the potential object file with the source code library file to make an executable file. It takes 2 arguments. The first one is the library file number, which is just the parameter INCLUDE file number you used when making the source code library file. (See the Subsection above.) The second argument is the potential file name (without the extension) you want to use. For example, if you want to use the source code library compiled with the parameter file param5.inc and the potential object module myoh3.o to make an executable file, change your working directory to the polyrate9.4.1/script directory and type link_exe.jc 5 myoh3 <Return>. The script will link the source code library and potential object files together and generate the executable file poly_myoh3.exe and a short summary file poly_myoh3.doc about this new executable file in the polyrate9.4.1/exe directory.

16.E. Test run scripts

The test run scripts have been discussed in Subsection 13.A.2. These scripts are designed such that you don't need to know much about POLYRATE in order to run the test runs. Some of the test runs might take a long time to run (see Section 19) so you should consider running them in background or submitting them as batch jobs. Some test runs generate very big output files that you may want to delete together with the test run executable files when you need to free up some disk space.

17. FILE USAGE

The POLYRATE program uses several files for input data, for storing restart information (optional), and for output. The program explicitly opens and closes all files required for a given run, and the files are assumed to have the file names poly.fu#, where # is an integer. The file poly.fu# connects to a FORTRAN I/O unit fu# which is defined in the INCLUDE file param.inc (see Subsection 5.A). This means that at the beginning of any run, all the POLYRATE input data files which have logical file names differing from poly.fu# must be linked or copied to the corresponding poly.fu# file. If the user wishes to assign different logical names to the output files, which are also poly.fu# files, these reassignments must be carried out after the calculation has completed. Any data file(s) needed by the potential energy surface subroutine is (are) not opened by the POLYRATE program and therefore, do not automatically follow the poly.fu# format.

One would ordinarily expect FORTRAN unit 5 to be assigned to fu5, FORTRAN unit 6 to fu6, and so forth. The fu# system, however, allows one to avoid conflicts when hooking POLYRATE up to other programs, e.g., MOPAC or analytic potentials.

<u>Unit</u>	<u>Usage</u>
fu1	Reaction-path information for restarting the program
fu2	Additional reaction-path information to be merged with that from unit fu1
fu3	Output for merged reaction path information
fu4	Reserved for potential energy surface data (Note: A potential subroutine may read data from more than one file, if desired. In the test suite we always use unit 4 if data is read from only one file, and we use units 4 and 7 for the cases where data is read from two files)
fu5	General input data. This data has several sections—see section 11.A of this manual for the details.
fu6	Output (This is the full output file. Selected subsets of this output are also written to files poly.fu14 and poly.fu15)
fu7	Usually reserved for additional potential energy surface data (See Subsections 8.C and 14.B for further discussion)
fu14	Output table of dynamical bottleneck properties
fu15	Output table of selected forward rate constants

fu16	Reserved for output from electronic structure packages
fu22	Detailed output for the LCT calculations [LCTOPT: PRFREQ, PRPROB]
fu24	Output file of MEP and large curvature representative tunneling termini in mass-scaled Jacobian coordinates.
fu25	Output file of the potential energy information along the MEP
fu26	Output file of the generalized transition state frequency information along the MEP
fu27	Generalized transition state geometry information along the MEP in the XYZ input format of the XMOL program
fu28	Bond angles, bond distances, and dihedral angles along the MEP
fu29	Electronic structure input file for IVTST-0 and IVTST-1 calculations
fu30	Formatted electronic structure input file used if POTENTIAL = <i>unit 30</i> in the ENERGETICS Section of file fu5
fu31	Keyword-based electronic structure input file used if POTENTIAL = <i>unit31</i> in the ENERGETICS section of file fu5.
fu40	Keyword-based electronic structure input file used if POTENTIAL = <i>unit40</i> in the ENERGETICS section of file fu5
fu41–fu47	Output tables of LCT calculations [LCTDETAIL]
fu48, fu49	Output files for LCT restart calculations
fu50	Input file for higher-level data for VTST-IOC calculations [DL keyword set to <i>ioc</i> or ICOPT]
fu51	Input file for higher-level data for VTST-ISPE calculations [DL keyword set to <i>ispe</i>]
fu60	Reserved for the direct dynamics intermediate file
fu61	Reserved for information about stationary points that can be used to restart a calculation by the second restart method of Section 7. C.
fu62	Reserved for a subset of information that are presented in fu61 if the argument of the WRITE62 keyword is not set to <i>none</i>
fu65	Generalized transition state geometry information along the MEP in the XYZ input format of the MDMOVIE program

fu70	Reserved for input data needed by PREP
fu71	Reserved for input data needed by PREPJ when JTYPE = 1 (REACT1)
fu72	Reserved for input data needed by PREPJ when JTYPE = 2 (REACT2)
fu73	Reserved for input data needed by PREPJ when JTYPE = 3 (PROD1)
fu74	Reserved for input data needed by PREPJ when JTYPE = 4 (PROD2)
fu75	Reserved for input data needed by PREPJ when JTYPE = 5 (START)
fu76	Reserved for input data needed by PREPJ when JTYPE = 6 (PATH)
fu77	Reserved for input data needed by PREPJ when JTYPE = 7 (WELLR)
fu78	Reserved for input data needed by PREPJ when JTYPE = 8 (WELLP)
fu80-fu89	Reserved for electronic structure codes and interfaces

18. MEMORY REQUIREMENTS

Memory (in Megabytes) required for the POLYRATE test suite executable files on all the machines on which the code was tested. The parameter INCLUDE file used is param4.inc for the ch5, cwmc, and ch4o test runs, param3.inc for the dumpot test runs, and param2.inc for all other test runs. The memory requirements were determined using version 8.1 and the Unix SIZE command.

Test run	dumpot*	ch4o	ch5	clhbr	cmc	cwmc
IBM SP	36.9	37.0	37.0	15.7	15.7	37.0
SGI	36.2	36.2	36.2	15.1	15.1	36.3

Test run	h2ni	h3	hni	ho2	nh3	oh3
IBM SP	15.7	15.7	22.9	15.7	15.7	15.7
SGI	15.2	15.1	22.2	15.1	15.1	15.1

* The dummy potential is used for the fu30, fu31, and fu40 test runs.

19. TEST RUNS AND TEST RUN TIMINGS

19.A. Extent of testing

The full test suite of version 8.1 was run on the IBM SP with PowerPC 604e processor (AIX 4.3.1) and Silicon Graphics Power Challenge L with R10000 processor (IRIX 6.5).

19.B. Timings

The following timing trials for POLYRATE may be of interest. The running time of POLYRATE varies quite a bit, depending mainly on the number of atoms, the complexity of the potential, the range of reaction coordinate examined, step sizes, and options for anharmonicity and tunneling. The rest of this section gives some examples of timings. The times (system time + user time) are in CPU seconds. The timing under SGI is for the Power Challenge L machine.

1) ch4cl test run

	IBM SP	SGI
ch4cltr1	0.0	0.1
ch4cltr2	0.1	0.3
Total	0.1	0.4

2) ch4o testrun

	IBM SP	SGI
ch4otr1	26.7	44.8

3) ch4oh test runs

	IBM SP	SGI
ch4ohtr1	0.3	0.2
ch4ohtr2	0.1	0.2
Total	0.4	0.4

4) ch5 test runs

	IBM SP	SGI
ch5fu29tr1	0.0	0.1
ch5fu29tr2	0.1	0.1
ch5fu30tr1	0.0	0.0
ch5fu30tr2	0.0	0.0
ch5fu30tr3	0.0	0.0
ch5fu30tr4	0.0	0.0
ch5fu30tr5	7.7	12.1
ch5fu30tr6	2.1	3.2
ch5fu31tr1	4.6	7.2
ch5fu40tr1	0.0	0.0
ch5fu40tr2	0.0	0.1
ch5fu40tr3	0.0	0.1
ch5fu40tr4	0.0	0.0
ch5fu40tr5	7.7	12.2
ch5fu40tr6	2.2	3.5
ch5fu50tr1	0.0	0.1
ch5fu50tr2	7.3	10.6
ch5fu50tr3	7.4	10.5
ch5fu51tr1	2.8	5.1
ch5icfu30tr1	17.3	27.1
ch5icfu30tr2	17.4	27.1
ch5icfu31tr1	13.6	21.2
ch5fcfu40tr1	17.3	27.3
ch5icfu40tr2	17.4	27.4
ch5j1tr1	6.3	9.5
ch5j1tr2	4.0	4.3
ch5j1tr3	9.0	13.3
ch5j1tr4	6.3	9.8
ch5j1tr5	5.5	8.6
ch5j1tr6	8.0	8.2
ch5j1tr7	9.1	13.4
ch5j1tr8	0.2	0.4
ch5j2itr1	0.0	0.1
ch5j2itr2	0.1	0.3
ch5j2tr1	1.2	2.0
ch5j2tr2	1.2	2.0
ch5j2tr3	1.6	2.9
Total	177.4	269.8

5) clhbr test runs

	IBM SP	SGI
clhbrtr1	6.9	10.5
clhbrtr2	5.5	7.9
Total	12.4	18.4

6) cmc test runs

	IBM SP	SGI
cmctr1	11.8	21.9
cmctr2	7.3	13.0
cmctr3	11.9	22.4
Total	31.0	57.3

7) cwmc test runs

	IBM SP	SGI
cwmctr1	0.2	0.4
cwmctr2	35.2	64.9
cwmctr3	34.8	70.9
Total	70.2	136.2

8) h2ni test run

	IBM SP	SGI
h2nitr1	22.5	37.6

9) h3 test runs

	IBM SP	SGI
h3tr1	3.0	6.5
h3tr2	3.3	4.5
h3tr3	1.3	2.3
h3tr4	2.3	5.5
Total	9.9	18.8

10) hni test run

	IBM SP	SGI
hnit1	88.6	118.7

11) ho2 test runs

	IBM SP	SGI
ho2tr1	3.1	5.7
ho2tr2	2.8	5.1
ho2tr3	16.4	24.6
ho2tr4	5.3	10.7
ho2tr5	3.4	6.9
Total	31.0	53.0

12) nh3 test run

	IBM SP	SGI
nh3tr1	1.0	2.8
nh3tr2	0.5	1.6

13) oh3 test runs

	IBM SP	SGI
oh3fu30tr1	0.6	0.9
oh3fu30tr2	0.1	0.2
oh3fu30tr3	0.1	0.1
oh3fu30tr4	0.0	0.0
oh3fu30tr5	0.0	0.0
oh3fu30tr6	0.1	0.3
oh3fu31tr1	2.0	3.4
oh3fu31tr2	7.5	11.6
oh3fu40tr1	0.6	0.9
oh3fu40tr2	0.1	0.2
oh3fu40tr3	0.0	0.1
oh3fu40tr4	0.0	0.0
oh3fu40tr5	0.0	0.1
oh3fu40tr6	0.2	0.3
oh3tr1	10.6	4.9
oh3tr2	1.8	2.5
oh3tr3	1.5	2.5
oh3tr4	2.8	4.2
oh3tr5	2.6	3.9
oh3tr6	1.6	2.6
oh3tr7	0.3	0.6
oh3tr8	0.1	0.2
oh3tr9	1.6	2.3
oh3tr10	2.0	3.0
Total	36.2	44.8

20. FOOTNOTES TO TEXT

- [1] R. E. Weston, Jr. and H. A. Schwarz, *Chemical Kinetics* (Prentice-Hall, Englewood Cliffs, 1972), p. 101.
- [2] I. Shavitt, Theoretical Chemistry Laboratory Report No. WIS-AEC-23 (University of Wisconsin, Madison, 1959); R. E. Weston, Jr., *J. Chem. Phys.* **31** (1959) 892; I. Shavitt, *J. Chem. Phys.* **49** (1968) 4048.
- [3] R. A. Marcus, *J. Chem. Phys.* **45** (1966) 4493; **49** (1968) 2610, 2617.
- [4] D. G. Truhlar and A. Kuppermann, *J. Chem. Phys.* **52** (1970) 3842; *J. Amer. Chem. Soc.* **93** (1971) 1840.
- [5] K. Fukui, in *The World of Quantum Chemistry*, R. Daudel and B. Pullman, Eds. (Reidel, Dordrecht, 1974), p. 113.
- [6] B. C. Garrett and D. G. Truhlar, *J. Chem. Phys.* **70** (1979) 1593.
- [7] W. H. Miller, N. C. Handy, and J. E. Adams, *J. Chem. Phys.* **72** (1980) 99.
- [8] G. Herzberg, *Infrared and Raman Spectra of Polyatomic Molecules* (D. van Nostrand, Princeton, NJ, 1945), pp. 505-10.
- [9] D. A. McQuarrie, *Statistical Mechanics* (Harper & Row, New York, 1976), pp. 133-6.
- [10] B. C. Garrett, D. G. Truhlar, R. S. Grev, and A. W. Magnuson, *J. Phys. Chem.* **84** (1980) 1730, **87** (1983) 4554E.
- [11] R. T. Skodje, D. G. Truhlar, and B. C. Garrett, *J. Phys. Chem.* **85** (1981) 3019.
- [12] B. C. Garrett, N. Abusalbi, D. Kouri, and D. G. Truhlar, *J. Chem. Phys.* **83** (1985) 2252.
- [13] D. G. Truhlar and B. C. Garrett, *Accounts Chem. Res.* **13** (1980) 440.
- [14] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes* (Cambridge University Press, Cambridge, 1986), p. 412.
- [15] *ibid.*, p. 418-422.
- [16] *ibid.*, p. 283.
- [17] M. Page and J. W. McIver, Jr., *J. Chem. Phys.* **88** (1988) 922.
- [18] B. C. Garrett and D. G. Truhlar, *J. Phys. Chem.* **83** (1979) 200, 1052, **83** (1979) 3058E, **87** (1983) 4553E.
- [19] K. Ishida, K. Morokuma, and A. Komornicki, *J. Chem. Phys.* **66** (1977) 2153.
- [20] M. W. Schmidt, M. S. Gordon, and M. Dupuis, *J. Amer. Chem. Soc.* **107** (1985) 2585.

- [21] L. D. Landau and E. M. Lifshitz, *Quantum Mechanics*, 2nd ed. (Pergamon, Oxford, 1965), p. 163.
- [22] J. N. L. Connor, *Chem. Phys. Lett.* **4** (1969) 419.
- [23] S. C. Tucker and D. G. Truhlar, *J. Amer. Chem. Soc.* **112** (1990) 3338.
- [24] X. G. Zhao, D.-h. Lu, Y.-P. Liu, G. C. Lynch, and D. G. Truhlar, *J. Chem. Phys.* **97** (1992) 6369.
- [25] T. N. Truong and D. G. Truhlar, *J. Chem. Phys.* **93** (1990) 2125.
- [26] V. K. Babamov, V. Lopez, and R. A. Marcus, *J. Chem. Phys.* **78** (1983) 5621.
- [27] X. G. Zhao, A. Gonzalez-Lafont, D. G. Truhlar, and R. Steckler, *J. Chem. Phys.* **94** (1991) 5544.
- [28] A. J. C. Varandas, F. B. Brown, C. A. Mead, D. G. Truhlar, and N. C. Blais, *J. Chem. Phys.* **85** (1987) 6258.
- [29] C. F. Melius and R. J. Blint, *Chem. Phys. Lett.* **64** (1979) 183.
- [30] S. P. Walch and T. H. Dunning, *J. Chem. Phys.* **72** (1980) 72.
- [31] G. C. Schatz and H. Elgersma, *Chem. Phys. Lett.* **73** (1980) 21.
- [32] T. N. Truong and D. G. Truhlar, *J. Phys. Chem.* **94** (1990) 8262.
- [33] C. Y. Lee and A. E. DePristo, *J. Chem. Phys.* **85** (1986) 4161.
- [34] S. C. Tucker and D. G. Truhlar, *J. Phys. Chem.* **93** (1989) 8138.
- [35] R. Q. Topper, Q. Zhang, Y.-P. Liu, and D. G. Truhlar, *J. Chem. Phys.* **98** (1993) 4991.
- [36] E. B. Wilson, Jr., J. C. Decius, and P. C. Cross, *Molecular Vibrations*, (MacGraw-Hill, New York, 1955).
- [37] G. Fogarasi and P. Pulay, in *Vibrational Spectra and Structure*, Vol. 14, edited by J. R. Durig (Elsevier, Amsterdam, 1985), p. 125.
- [38] M. A. Pariseau, I. Suzuki, and J. Overend, *J. Chem. Phys.* **42** (1965) 2335.
- [39] A. R. Hoy, I. M. Mills, and G. Strey, *Mol. Phys.* **24** (1972) 1265.
- [40] J. Cioslowski and M. Challacombe, ANHTRAX, QCPE Program No. 609 (Quantum Chemistry Program Exchange, Bloomington, IN, 1991).
- [41] M. Challacombe and J. Cioslowski, *J. Chem. Phys.* **95** (1991) 1064.
- [42] D. G. Truhlar and A. D. Isaacson, *J. Chem. Phys.* **94** (1991) 357.
- [43] D. G. Truhlar, N. J. Kilpatrick, and B. C. Garrett, *J. Chem. Phys.* **78** (1983) 2438.

- [44] B. C. Garrett and D. G. Truhlar, *J. Chem. Phys.* **72** (1980) 3460.
- [45] D. G. Truhlar, *J. Chem. Soc. Faraday Trans.* **90** (1994) 1740.
- [46] D. G. Truhlar and M. S. Gordon, *Science* **249** (1990) 491.
- [47] D. G. Truhlar, B. C. Garrett, and S. J. Klippenstein, *J. Phys. Chem.* **100** (1996) 12771.
- [48] T. N. Truong, *J. Chem. Phys.* **100** (1994) 8014.
- [49] B. C. Garrett and D. G. Truhlar, *J. Phys. Chem.* **83** (1979) 1079.
- [50] S. L. Mielke, D. G. Truhlar, and D. W. Schwenke, *J. Phys. Chem.* **99** (1995) 16210.
- [51] B. C. Garrett and D. G. Truhlar, *J. Chem. Phys.* **66** (1982) 1853.
- [52] J. C. Corchado, J. Espinosa-Garcia, O. Roberto-Neto, Y.-Y. Chuang, and D. G. Truhlar, *J. Phys. Chem. A* **102** (1998) 4899.
- [53] T. C. Allison and D. G. Truhlar, in *Modern Methods for Multidimensional Dynamics Computations in Chemistry*, edited by D. L. Thompson (World Scientific, Singapore, 1998), p. 618.
- [54] J. Backer, F. Jensen, H. S. Rzepa, A. Stebbings, LOCATION OF TRANSITION STATES IN AMPAC AND MOPAC USING EIGENVECTOR FOLLOWING (EF), QCPE Program No. 597 (Quantum Chemistry Program Exchange, Bloomington, IN, 1990).
- [55] A. Banerjee, N. Adams, J. Simons, and R. Shepard, *J. Phys. Chem.* **89** (1985) 52.
- [56] J. Baker, *J. Comp. Chem.* **7** (1985) 385.
- [57] P. Culot, G. Dive, V. J. Nguyen, and J. M. Ghuysen, *Theor. Chim. Acta* **82** (1992) 189.
- [58] T. Helgaker, *Chem. Phys. Lett.* **182** (1991) 503.
- [59] K. S. Pitzer, *J. Chem. Phys.* **14** (1946) 239.
- [60] Z. Chen, *Theor. Chim. Acta* **75** (1989) 481.

21. BIBLIOGRAPHY

The following is a list of some publications reporting calculations carried out with POLYRATE or with its original version, which was entitled ICVTST. This includes all articles in which new procedures incorporated in POLYRATE are described, but it does not include all applications.

1. "Polyatomic Canonical Variational Theory for Chemical Reaction Rates. Separable-Mode Formalism with Application to $\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$ ", A. D. Isaacson and D. G. Truhlar, *J. Chem. Phys.* **76** (1982) 1380.
2. "The Incorporation of Quantum Effects in Generalized Transition State Theory", D. G. Truhlar, A. D. Isaacson, R. T. Skodje, and B. C. Garrett, *J. Phys. Chem.* **86** (1982) 2252; erratum: **87** (1983) 4554.
3. "Statistical-Diabatic Model for State-Selected Reaction Rates. Theory and Application of Vibrational-Mode Correlation Analysis to $\text{OH}(n_{\text{OH}}) + \text{H}_2(n_{\text{HH}}) \rightarrow \text{H}_2\text{O} + \text{H}$ ", D. G. Truhlar and A. D. Isaacson, *J. Chem. Phys.* **77** (1982) 3516.
4. "Variational Transition State Theory Calculations for an Atom-Radical Reaction with no Saddle Point: $\text{O} + \text{OH}$ ", S. N. Rai and D. G. Truhlar, *J. Chem. Phys.* **79** (1983) 6049.
5. "Generalized Transition State Theory", D. G. Truhlar, A. D. Isaacson, and B. C. Garrett, in *The Theory of Chemical Reaction Dynamics*, edited by M. Baer (CRC Press, Boca Raton, FL, 1985), Vol. 4, pp. 65-137.
6. "Improved Canonical and Microcanonical Variational Transition State Theory Calculations for a Polyatomic System: $\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$ ", A. D. Isaacson, M. T. Sund, S. N. Rai, and D. G. Truhlar, *J. Chem. Phys.* **82** (1985) 1338.
7. "Semiclassical Reaction-Path Methods Applied to Calculate the Tunneling Splitting in NH_3 ", F. B. Brown, S. C. Tucker, and D. G. Truhlar, *J. Chem. Phys.* **83** (1985) 4451.
8. "Diffusion of Hydrogen, Deuterium, and Tritium on the (100) Plane of Copper: Reaction-Path Formulation, Variational Transition State Theory and Tunneling Calculations", J. G. Lauderdale and D. G. Truhlar, *Surface Science* **164** (1985) 558.
9. "Embedded-Cluster Model for the Effect of Phonons on Hydrogen Surface Diffusion on Copper", J. G. Lauderdale and D. G. Truhlar, *J. Chem. Phys.* **84** (1986) 1843.
10. "Test of Variational Transition State Theory and the Least-Action Approximation for Multidimensional Tunneling Probabilities Against Accurate Quantal Rate Constants for a Collinear Reaction Involving Tunneling into an Excited State", B. C. Garrett, N. Abusalbi, D. J. Kouri, and D. G. Truhlar, *J. Chem. Phys.* **83** (1985), 2252.

11. "The Representation and Use of Potential Energy Surfaces in the Wide Vicinity of a Reaction Path for Dynamics Calculations on Polyatomic Reactions", D. G. Truhlar, F. B. Brown, R. Steckler, and A. D. Isaacson, in *The Theory of Chemical Reaction Dynamics*, edited by D. C. Clary (D. Reidel, Dordrecht, Holland, 1986), pp. 285.
12. "Semiclassical Variational Transition State Calculations for the Reactions of H and D with Thermal and Vibrationally Excited H₂", B. C. Garrett, D. G. Truhlar, A. J. C. Varandas, and N. C. Blais, *International J. Chem. Kinetics* **18** (1986) 1065.
13. "Reaction-Path Analysis of the Tunneling Splitting in Fluxional Molecules: Application to the Degenerate Rearrangement of Hydrogen Fluoride Dimer", G. C. Hancock, P. A. Rejto, R. Steckler, F. B. Brown, D. W. Schwenke, and D. G. Truhlar, *J. Chem. Phys.* **85** (1986) 4997.
- 13A. "Evaluation of Dynamical Approximations for Calculating the Effect of Vibrational Excitation on Reaction Rates", B. C. Garrett, D. G. Truhlar, JM Bowman, and AF Wagner, *J. Phys. Chem.* **90** (1986) 4305.
14. "POLYRATE: A General Computer Program for Variational Transition State Theory and Semiclassical Tunneling Calculations of Chemical Reaction Rates", A. D. Isaacson, D. G. Truhlar, S. N. Rai, R. Steckler, G. C. Hancock, B. C. Garrett, and M. J. Redmon, *Comp. Phys. Comm.* **47** (1987) 91.
15. "Surface Diffusion of H on Copper: The Effect of Phonon-Adsorbate Coupling on the Diffusion Rate", T. N. Truong and D. G. Truhlar, *J. Phys. Chem.* **91** (1987) 6229.
16. "A Comparative Study of Potential Energy Surfaces for CH₃ + H₂ → CH₄ + H", R. Steckler, K. Dykema, F. B. Brown, D. G. Truhlar, and T. Valencich, *J. Chem. Phys.* **87** (1987) 7024.
17. "A New Potential Energy Surface for the CH₃ + H₂ → CH₄ + H Reaction: Calibration, Rate Constants, and Kinetic Isotope Effects by Variational Transition State Theory and Semiclassical Tunneling Calculations", T. Joseph, R. Steckler, and D. G. Truhlar, *J. Chem. Phys.* **87** (1987) 7036.
18. "Algorithms and Accuracy Requirements for Computing Reaction Paths by the Method of Steepest Descent", B. C. Garrett, M. J. Redmon, R. Steckler, D. G. Truhlar, M. S. Gordon, K. K. Baldrige, and D. Bartol, *J. Phys. Chem.* **92** (1988) 1476.
19. "Calculation of Reaction Rates and Kinetic Isotope Effects for Dissociative Chemisorption of H₂ and D₂ on Ni(100), Ni(110) and Ni(111) Surfaces", T. N. Truong, G. Hancock, and D. G. Truhlar, *Surface Science* **214** (1989) 523.
20. "Ab Initio Reaction Paths and Direct Dynamics Calculations", K. K. Baldrige, M. S. Gordon, R. Steckler, and D. G. Truhlar, *J. Phys. Chem.* **93** (1989) 5107.
21. "Application of the Large-Curvature Tunneling Approximation to Polyatomic Molecules: Abstraction of H or D by Methyl Radical", B. C. Garrett, T. Joseph, T. N. Truong, and D. G. Truhlar, *Chem. Phys.* **136** (1989) 271; erratum: **140** (1990) 207.

22. "Transition State Structure, Barrier Height, and Vibrational Frequencies for the Reaction $\text{Cl} + \text{CH}_4 \rightarrow \text{CH}_3 + \text{HCl}$ ", T. N. Truong, D. G. Truhlar, K. K. Baldrige, M. S. Gordon, and R. Steckler, *J. Chem. Phys.* **90** (1989) 7137.
23. "Projection Operator Method for Geometry Optimization with Constraints", D.-h. Lu, M. Zhao, and D. G. Truhlar, *J. Comp. Chem.* **12** (1991) 376.
24. "A Simple Approximation for the Vibrational Partition Function of a Hindered Internal Rotation", D. G. Truhlar, *J. Comp. Chem.* **12** (1991) 266.
25. "The Definition of Reaction Coordinates for Reaction-Path Dynamics", G. A. Natanson, B. C. Garrett, T. N. Truong, T. Joseph, and D. G. Truhlar, *J. Chem. Phys.* **94** (1991) 7875.
26. "Interpolated Variational Transition State Theory: Practical Methods for Estimating Variational Transition State Properties and Tunneling Contributions to Chemical Reaction Rates from Electronic Structure Calculations", A. Gonzalez-Lafont, T. N. Truong, and D. G. Truhlar, *J. Chem. Phys.* **95** (1991) 8875.
27. "Optimized Calculation of Reaction Paths and Reaction-Path Functions for Chemical Reactions", V. S. Melissas, D. G. Truhlar, and B. C. Garrett, *J. Chem. Phys.* **96** (1992) 5758.
28. "Variational Transition-State Theory with Multidimensional, Semiclassical, Ground-State Transmission Coefficients: Application to Secondary Deuterium Kinetic Isotope Effects in Reactions Involving Methane and Chloromethane", D. G. Truhlar, D.-h. Lu, S. C. Tucker, X. G. Zhao, A. Gonzalez-Lafont, T. N. Truong, D. Maurice, Y.-P. Liu, and G. C. Lynch, *ACS Symp. Series* **502** (1992) 16.
29. "POLYRATE 4: A New Version of a Computer Program for the Calculation of Chemical Reaction Rates for Polyatomics", D.-h. Lu, T. N. Truong, V. S. Melissas, G. C. Lynch, Y.-P. Liu, B. C. Garrett, R. Steckler, A. D. Isaacson, S. N. Rai, G. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar, *Comp. Phys. Comm.* **71** (1992) 235.
30. "MORATE: A Program for Direct Dynamics Calculations of Chemical Reaction Rates by Semiempirical Molecular Orbital Theory", T. N. Truong, D.-h. Lu, G. C. Lynch, Y.-P. Liu, V. S. Melissas, J. J. P. Stewart, R. Steckler, B. C. Garrett, A. D. Isaacson, A. Gonzalez-Lafont, S. N. Rai, G. C. Hancock, T. Joseph, and D. G. Truhlar, *Comp. Phys. Comm.* **75** (1993) 143.
31. "Molecular Modeling of the Kinetic Isotope Effect for the [1,5]-Sigmatropic Rearrangement of cis-1,3-Pentadiene", Y.-P. Liu, G. C. Lynch, T. N. Truong, Da-hong Lu, and D. G. Truhlar, *J. Amer. Chem. Soc.* **115** (1993) 2408.
32. "Interpolated Variational Transition State Theory and Tunneling Calculations of the Rate Constant of the Reaction $\text{OH} + \text{CH}_4$ at 223-2400 K", V. S. Melissas and D. G. Truhlar, *J. Chem. Phys.* **99** (1993) 1013.
33. "Bond-Distance and Bond-Angle Constraints in Reaction-Path Dynamics Calculations", D.-h. Lu and D. G. Truhlar, *J. Chem. Phys.* **99** (1993) 2723.

34. "Direct Dynamics Calculation of the Kinetic Isotope Effect for an Organic Hydrogen-Transfer Reaction, Including Corner-Cutting Tunneling in 21 Dimensions", Y.-P. Liu, D.-h. Lu, A. Gonzalez-Lafont, D. G. Truhlar, and B. C. Garrett, *J. Amer. Chem. Soc.* **115** (1993) 7806.
35. "Effect of Phonon Coupling on Hydrogen Tunneling Rates at Gas-Surface Interfaces", S. E. Wonchoba and D. G. Truhlar, *J. Chem. Phys.* **99** (1993) 9637.
36. "Variational Transition State Theory and Semiclassical Tunneling Calculations with Interpolated Corrections: A New Approach to Interfacing Electronic Structure Theory and Dynamics for Organic Reactions", W.-P. Hu, Y.-P. Liu, and D. G. Truhlar, *Faraday Trans. Chem. Soc.* **90** (1994) 1715.
37. "Reaction Path Approaches to Dynamics at a Gas-Solid Interface: Quantum Tunneling Effects for an Adatom on a Non-Rigid Metallic Surface", S. E. Wonchoba, W.-P. Hu, and D. G. Truhlar in *Theoretical and Computational Approaches to Interface Phenomena*, edited by H. L. Sellers and J. T. Golab (Plenum, New York, 1994), p. 1.
38. "Surface Diffusion of H on Ni(100). Interpretation of the Transition Temperature", S. E. Wonchoba, W.-P. Hu, and D. G. Truhlar, *Phys. Rev. B* **51** (1995) 9985.
39. "POLYRATE 6.5: A New Version of Computer Program for the Calculation of Chemical Reaction Rates for Polyatomics", R. Steckler, W.-P. Hu, Y.-P. Liu, G. C. Lynch, B. C. Garrett, A. D. Isaacson, V. S. Melissas, D.-h. Lu, T. N. Truong, S. N. Rai, G. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar, *Computer Physics Communications* **88** (1995) 341. [new version announcement]
40. "Direct Dynamics Method for the Calculation of Reaction Rates", D. G. Truhlar, in *The Reaction Path in Chemistry: Current Approaches and Perspectives*, edited by D. Heidrich (Kluwer, Dordrecht, 1995), p. 229.
41. "Reaction-Path Potential and Vibrational Frequencies in Terms of Curvilinear Internal Coordinates", C. F. Jackels, Zhen Gu, and D. G. Truhlar, *J. Chem. Phys.* **102** (1995) 3188.
42. "Dual-Level Reaction Path Dynamics (The /// Approach to VTST with Semiclassical Tunneling). Application to $\text{OH} + \text{NH}_3 \rightarrow \text{H}_2\text{O} + \text{NH}_2$ " J.C. Corchado, J. Espinosa-Garcia, W.-P. Wu, I. Rossi, and D. G. Truhlar, *J. Phys. Chem.* **99** (1995) 687.
43. "Reaction-Path Dynamics in Curvilinear Internal Coordinates Including Torsions," K. A. Nguyen, C. F. Jackels, and D. G. Truhlar, *J. Chem. Phys.* **104** (1996) 6491.
44. "Improved Dual-Level Direct Dynamics Method for Reaction Rate Calculations with Inclusion of Multidimensional Tunneling Effects and Validation for the Reaction of H with *trans*- N_2H_2 " Y.-Y. Chuang, and D. G. Truhlar, *J. Phys. Chem. A* **101** (1997) 3808; erratum: **101**, 8741 (1997).

45. "Reaction-Path Dynamics with Harmonic Vibrational Frequencies in Curvilinear Internal Coordinates: $\text{H} + \text{trans-N}_2\text{H}_2 \rightarrow \text{H}_2 + \text{N}_2\text{H}$ ", Y.-Y. Chuang and D. G. Truhlar, *J. Chem. Phys.* **107** (1997) 83.
46. "Reaction Path Dynamics in Redundant Internal Coordinates," Y.-Y. Chuang and D. G. Truhlar, *J. Phys. Chem. A* **102** (1998) 242.
47. "Variation Transition State Theory Without the Minimum-Energy Path," J. Villà and D. G. Truhlar, *Theor. Chem. Acc.* **97** (1997) 317.
48. "Mapped Interpolation Methods for Variational Transition State Theory," J. C. Corchado, E. L. Coitiño, Y.-Y. Chuang, P. L. Fast, D. G. Truhlar, *J. Phys. Chem. A* **102** (1998) 2424.
49. "Variational Reaction Path Algorithm," P. L. Fast and D. G. Truhlar, *J. Chem. Phys.* **109** (1998) 3721.
50. "The Calculation of Kinetic Isotope Effects Based on a Single Reaction Path," P. L. Fast, J. C. Corchado, and D. G. Truhlar, *J. Chem. Phys.* **109** (1998) 6237.
51. "Mapped Interpolation Scheme for Single-Point Energy Corrections in Reaction Rate Calculations and a Critical Evaluation of Dual-Level Reaction-Path Dynamics Methods," Y.-Y. Chuang, J. C. Corchado, and D. G. Truhlar, *J. Phys. Chem. A* **103** (1999) 1140.
52. "Statistical Thermodynamics of Bond Torsional Modes", Y.-Y. Chuang and D. G. Truhlar, *J. Chem. Phys.* **112** (2000) 1221. **121**, (2004) 7036(E), **124**, (2006) 0000(E).
53. "Nonequilibrium Solvation Effects for a Polyatomic Reaction in Solution", Y.-Y. Chuang and D. G. Truhlar, *J. Amer. Chem. Soc.*, **121** (1999) 10157.
54. "Improved Algorithm for Corner Cutting Calculations", A. Fernandez-Ramos and D. G. Truhlar, *J. Chem. Phys.* **114** (2001) 1491.
55. "Interpolated Algorithm for Large-Curvature Tunneling Calculations of Transmission Coefficients for Variational Transition State Theory Calculations of Reaction Rates", A. Fernandez-Ramos, D. G. Truhlar, J. C. Corchado, and J. Espinosa-Garcia, *J. Phys. Chem. A* **106** (2002) 4957.
56. "Tests of Potential Energy Surfaces for $\text{H} + \text{CH}_4 \leftrightarrow \text{CH}_3 + \text{H}_2$: Deuterium and Muonium Kinetic Isotope Effects for the Forward and Reverse Reaction," J. Pu and D. G. Truhlar, *J. of Chem. Phys.* **117**, (2002) 10675-10687.
57. "Efficient Molecular Mechanics for Chemical Reactions: Multiconfiguration Molecular Mechanics using Partial Electronic Structure Hessians," H. Lin, J. Pu, T. V. Albu, and D. G. Truhlar, *J. Phys. Chem. A* **108**, (2004), 4112-4124.

58. "Statistical Thermodynamics of Bond Torsional Modes. Tests of Separable and Almost-Separable Approximations Applied to H₂O₂, HOOD, D₂O₂, H₂¹⁸O₂, H₁₈OOH, D₁₈OOH, and H₁₈OOD," V. A. Lynch, B. A. Ellingson, S. L. Mielke, and D. G. Truhlar, *J. Phys. Chem. A*, submitted Feb. 28, 2006.
59. "Variational Transition State Theory with Multidimensional Tunneling," A. Fernandez-Ramos, B. A. Ellingson, B. C. Garrett, and D. G. Truhlar, in *Reviews in Computational Chemistry*, edited by K. B. Lipkowitz (Wiley-VCH, Hoboken, NJ), in press.

22. ERRATA

Note: The errata listed in Subsections 22.B and following subsections for previous manuals are not intended to be complete (unlike the revision history which is intended to be complete), but these errata include some corrections to errors that may have been particularly confusing to users of older versions. The error is listed under the last version in which it occurred.

22.A. Errata for "Generalized Transition State Theory" [Ref. 5, Sect. 21]

This section contains all known errata to the GTST book chapter (Ref. 5 in Sect. 21) by D. G. Truhlar, A. D. Isaacson, and B. C. Garrett.

1. p. 71, line 1: $V[x_{i\gamma}]$ should be $V([x_{i\gamma}])$.
2. p. 71, third line after eq. (6): "coordinate(s)" should be "coordinate (s)".
3. p. 77, Fig. caption line 3: "1 molecule and per centimeter, anharmonicity" should be "1 molecule per centimeter, and anharmonicity".
4. p. 78, second line after eq. (15): $\nabla V(R)'_j$ should be $\nabla V(R'_j)$.
5. p. 81, eq. (31): $x_{i'\gamma'}$ should be $x_{i'\beta'}$.
6. p. 82, eq. (35), line 1: $(m_A + m_B)^{-1}$ should be M^{-1} where M is the total mass.
7. same equation, line 2: insert " μ_f " in numerator.
8. p. 83, line 2: $Q^{\text{GT},0}(T,s)$ should be $Q^{\text{GT}}(T,s)$.
9. p. 83, line 8: $\Phi_V^{\text{R}}(T)$ should be $\Phi_C^{\text{R}}(T)$.
10. p. 83, second line from the bottom: Δ should be ΔG .
11. p. 84, first line after eq. (42): ΔG_C^{GT} should be $\Delta G_C^{\text{GT},0}$.
12. p. 88, line 16: "accurage" should be "accurate".
13. p. 94, eq.(85c): add " β " before "V" in first exponential.
14. p. 95, line after eq. (91): "MED" should be "MEP".

15. p. 97, eq. (98) should read

$$B_{mF} = - \sum_{i=1}^N \sum_{\gamma} \frac{d \left\{ \text{sign}(s) \right\} v_{i\gamma}(s)}{ds} L_{i\gamma,m}^{\text{GT}}(s)$$

or

$$B_{mF} = - \sum_{i=1}^N \sum_{\gamma} \frac{dv_{i\gamma}(s)}{d|s|} L_{i\gamma,m}^{GT}(s)$$

16. p. 97, second line after eq. (98): $B_{mF}(s)$ should be $-B_{mF}(s)$.
17. p. 97, third line after eq. (98): $[1 - \kappa(s) \cdot \mathbf{Q}(s)]^2$ should be $[1 - \kappa(s) \cdot \mathbf{Q}(s)]^2$.
18. p. 97, second and third lines after eq. (100): " μ " should not be a subscript.
19. p. 97, third line after eq. (101): $[\hbar/\mu\omega_m(s)]^{1/2}$ should be $[\hbar/\mu\omega_m(s)]^{1/2}$.
20. p. 101, in the last sentence, the gradient in the product channel (or reactant channel) should be multiplied by -1.
21. p. 102, line 7: "is $x(s)$ " should be "is $\dot{x}(s)$ ".
22. p. 102, fourth and fifth lines after eq. (126): " \mathbf{x} " should be " $\dot{\mathbf{x}}$ " in three places.
23. p. 102, line after eq. (127b): "B" should be β in two places, and "B/2" should be $\pi/2$. Thus below eq. (127), the inequality should be $0 < \beta < \pi/2$.
24. p. 102, line after eq. (128): "XB" should be "A + XB".
25. p. 102, second line after eq. (128): x^R should be \dot{x}^R .
26. p. 103: eq. (130) should be the special case of eq. (157) with $\tilde{\alpha}_0 = 1$, i.e., insert $\sin(\chi(\tilde{\alpha}_0 = 1, s_0))$ before $d\tilde{s}_0$.
27. p. 104, fifth line after eq. (131): "mass-weighted coordinates \mathbf{x} the" should be "mass-scaled coordinates \mathbf{x} ; the".
28. p. 104, second line after eq. (133): \tilde{s}_1 should be \tilde{s}_0 .
29. p. 104, second line after eq. (134): "zt" should be "at" and "normal" should be "generalized normal coordinates".
30. p. 104, third line after eq. (134): "coordinate(s)" should be "coordinate (s)".
31. p. 104, eq. (135) and following: notice that $q_{\perp,m}$ depends on \tilde{s}_1 as well as \tilde{s}_0 .
32. p. 106, eq. (144): insert a minus sign before the left-hand side because every term in the sum is negative.
33. p. 106, eq. (146): the condition $Q_m < t_m$ should be replaced by eq. (143).
34. p. 106: eq. (147) should be the special case of eq. (159) with $\tilde{\alpha}_0 = \tilde{\alpha}_1 = 1$.

35. p. 110, line 6: 147 should be 148a.
36. p. 117, line 3: "reactant" should be "reactants".
37. p. 117, second line from bottom: $p_2 : p_1$ should be p_2/p_1 .
38. p. 118, second line from bottom: ":" should be "/" in two places.
39. p. 121, line below eq. (209): " $N_m - 1/2$ " should be " $(N_m + 1/2)/h$ ".
40. p. 121, eq. (210): $1/4\pi h$ should be π .
41. p. 123, second line above eq. (217): "satte" should be "state".
42. Ref. 58, line 1: "B.D." should be "B.C."
43. Ref. 71, line 2: "Vol. 2" should be "Vol. 4 of Modern Theoretical Chemistry".
44. Ref. 192, line 2: "Phyus" should be "Phys".
45. Ref. 231, line 2: "52" should be "62".

22.B. Errata for "Ab Initio Reaction Paths and Direct Dynamics Calculations" [Ref. 20, Sect. 21]

Several errors were detected in the tables of this paper, and we were unable to reproduce some of the values reported in them:

1. p. 5116, line 4 in Table III.c: " $\Delta H^\circ -3.73$ " should be " $\Delta H^\circ +5.07$ ".
2. p. 5117, Table IV: We believe the correct results for the transmission coefficients for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ are as follows:

TEMP (K)	\neq/W	ICVT/MEPSAG	CVT/CD-SCSAG
298	8.28	2.41E+03	1.85E+05
400	5.04	5.61E+01	9.55E+02
667	2.45	3.40E+00	1.08E+01
1000	1.65	1.68E+00	2.78E+00
1550	1.27	1.24E+00	1.51E+00

3. p.5118, Table V: We believe that the correct results for the forward rate coefficients (in $\text{cm}^3\text{molecule}^{-1}\text{s}^{-1}$) at the UHF/STO-3G level are as follows:

TEMP (K)	TST	ICVT/MEPSAG	CVT/CD-SCSAG
298	1.02E-31	2.44E-28	1.87E-26
400	7.98E-27	4.00E-25	7.55E-24
667	3.96E-21	1.34E-20	4.25E-20
1000	3.09E-18	5.15E-18	8.50E-18
1550	4.72E-16	5.77E-16	7.06E-16

The data reported in these tables correspond to a POLYRATE 7.0 run using a reaction path generated with GAMESS (ES1 integrator, mass-weighted Cartesian coordinates, stepsize = 0.005 amu^{1/2} bohr) and forces and Hessians calculated every 0.05 amu^{1/2} bohr with GAUSSIAN94 without reorientation of the geometries.

22.C. Errata for "Interpolated Variational Transition State Theory: Practical Methods for Estimating" [Ref. 26, Sect. 21]

The data for the electronic partition function of the Cl atom was incorrect. In the calculations reported in this article the degeneracy g of the ground state of Cl was taken as 2, and the excited state was ignored, instead of the correct electronic partition function $(4 + 2e^{-\epsilon/RT})$ where ϵ is the energy of the first excited state for Cl. Thus all reported rate constants for the Cl + CH₄ reaction are too large by a factor between 2 and 3.

22.D. Errata for manual of version 1.1

1. p. ii: Subsections 1.B and 1.C should begin on pages 1.5 and 1.6, respectively.
2. p. 1.5, line 1: "PROJECT" should be "PROJECT";.
3. p. 2.15, eq. (36b): a minus sign in front of the third term should be a plus sign.
4. p. 4.1, line 4: "heirachy" should be "hierarchy".
5. p. 5.17, third row: "RPHFJT" should be "RPHFIT".
6. p. 7.8, line 2: "3*NATOM-6 distances" should be "3*NATOM-6 derivatives".
7. p. 7.13, line 19: "the direction" should be "the other direction".
8. p. 7.16: POLYRATE version 1.1 allows a maximum of 10 atoms; however if LGS(23) is not zero, the number of vibrational frequencies in a reactant or product should not exceed 22.
9. p. 7.16, fifth line from bottom: "28" should be "53".
10. p. 7.17, line 3: "skip to record 55" should be "return to record 32".
11. p. 8.2, line 6: "LGS(9) = 11" should be "LGS(9) = 1".
12. p. 8.38, line 13: "temperature" should be "temperatures".
13. p. 8.50, line 4: "long-time" should be "long-line".
14. p. 8.63, line 23; 8.67, line 23; 8.71, line 27: ICODE = 4 should be specified for the starting geometry (also see 7.10, record 16 for input format).

22.E. Errata for manual of version 1.5

1. p. iii, line 8: "File 20" should be "File 30".
2. p. 2.6, line 18: Reference for the Brent method: W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, "Numerical Recipes" (Cambridge University Press, Cambridge, 1986), pg. 283.
3. p. 2.11, eq. (34): "D" should have an overbarb [this vector D should not be confused with the lowest dissociation energy D mentioned in the explanation of LGS(15)].
4. p. 2.12, eqs. (42) and (43): "t" should not be a subscript.
5. p. 2.13, eq. (46a): " $=x,y,z$ " should be " $\alpha,\beta,\gamma = x,y,z$ ".
6. p. 2.13, eq. (46b): " $P_{i\gamma,i'\nu}$ " should be " $P_{i\gamma,i'\gamma}(s)$ ".
7. p. 2.13, line 19: "asymmetric" should be "antisymmetric".
8. pages 5.4, 5.5: On both pages the chart should include the fact that the subroutine F1DIM calls SURF.
9. p. 7.1: In the LGS(3) section, the first " $LGS(1) = 0$ " should be " $LGS(1) \neq 0$ ".
10. p. 7.11: for record 15, "(6F10.6)" should be "(20I3)".
11. p. 7.12: ICODE is not read in SADDLE in record 16.
12. p. 7.10: "Gauss-Legendre" should be "Gauss-Laguerre".
13. p. 7.16: record 31 is read only if $LGS(3) > 0$.
14. p. 8.52: page number appears on the following page.

The authors are grateful to Ronald J. Duchovic, then of NASA Ames Research Center, for finding several of these errors.

22.F. Erratum for manual of version 1.6.2

1. p. 47, option 0 for LGS(15): should be "set $LGS(15) = 0$ if $LGS(5) = 0$ ".

22.G. Errata for manual of version 1.7

1. p. 50, line 6: "LGS2" should be "record 1A".
2. p. 50, line 11 should be moved above line 8.
3. p. 50, line 12: add "LGS2(2-39):" to beginning of line.

4. p. 50, line 21: "ANTL" should be "ANTLR".
5. p. 70, line 16: "skip to record 28" should be "skip to record 53".

22.H. Errata for manual of version 2.6

1. p. 44, line 26: "NG" should be "NGS0".
2. p. 44, line 29: "NGS0" should be "NG".

22.I. Erratum for manual of version 2.7

For the manuals of versions 2.6 and 2.7, the output files for units 15 and 16 of test run oh3tr3 were misprinted; this has been corrected in later manuals. This error did not involve the code.

22.J. Errata for manual of version 4.0

1. p. 4-17, line 44: "caused" should be "would cause".
2. p. 5-1, lines 23, 25, and 34: "dattim.for" should be "dattim.ibm".
3. p. 11-19, line 6: "DEL/6" should be "DEL".
4. p. 12-8, line 28: "dattim.for" should be "dattim.ibm".
5. p. 12-12, line 14: "dattim.for" should be "dattim.ibm".
6. p. 14-2, line 17: "dattim.for" should be "dattim.ibm".
7. p. 16-1, line 57: "fu31-fu35" should be "31-35".

22.K. Errata for manual of version 4.1

1. p. 4-16, line 1: item 15 in the revision history of version 3.0.7 was accidentally missing.
2. p. 4-18, lines 32-33: "An option to select" should be "A discussion of".
3. p. 11-5, line 18: "Morse model used" should be "Morse model not used".

22.L. Errata for manual of version 4.5.1

1. p. 11-32, lines 21-23: "If ICODE = 2, read record 5." should be deleted. Record 5 in unit fu30 should always be read if the electronic structure input file option is selected [LGS(30) = 1].
2. p. 13-11, line 37: "ch5j1tr6.fu1" should be "ch5j1tr5.fu1".

22.M. Errata for manual of version 5.0

1. p. 11-35, line 40: "Loop over records 32 through 54 NPTS times." should be "Loop over records 32 through 54 NPTS times, from the lowest (or most negative) value of s to the highest value of s ."
2. p. 13-14, line 8 was accidentally missing.

22.N. Errata for manual of version 5.0.1

1. p. 13-38, lines 20-23: The two sentences "The MEP is followed ..." and "A generalized normal mode ..." should be deleted.

22.O. Erratum for manual of version 5.1

1. p. 6-4, line 1 should be "and in Refs. 20 and 25 in Sect. 20."

22.P. Errata for manual of version 6.0

1. p. 125, line 1, part of the text is missing. The complete text should be:
 $N3 = 3 * (\text{number of atoms in total system})$. Ordering is the same as that specified in the ATOM keyword of the FORTRAN unit fu5 data.
2. p. 235, line 7, the last sentence should be "The timing under IRIS is for the Indigo R4000 machine."

22.Q. Errata for manual of version 6.5

1. p. 85, line 6: "...into four categories: basic, geometry optimization, normal mode analysis, and anharmonicity" should be: "...into two categories: basic and anharmonicity."
2. p. 88, line 4: "The is only used ..." should be: "The ANTLR keyword is only used...."

3. p. 88, the keyword CONVERGE should be deleted.
4. p. 89, line 4: "...R_e, β and D_e," should be: "..., R_e, D_e, and β,"
5. p.105, line 6: "keywordXE ?vscale keyword: definition?}" should be "keyword".
6. Section 11.B, page 124: In five places where the save grid is mentioned, the correct explanation involves the basic input grid.
7. p. 130, line 15: "...if LOPT(7) ≥ 0, skip records 57 through 65" should be: "if LOPT(7) ≤ 0, skip records 57 through 65."
8. p. 130, line 16: "...if LOPT(8) ≥ 0, skip records 57 through 65" should be: "if LOPT(8) ≤ 0, skip records 57 through 65."
9. p. 197, line 2: "...zero order VTST-IOC calculation...." should be: "...zero-order IVTST calculation...."

22.R. Errata for manual of version 6.6

1. p. 54, line 12: "...we assume that 2 and $s_2 \geq 0$, ..." should be "...we assume that $s_1 < s_2 < s_3$ and $s_2 \geq 0$, ..."
2. p. 137, line 21: "For reactants, if LOPT(7) ≥ 0, skip records 57 through 65." should be "For reactants, if LOPT(7) ≤ 0, skip records 57 through 65."
3. p. 129, line 11: "UNITS is a keyword that specifies the units for the geometry, gradients and Hessian matrix: 0—The geometry will be in bohrs, and both the gradient vectors and the Hessian matrix will be in non-mass-scaled atomic units. 1—The geometry will be in Å, and both the gradients and the Hessian matrix will be in mass-scaled atomic units (default)." should be "UNITS is a keyword that specifies the units for the geometry, gradients, and Hessian matrix: 0—The geometry will be in bohrs, and both the gradients and the Hessian matrix will be in mass-scaled atomic units (default). 1—The geometry will be in Å, and both the gradient vectors and the Hessian matrix will be in non-mass-scaled atomic units. "
4. The definition of the ENERXN keyword was wrong and has been corrected for version 7.0.
5. See also item 7 in the version 7.0 revision history.

22.S. Errata for manual of version 7.1

1. p. 32, line 22: 1 should be 0.
2. p.57, line 27: ATOMS should be GEOM

22.T. Erratum for manual of version 7.2

1. The discussion of GAUSSIAN options in section 9.B. was incomplete.

22.U. Erratum for manual of versions 7.8 – 7.9.1

1. The options for the GEOMUNIT keyword of the GENERAL section of the fu5 input file were missing.

22.V. Erratum for manual of versions 8.0

1. The description of the NOPROJECT keyword of fu5 input was incorrect. NOPROJECT turns off projection at equilibrium structures but not at saddle point.

23. APPENDIX: POLYRATE REVISION HISTORY

23.A. Computer program revision history

This section contains the revision history subsequent to version 1.1 (August 1987), which was the first distributed version. First and second level revisions (e.g., 2.0 or 2.1) denote enhancements; third level revisions (e.g., 1.5.1) denote bug fixes.

Version 1.5 (September 1988)

1. Projection operators suitable for reactions on solid surfaces are included as well as those for gas-phase nonlinear and linear complexes. An option for solid-state and gas-solid-interface reactions is added employing these projection operators. This can also be used for embedded cluster calculations.
2. Variable dimensions are used, removing the previous restriction to 10 or fewer atoms.
3. An extra option, $LGS(13) = 2$, for unit 5 input is added.
4. An option, $LGS(3) = 3$, is added to allow calculation of the MEP without calculating frequencies of generalized transition states.
5. The less useful anharmonicity options are removed. All methods except harmonic, Morse I/II/III, quadratic-quartic, and WKB are removed.
6. The Newton-Raphson method is improved to use a variable search step determined by the Brent method as discussed in the section "Reactant and product properties" in this ASCII manual.
7. The methods for calculating the MEP are updated. All previous methods except Euler and ES1 are removed. The Page-McIver method is added.
8. The output format is improved.
9. A list of all known errata and typographical errors in the GTST book chapter is included in the manual.

Version 1.5.1 (December 1988) (Manual fix)

1. Test runs W and X were rerun with smaller DELSV to get converged results.

Version 1.5.2 (February 1989) (Bug fixes)

1. In subprogram NOROUT:

```
      if (... pmode.eq.false)
```

is non-standard FORTRAN and was changed to

```
      if (... .not.pmode).
```

2. In subprogram PATH:
When the option of scaling V_{MEP} is chosen, the vibrational frequencies of the reactant and product which correspond to the reaction coordinate are also scaled appropriately by a factor of $\text{SQRT}(VFAC)$. If $VFAC$ is not zero or one, the indices of the reactant and product vibrational frequencies being scaled are read on unit 5.
3. In subprogram RATE:
For gas-phase reactions which involve only one reactant or product, the relative translational partition function for the forward or reverse rate, respectively, is not calculated unless an external field is present. This affects the reverse rate constants of test runs P-R.
4. In subprogram REACT:
 $\text{IF}(\text{LGS}(4) \text{ .EQ. } 4)$ is corrected to $\text{IF}(\text{LGS}(6) \text{ .EQ. } 4)$. This affects the free format reading of the variables $ICODE$ and $IATOM$ in the case of a unimolecular reaction.
5. Test run W is removed because unstable results are caused by the nature of the potential energy surface.
6. In subprogram OVERLP:
Frequencies and normal mode eigenvectors are also switched when two modes switch. (Note this will introduce a small difference in the SCSAG transmission probability compared to previous results for a given set of numerical parameters, but both algorithms should converge to the same result when the step size is decreased.)
7. In function DGAMMA:
An error that affected the operation of GAUSSQ when it was called for generalized Gauss-Laguerre quadrature with $\alpha > 2$ or for Jacobi quadrature with $\alpha > 2$, $\beta > 2$, or $\alpha + \beta > 1$ has been corrected, even though POLYRATE does not require these cases.
8. In subprogram RESTOR:
Errors in reading information about MEP #2 and calculating the reverse rate constants when $\text{LGS}(8) = 3$ have been fixed.

Version 1.6 (July 1989)

1. Two subprograms, FIOpen and FICLOSE, were added to open and close all files. These were added to make the program more portable, in particular to run under the CTSS operating system. (Note: these statements are commented out in distribution versions from 1.6 to 3.9.3, and they are reorganized and supported in versions 4.0 and later)
2. A 'call link' statement has been added to MAIN to make the program usable on a Cray computer running CTSS. Users of different machines will need to add a stub subprogram called link that just returns when called.
3. The Newton-Raphson search convergence criterion has been modified in subprogram NEWT to search until the norm of the gradient is small. In previous versions the search was terminated when changes in the energy from iteration to iteration became small.

4. The total vibrational partition functions Q_v are calculated in terms of $\ln(Q_v)$ to avoid underflow in medium-size systems, an option has been introduced to allow Q_v 's to be multiplied by a factor of $10^{\text{LGS}(37)}$ to avoid underflow in large systems.
5. Option LGS(36) has been added to compute the geometry of the next point on the MEP from the electronic structure input data.
6. The reduced mass for mass-scaled coordinates is read in record 5, and relative translational reduced masses for bimolecular reactions are now calculated internally by the program.

Version 1.6.1 (November 1989) (Bug fixes)

1. In subprogram CENTER:
For a polyatomic molecule with a nonlinear geometry, the new version correctly prints out FMOM, which is the product of the three principal moments of inertia, whereas versions 1.1-1.6 incorrectly printed out the determinant of the moment of inertia tensor without the factor of μ^3 in eq.(47) of the GTST book chapter. This printout error did not affect the rate calculations.
2. In subprogram NORMOD:
For the Page-McIver reaction-path method, unnormalized derivatives of the potential are stored in the array DXP as for the Euler method. This corrected a printout error in versions 1.1-1.6.
3. In subprogram OPTION:
The print statement reflecting the action carried out when $\text{LGS}(17) \neq 0$ has been changed to say that the reverse rate constant will be multiplied by $10^{\text{LGS}(17)}$. This required deleting a minus sign in FORMAT 1340. This also corrected a printout error in versions 1.1-1.6.
4. The manual has been corrected to include subprogram NEXTPT which was added in version 1.6. Furthermore, this subprogram has been corrected so that the geometry of the next point on the MEP can be calculated correctly from RPH input data. In addition, a bug for option $\text{LGS}(36) \neq 0$ is fixed in subprogram RPHRD2, where the generalized normal mode analysis is now performed only if $\text{LGS}(36) = 0$.
5. In the evaluation of turning points for a symmetric adiabatic barrier, the tunneling energies should be selected on a grid from the threshold energy up to the maximum of the adiabatic barrier. The adiabatic barrier maximum obtained by spline fit is always greater than or equal to the largest value of the adiabatic potential on the Hessian grid. Thus the barrier maximum for determining the grid of tunneling energies should be obtained from the fit, and not just set to the largest value on the save grid. Versions 1.1-1.6 used the maximum value from the save grid, but version 1.6.1 uses the value from the fit. This required changes in four subprograms: KAPVA, PSAG, PSATP, and PSATP2.
6. In subprogram SPL1D1:
A non-standard FORTRAN statement involving a transfer into a DO loop is corrected by rearranging the subprogram.

7. The POLYRATE manual has been corrected and improved. Especially important corrections occur on pages 16, 21, 24, and 25. In addition, descriptions of subprograms INTFNC, PITZ, NEXTPT, and STABLE have been added to the section "Description of FORTRAN subprograms".

Version 1.6.2 (February 1990) (Bug fix)

1. In subprogram OVERLP:
An error introduced in version 1.5.2 in the way that frequencies and eigenvectors are switched at crossings and/or avoided crossings of vibrationally adiabatic potential curves has been corrected so that the effective mass for an SCSAG tunneling calculation is evaluated correctly.

Version 1.6.3 (March 1990) (Bug fixes)

1. In subprogram RATE:
The size of array QVSV, which is only used in the subprogram RATE for vibrational partition functions of reactants and products, is increased from $N3TM = 3 \times NATOM$ to $N6TM = 6 \times NATOM$, since $QVSV(N3TM)$ may not be sufficiently large for some reactions.
2. In subprogram PATH:
The format of the input data for the variable NST0, which is the number of steps of size DEL1 to be taken in following the reaction path away from the saddle point, was incorrect from versions 1.5 to 1.6.2, and is correct in version 1.6.3. (Note: the format was correct in version 1.1.)

Version 1.7 (March 1990)

1. All input information read from unit 5, except the routing and output options, is now read in and checked by two new subprograms READ5 and REDGEO at the beginning of the calculation. Previously the input was read in many different subprograms and at various stages of the calculation. A new input option, LGS2(1) = 1, is used to request that the program read the input from unit 5, write it on unit 6, and then stop. This is useful for checking input data interactively before submitting a job to a batch queue. This consolidation of input involved changes in subprograms MAIN, DIATOM, FINOUT, PATH, POLYAT, RATE, RESTOR, and SADDLE.
2. Another new option, LGS(29) = 2, is now added for calculating the reaction-path curvature by the method of Page and McIver from the gradient and the Hessian and its eigenvectors. This option involved changes in subprograms BCALC, INTPM, and OPTION. Another test run Z is added to show the use of this option.
3. The normalization factor of the gradient along the reaction path is computed and stored in variable DXNORM of the IRCCM common block which has been rearranged in subprograms INTFNC, INTPM, IRCX, NEXTPT, and NORMOD.

Version 1.7.1 (April-July 1990) (Bug fixes)

1. In subprogram PATH:
The conversion factor for energy, CKCAL, has been corrected so that it is consistent with the value used elsewhere in the code.

2. In versions 1.5-1.7, the frequencies written on the short output file for the reaction-bottleneck properties were truncated to integers rather than rounded in the subprogram FINOUT, and the option, LGS(9) = 0, for calculation of the rate constants without tunneling calculations, did not work because of the undefined array BETA. Both bugs are corrected in version 2.0.

Version 2.0 (July 1990)

1. A new type of semiclassical tunneling calculation is added, in particular the large curvature ground state method, version 3 (LCG3), which is especially appropriate for reactions with large curvature of the reaction path. The implementation of the LCG3 method involved adding 22 new subprograms and functions as well as modifications in 17 other subprograms of POLY1 and 6 subprograms of POLY2. Notice that the LCG3 calculation is available only for the harmonic approximation and for ground-state products as well as ground-state reactants in version 2.0.
2. A new option, LGS2(2) = 1, is now added for explicitly projecting out the translational and rotational motions at stationary points. In some cases this results in more accurate values for nonzero frequencies. This option involved changes in the subprograms: OPTION, CENTER, NORMOD, POLYAT, and SADDLE.
3. An algorithm is added for finding turning points when the tunneling energy is greater than the largest value of the adiabatic potential on the save grid, but less than the maximum of the spline fit. This is accomplished by interpolation between the maximum of the spline fit and the largest values on the grids using the cubic polynomial corresponding to that section of the spline fit. Previous versions did not handle this situation correctly, although the quadrature grid was seldom dense enough for the problem to occur. Installing the new algorithm involved changes in the subprograms: PSAG, PSATP, PSATP2, BOLTZ, FINOUT, FITMAX, VTMUSN, and FIVPT.
4. A new type of input has been introduced for the step sizes of the Hessian and save grids. This enhancement involved changes in the subprograms: INTPM, PATH, READ5, RESTOR, and NEXTPT. See the section on "Reaction-path following methods" in this manual.
5. For version 2.0, the two subprograms INTERP and LINMIN have been renamed INTRPL and LINMN, respectively, so that the interface with MOPAC can be made without subprogram name conflicts.
6. Two new test runs have been added, and all the previous test runs have been modified to test the new enhancements. A new, more convenient convention was introduced for naming test runs. The two new test runs, ch5tr2 and h3tr2, involve LCG3 calculations for two different reactions; the former one uses the restart input file option, and the latter computes the reaction path information from an analytic potential energy surface. The LCG3 option has also been turned on for test runs ch5tr1 and h2nitr1, and test runs oh3tr3 and ch5tr4 have the new stationary-point projection option turned on. In the majority of the test runs, the Hessian and save grid step sizes are now calculated from input values of INH and INM, which specify them as integer multiples of the gradient grid step size. The gradient step and Hessian step size for most test runs have been increased to be closer to optimum choices for 1% accuracy.

7. For version 2.0, all test runs are verified on four different computers, and the system-dependent subprograms, in particular, DATTIM.VMX or DATTIM.UNIX, have been moved to separate files from the rest of the source code. The code is now distributed with sample command files for both VMS and UNICOS.
8. Various changes are made to allow the convenient creation of MORATE from POLYRATE.

Version 2.0.1 (September 1990) (Bug fix)

1. In subprogram REDGEO:
An error introduced in version 1.7 for reading input for a unimolecular reaction has been corrected so that the information is stored in the correct variable.

Version 2.1 (October 1990)

1. In order to make the code more portable, all statements containing non-ANSI FORTRAN 77 that are flagged by the cft77 compiler on the Cray-2 have been modified to conform to the FORTRAN 77 standard. Also, all common blocks have been made to have the same length every time they occur, and subprogram FICLOSE has been renamed FCLOSE. The code is written to execute in double precision so it runs with REAL*8 floating point variables on VAX, Sun, Silicon Graphics, and IBM computers. On the Cray, REAL*8 is single precision so the code should be compiled with the disable precision (-dp) option.
2. All common blocks and statements used only in MORATE have been commented out with C*M in columns 1-3 so that the memory and execution time of POLYRATE itself are as small as possible.
3. A symmetric reaction path option, LGS(3) < 0, has been added, and a related test run, h3tr3, has been provided for this option. For the current version, this option is available for calculations with all transmission coefficients except LCG3.
4. A warning was added to the manual about using LCG3 with only ground-state final states. Restricting the final state of the tunneling process to the ground state is not always appropriate.
5. A large number of typos were corrected in the manual, and various clarifications were added.

Version 2.1.1 (November 1990) (Bug fix)

1. In previous versions, the frequencies written on unit fu6 were truncated to integers rather than rounded in subprograms PATH and RESTOR. This bug is fixed in version 2.1.1.

Version 2.5 (November 1990)

1. The main changes in version 2.5 are all related to portability. A single version of POLYRATE now runs correctly on all eight computers mentioned in the section labeled "computers and operating systems on which the code has been checked". The full 20-run test suite executes successfully on all eight computers. All non-ANSI usage that was uncovered in the course of these tests has now been converted to standard FORTRAN 77 and all non-portable code has been made portable. The

modifications include the declarations of external functions DIGAMMA and ERF in subprograms CLASS and ZIGAMA respectively, a correction in subprogram PSATP2 to avoid possible floating point errors, and the correction of all statements containing non-ANSI FORTRAN 77 in the analytic potential energy surfaces used for the 20 test runs. For example, the subprogram TRIPLET in CH5POT.FOR, which is used in the seven test runs ch5tr1 to ch5tr7, has been renamed TRIPLE, and the variables NCOUNTA and NCOUNTB in H2NIPOT.FOR which is used in test run h2nitr1 have been renamed NCOUNA and NCOUNB, respectively.

2. A large number of precision changes were made to many subprograms to make sure that the code is executed completely in 64-bit precision. The five non-Cray machines on which the code has been tested have 32-bit words, and explicit use of double precision typing ensures REAL*8 precision on these machines. On the Cray computers, the word length is 64 bits, so single precision corresponds to REAL*8. Thus on these machines we use the disable precision (-dp) compiler option so the code again runs with REAL*8 precision. But the non-standard REAL*4 and REAL*8 type declarations do not actually appear in the program.
3. The Boltzmann constant has been replaced with the more accurate value 3.166830×10^{-6} a.u. in subprogram RATE. Thus, for some test runs the rate constants might be slightly different from the ones in previous versions. The differences should be less than 0.2%.
4. The code is now distributed with sample command files suitable for its execution on all eight computers on which it has been tested. Actually only seven sets of command files are used since the same set of command files can be used for the Cray-2 and Cray X-MP.

Version 2.5.1 (February 1991) (Bug fix)

1. A correction has been made in the manual only. The correction concerns allowed values of other parameters that may be used with the electronic structure input option, LGS(30). The combination of LGS(30) = 1 and LGS(6) = 0 is allowed; however LGS(30) = 1 must be used when LGS(36) = 1. The statement just below LGS(30) = 1 on page 59 in the manual of version 2.5 is corrected.

Version 2.5.2 (March 1991) (Bug fixes)

1. In subprogram BCALC:
A correction has been made for the storage of the turning points of generalized normal modes with imaginary frequencies for LCG3 calculations. In versions 2.0-2.5 these were set to zero. Now they are set more appropriately to infinity, i.e., a very large value (10^{10} bohr). Note that this modification only affects LCG3 calculations, not SCSAG calculations. (For SCSAG calculations the turning points of modes with imaginary frequencies are set equal to zero.)
2. In subprogram IRCX:
The Euler stabilization algorithm now performs correctly, even when the stabilization step is omitted. This correction supplies a previously missing statement, calculating the reaction coordinate s , when the stabilization procedure is skipped. This error existed in versions 1.0-2.5.

3. In subprogram NEXTPT:
An extra READ statement, which was not deleted when all READ statements were moved in version 1.7 and which caused a fatal error by reading beyond the end-of-file mark whenever the option for the calculation of the geometry of the next point on the MEP from electronic structure input data was selected [LGS(36) = 1], was removed.
4. In subprogram OPTION:
An error allowing the user to select options LGS(30) \neq 1 and LGS(36) = 1 simultaneously was removed. (If LGS(36) = 1, then the only allowed value of LGS(30) is 1.) This error existed in versions 1.6-2.5.
5. In subprogram REDGEO:
An error in the printout of the initial geometry for the optimization of the saddle point has been corrected. This error existed in versions 1.7 - 2.5.
6. A correction has been made to an error introduced in the manual at the time version 1.7 was prepared. The manual stated that the normalization factor of the gradient along the reaction path is computed and stored in variable DXNORM of the IRCCM common block, but actually it is correctly stored in DXMAG.
7. A correction has been made in the manual concerning the input format for variable NST0, the number of steps of size DEL1 to be taken in following the reaction path away from the saddle point, when LGS(13) = 1. Before, it was given in the manual as (.../,20X,I5), but the program actually reads it as (.../,30X,I5). This error existed in the manual since version 1.5.
8. Two subprograms, PITZ and STABLE, that are no longer used, have been removed, and common block MINCOM has been adjusted to have the same length in all subprograms.
9. In function VPART:
The treatment of imaginary frequency generalized normal modes is made consistent with the procedures described in the section "Imaginary frequency modes" in this manual.

Version 2.6 (March 1991)

1. A new method for calculating the initial step from the saddle point with analytical potential energy surfaces is now included. This method bases the first step on a cubic expansion of the potential energy surface, instead of a quadratic one. The new method can be selected through option LGS(31) and is responsible for changes in subprograms INTEUL, INTPM, OPTION, PATH, and READ5.
2. The Page-McIver integration method is now able to use NST0 \neq 0 in the unit 5 input, which allows it to take smaller steps than the pre-selected gradient step size in the vicinity of the saddle point. This enhancement involved changes in subprograms INTPM and PATH.
3. The stabilization criterion for the ES1 integration method, which is equal to the vector difference between successive normalized gradients, can now be selected by the user. The old value, 10^{-8} , remains as the default. This improvement involved changes in subprograms IRCX, PATH, and READ5.

4. The calculation of the reaction-path curvature by the method of Page and McIver from the gradient and the Hessian and its eigenvectors—option LGS(29) = 2—has been optimized to avoid unnecessary storage. This optimization involved changes in subprograms BCALC, INTFNC, INTPM, IRCX, NORMOD, and PATH.
5. In subprogram OPTION:
Three STOP statements were replaced with LEXIT = .TRUE. statements, to allow the code to flag all incorrectly selected options before stopping. The STOP statements were introduced in version 2.0.
6. The test runs oh3tr1 and oh3tr3 were modified to incorporate the new enhancements. In particular, test run oh3tr1 now uses the Page-McIver method with the cubic starting algorithm and performs five steps of size DEL1 in the vicinity of the saddle point before taking steps of size DEL. Furthermore, test run oh3tr3 now also uses the cubic starting algorithm, and it uses the Euler stabilization method along the reaction path with a bigger stabilization criterion, equal to 10^{-3} . Differences in rate constant results between the old and new test runs in both cases do not exceed 1%.

Version 2.6.1 (June 1991) (Bug fixes)

1. In subprogram EXTRAP:
Minor modifications have been made in order to incorporate changes in subprogram READ5 related to the extrapolation procedure. This corrects wrong statements which existed since version 1.0.
2. In program MAIN:
In versions 1.0-2.6, the energy and gradient were not computed correctly at the starting point when no saddle point calculation was performed. This is now corrected.
3. Also in program MAIN:
Another change has been made in MAIN to correct an error in restart runs when POLYRATE is used in conjunction with MOPAC as MORATE. This error existed since version 1.6.
4. In subprograms LCNWTN and LCG3:
A bug in the root search to determine the unique s value corresponding to a point along the LCG3 tunneling path has been eliminated. As explained in pages 105-106 of the GTST book chapter, the value of s should be a continuous function of the progress along the tunneling path. In versions 2.0-2.6 of POLYRATE, the root search was coded incorrectly to find the smallest (i.e., most negative or least positive) root on the reactant side and the greatest root on the product side. This is corrected in the new subprograms so that the iterations start with the value of s for the previous point on the tunneling path. Thus the iterations should converge to the continuous root.
5. In subprogram OPTION:
In versions 1.0-2.6, the statements printed about whether the first step away from the saddle point towards the positive s -direction (products) is along the positive or negative direction of the imaginary frequency eigenvector were incorrect; in particular, FORMAT statements 1110 and 1111 were interchanged. This is now corrected.

6. Also in subprogram OPTION:
A second correction prevents the user from selecting an ICVT, μ VT, and/or unified statistical (US) rate constant calculation while also choosing no MEP calculation ($LGS(3) = 0$) or choosing only an MEP calculation ($LGS(3) = 3$). This change corrects an error which existed since version 1.0.
7. In subprogram PATH:
Extra statements which reversed the sign of SOB and SOE (the initial and final s values in bohr between which extra vibrational mode and Q_{vib} information is printed) when ISEN is positive (initial step from the saddle point is in the direction of products) have been eliminated. This corrects a bug which existed since version 1.5.
8. Also in subprogram PATH:
An extra statement was added in order to save the imaginary frequency eigenvector when subprogram SSAVE is called for the first time. This corrects a bug which existed since version 1.5.
9. In subprogram READ5:
More statements were introduced to prevent the user from selecting incompatible input for LGS(12), ISEN, and IWHAT for a particular run. This error existed since version 1.0.
10. Also in subprogram READ5:
The output values of SLMG and SLPG for NGSPEC = -1 were wrong in version 1.0-2.6 but are correct now.
11. In subprogram REDGEO:
Minor modifications were introduced to allow the user to choose anharmonicity option $LGS(5) = 21$ at the saddle point in MORATE runs (in which POLYRATE is interfaced to MOPAC). This corrects a bug which existed since version 1.6.

Version 2.7 (June 1991)

1. The user now has the option of calculating conventional transition state theory (TST) rate constants without performing an MEP calculation and without being required to carry out canonical variational transition state (CVT) rate constants. This enhancement can be selected by setting $LGS(7) \leq -1$, and it required changes in program MAIN and subprograms OPTION, READ5, and RATE, as well as two new subroutines, TSOUT and TSRATE.
2. In subprogram OPTION:
Seven STOP statements were replaced with LEXIT = .TRUE. statements, to allow the code to flag all incorrectly selected options before stopping. The STOP statements were introduced in version 2.0.
3. The user will now receive a warning whenever inappropriate values are selected for NTEMP and NPAIR. This required modifications in subprogram READ5.

Version 3.0 (July 1991)

1. The large-curvature tunneling method for semiclassical ground-state transmission coefficients, version 3 (LCG3) is now fully implemented for analytic potential

energy surfaces in the harmonic approximation. In particular the contributions of all accessible states populated by the tunneling process in the LCG3 exit arrangement (see "POLYRATE 4.0: ...", *Comp. Phys. Comm.* **71** (1992) 235) can be evaluated, i.e., tunneling into excited states may now be calculated. The new options are specified by $LGS(9) = 3$ or $LGS(9) = -n$ with $n \geq 2$; $LGS(9) = 3$ causes the program to include all accessible final states, whereas the $LGS(9) = -2, -3, \dots$ allows the user to specify the highest excited state to be included in terms of a maximum quantum number for a quasideiabatic mode. Five subprograms for LCG3 calculations, PROB, SINTGL, XINTGL, XTAU, and VMODE, have been renamed LCPROB, LCSINT, LCXINT, LCXTAU, and LCVMOD, respectively. Three new subprograms, LCG3TH, LCSTX, and LCTP, have been added, and more than twelve subprograms have been modified in conjunction with this enhancement of the LCG3 calculation.

2. A new option, $LGS2(3) = 1$, is now added that allows a user to omit the contributions of low-frequency modes to μ_{eff} when SCSAG tunneling calculations are performed. For this option, the user can provide a frequency ω_0 in units of cm^{-1} , and all modes with frequencies less than ω_0 will be neglected during the calculation of μ_{eff} .
3. Three new test runs have been added. Two new test runs, ch5tr8 and ch5tr9, were introduced to show LCG3 calculations with tunneling into all accessible final states [$LGS(9) = 3$] and to show the omission of curvature from low-frequency modes in SCSAG calculations [$LGS2(3) = 1$]. The third test run, involving Cl + HBr, shows an LCG3 calculation with the option, $LGS(9) = -3$, of a user-specified maximum quantum number for the quasideiabatic mode.
4. For version 3.0, unit 16 is reserved for use by MOPAC so that output previously written to unit 16 now has been assigned to unit 14. This change involves only subprograms FINOUT and FCLOSE.
5. The maximum number of points in the save grid along the MEP has been increased to 1500 in this distributed version.

Version 3.0.1 (August 1991) (Bug fixes)

1. In the subprogram READ5:
A non-standard statement introduced in version 2.7 "IF (LGS(7) .LT. 0) THEN NGSPEC = 0" is changed to "IF (LGS(7) .LT. 0) NGSPEC = 0".
2. In the subprogram REDGEO:
When $LGS(13) = 2$ is selected for the input format, the index for the coordinates is not stored correctly. It is fixed by changing "INDX(IOP,I) = I" to "INDX(I,IOP) = I". This error exists in versions 2.0-2.7.

Version 3.0.7 (September 1991) (Bug fixes)

1. Not all the FORTRAN statements involving unit 16 were changed to unit 14 in version 3.0. This inconsistency has been fixed and it involved changes in the subroutine FIOPEN.
2. Subprogram LCG3TH is modified. When only one quadratic point on a tunneling path is in the nonadiabatic region, the effective adiabatic potential for that point is

now taken as the average value of the adiabatic potential at the ending grid points of the adiabatic region on the reactant and the product sides.

3. In the subprogram LCPROJ, a ratio of "0/0" could occur when the large-curvature tunneling path is almost parallel to the reaction path, and causes an infinite looping in the subprogram LCSTX. This bug is now fixed.
4. Subprogram PSATP2 was modified so that at energies at or above the maximum of the adiabatic potential on the save grid but less than the maximum of the spline fit to the vibrational adiabatic potential curve, the turning points (both left and right) are set to the value of s at the maximum. This correction only affects LCG3 calculations involving tunneling into excited states.
5. The adiabatic potential and the turning points along each generalized normal mode, which are needed for calculating the contribution of tunneling into excited states in the large-curvature tunneling approximation, were stored only at save points greater than STP0. They are now saved at all the save points along the minimum energy path. This correction involves subprogram LCSTX.
6. The criterion for determining that a point is within the globally multi-valued region is modified in the subprogram LCG3TH.
7. A bug is fixed in subprogram LCG3, where a Kronecker delta needed for calculating the primitive LCG3 tunneling probability was missing.
8. Redundant statements are removed from MAIN.
9. For reactions without a saddle point, a generalized normal mode analysis is now done on the projected force constant matrix for the first point on the reaction path. The first point was previously treated incorrectly. Note that the output files for test runs hootr1, hootr2, and hootr3 are also modified due to this change.
10. For calculations using the cubic initial step option, the stored classical energy at the saddle point was incorrect. This involves corrections in the subprogram PATH. This modification corrects the calculated conventional transition state rate constants of test runs oh3tr1 and oh3tr3.
11. The variable IMAX was used both as an index for the maximum of the adiabatic potential, VS(IMAX), and as an index for the spline segment where VMAX occurs. This error had caused turning points to be set incorrectly to zero in some cases. This involved changes in the subprograms KAPVA, PSAG, PSATP, PSATP2 and VSPLIN.
12. An unnecessary tolerance check in the subprogram FDIAG which could, under some circumstances, lead to an infinite loop has been removed. Also, one of the looping structures in this subprogram has been modified to avoid unnecessary checking.
13. The tolerance criterion in the subprograms VPART, WKBPOT, and HQSC has been improved. This change does not affect the rate constants within the level of accuracy reported, but it does make the iterative calculation more robust.
14. The number of excited states included in an LCG3 calculation:
The maximum excited state into which the p mode can be excited is limited by the parameter MAXPS defined in the INCLUDE file param.inc. The program is modified

such that if the number of excited states to be included is not specified (i.e., $LGS(9) = 3$) then the number of states to be included is at most MAXPS even though there may be more accessible states. If the number of states specified by the user is larger than MAXPS and the number of accessible states are also larger than MAXPS, a warning message will be printed and the calculation is carried out with MAXPS states.

15. A bug in the LCG3 calculations including tunneling into excited states was corrected. In versions 3.0 and 3.0.1, the uniformization is carried out only for $n_{\max} = n_p^{\text{Max}}$ rather than for all values from 0 to n_p^{Max} or to the largest value allowed by the dimensions (MAXPS).

Version 3.9 (September 1991)

1. A new small-curvature tunneling approximation, called the centrifugal-dominant small-curvature adiabatic ground-state tunneling approximation, the CD-SCSAG method, is implemented. Two new subprograms are added, and 15 others are modified. The section in the output files with the final results has been restructured accordingly. This involved changes in the subprogram FINOUT. We now distribute short output files of the sample runs using the CD-SCSAG method instead of the original SCSAG method, since the CD-SCSAG method is the one we recommend for the small-curvature cases. The default is now CD-SCSAG.

Version 3.9.1 (September 1991) (Bug fixes)

1. In subprogram VSPLIN, a missing RETURN is now added.
2. Subprogram ANHARM is modified to avoid losing accuracy due to multiple transformations between mass-scaled Cartesian and Cartesian coordinates.
3. The option of optimizing certain Cartesian components to search for a stationary point was not available when $LGS2(2) = 1$. This problem caused failures in searching for stationary points of certain systems. It is now fixed by modifying subprogram POLYAT.

Version 3.9.2 (November 1991) (Bug fix)

1. Subprogram FDIAG is modified to always give eigenvectors of $3N$ components, whether the species being analyzed is a reactant, a product, or a generalized transition state, where N is the number of atoms in the reaction. Due to this change, subprograms ANHDER, NOROUT, SADDLE, and SGND3V are also modified.

Version 3.9.5 (March 1992) (Bug fixes)

1. Subroutine READ5 is modified to always read in DEMIN when Morse anharmonicity or a μ VT calculation is requested.
2. The dimensions of certain arrays in subprograms AITKEN, AITKF2, AITKNF, KAPVA, LOCS, SL1B1, SL1B2, and VSPLIN are modified to match the sizes in the corresponding subprograms which pass the arrays through call statements to these subprograms. Among the machines on which the POLYRATE program has been

tested, a discrepancy due to this bug occurs only on the Sun SparcStation 4/360 (SunOs 4.0.3).

3. A gradient point was previously determined to be a save point if it is less than a gradient step away from an end point specified by the user. This criterion is now changed to be less than half a gradient step to avoid possible discrepancies caused by machine precision. The correction is made in PATH.
4. In subprogram OVERLP, the array INDEX, which stores the indices of the generalized normal mode eigenvectors, was initialized to zero. To avoid the error of trying to retrieve the zeroth array element, this array is now initialized as INDEX(I) = I.
5. In subprogram READ5:
For LGS(35) \neq 0 (i.e., a MORATE calculation), LGS(25) and LGS(33) are set equal to zero, as these options are not currently supported for a calculation that is not based on an analytic potential energy surface.
6. In subprogram RESTOR:
A typographical error introduced in version 1.6 has been corrected. This entailed changing the statement "READ(2, *) (XER(2), I = 1, NTOT2)" to "READ(2,*) (XER2(I), I = 1, NTOT2)". This bug fix will affect any calculations which used the option to merge reaction paths from units 1 and 2 and also involved anharmonicity for the generalized normal modes.
7. In subprogram LCSET:
A typographical error in the call statements for the subroutine INTRPL has been corrected. This involved modifying the statement
CALL INTRPL(XETS,SSUBI,ANHARM,...)
to
CALL INTRPL(XETS,SSUBI,ANHRM,...)
There were two occurrences of this problem in this subprogram. This bug fix will not affect any currently supported options because it relates to the anharmonicity parameters and it occurs in an LCG3 subprogram, but anharmonicity is not yet supported with LCG3 tunneling.
8. In subprogram LCG3TH:
The call statement for the subprogram INTRPL has been corrected:
CALL INTRPL(XETS,SSUBI,ANHARM,...)
has been changed to
CALL INTRPL(XETS,SSUBI,ANHRM,...)
The CALL statements for the subroutine LCVOR incorrectly used the variable XE in the argument list instead of the array ANHRM; this has also been corrected. These changes have no effect on any currently supported options because they affect the anharmonicity parameters in a large-curvature tunneling subprogram, and anharmonicity is not yet supported with LCG3 tunneling.
9. FORMAT statements that were incompatible with NATOMS larger than 6 and which would cause output conversion errors for the cwmctr1 and cwmctr2 test runs were fixed. This involved changes in the following subroutines: CENTER, NEWT, POLYAT, SADDLE, and REACT. Also, FORMAT statements 2300 and 2330 in RESTOR have been corrected; these errors existed since version 3.0.7.
10. Spelling and grammatical errors were corrected in FORMAT statements in the POLYRATE subroutines HEADER and OPTION, and in the potential routine cwmc.f.

Version 4.0 (March 1992)

1. All the common blocks are byte aligned. All the PARAMETER statements and common blocks that are used throughout the code are now placed in three INCLUDE files, param.inc for the adjustable parameters, percon.inc for the fixed parameters, and common.inc for the common blocks. This modification makes redimensioning arrays much more convenient. Also, to maintain the code's portability, non-ANSI standard FORTRAN 77 statements were corrected in the following subroutines: NEWT, NEXTPT, RATE, LCG3TH, and LCVMOD.
2. The dimensions of the arrays for storing the information along the minimum energy path are now controlled by two parameters, NSDM and NSDML. This modification allows users to reduce the required memory when LCG3 calculations are not requested.
3. A new method for approximating the partition functions of selected low-frequency vibrational modes corresponding to hindered internal rotation is implemented. This is called the hindered-internal-rotator approximation.
4. A new option, LCG2(4), has been implemented. This flag allows the user to specify a nonzero value of EZERO, which is defined such that $V - EZERO = 0$ at the reactants. If a restart file is written to unit fu1 or read from unit fu1 then a value for EZERO is also written/read. Because of this enhancement old restart data files will not work with this version of the code.
5. All STOP statements in all subprograms other than MAIN have been changed from STOP to STOP 'subprogram name #' where # is 1,2, etc. sequentially for each subprogram.
6. All STOP statements in the subprograms READ5, REDGEO, and OPTION which flagged incorrect input were changed to LEXIT = .TRUE. so that the code would check the entire input deck before quitting.
7. All input and output I/O units for the POLYRATE code are now adjustable. They can easily be changed by modifying the variables fu# (e.g., fu6 for unit 6) defined in the INCLUDE file param.inc.
8. All the input and output files used by POLYRATE, except for the files linked to FORTRAN units fu5 and fu6, are explicitly opened and closed in the subprograms FIOOPEN and FCLOSE, respectively. The files for units fu5 and fu6 are opened in the subprogram MAIN. The default file name is poly.fu#, which is linked to FORTRAN I/O unit fu# defined in the INCLUDE file param.inc. For example, if fu15 is set to unit 15 in param.inc, the poly.fu15 file will be opened and linked to unit 15.
9. The information for restart calculations is now written with formatted statements rather than with free formatting for certain long records. This modification enhances the portability of the POLYRATE code.
10. The treatment of diatomics has been made more flexible and now allows them to be treated as general linear molecules; because of this, the input data files of all the test runs that involve diatomic molecules are modified such that diatomic molecules are treated in the same manner as linear polyatomic molecules. Before, diatomic molecules were treated as Morse oscillators. This Morse oscillator option is still

available and can give accurate results with less effort if the diatomic molecules are indeed Morse oscillators. Due to this change, the fu30 file, oh3rph.dat, for the test run oh3tr6 is also modified.

11. The test runs are reorganized according to the potential functions. Four sample runs are added, and one is modified: cwmctr1 is an example of calculating the conventional transition state rate constants only; cwmctr2 uses the hindered-internal-rotator approximation; clhbrtr1 is modified to use the LGS2(4) = 1 option, i.e., using POLYRATE to calculate EZER0; clhbrtr2 is added to demonstrate the option of reading EZER0 in the fu5 input; and, finally, ch5j1tr6 is added to show that LCG3 calculations that include tunneling into excited states can be carried out using the restart information from a run that includes only LCG3 tunneling into the ground state.
12. The manual no longer includes the fu5 input and the fu14 and fu15 short output files for the sample runs. These input and output files are instead included in the POLYRATE 4.0 distribution package. The advantage of this is that users can make input data files by modifying the sample input files, and the comparison to the sample run outputs can easily be done by using the "difference" or "diff" command.
13. Since the number of accessible states as a function of reaction coordinate, $N^{\mu VT}(s,E)$, is not a continuous function of s , the 3-point and 5-point interpolations to find the maximum, as used in previous versions, may be unstable. Thus we now also print out the maximum value on the grid. This may be used for a "one-point" interpolation, which may be more reliable.
14. A new subprogram, INITZE, has been added to the code. This subprogram initializes variables and arrays used by the program. This subprogram currently does not initialize all the arrays and the variables used by the program.
15. The variable IOT has been changed. In version 4.0 it is a flag used by the LCG3 subprograms to indicate whether or not state-labeled primitive tunneling probabilities should be printed to unit fu22. Note: in versions 3.0.7-3.9.5, the printing of this extra information was controlled by LGS2(5).
16. A new option, LGS2(5), which allows the user to print extra information from the potential energy surface routine to the POLYRATE long output, unit fu6, has been added. The number of pieces of information which can be printed is controlled by the values of the input variable LGS2(5) and the variable NPOTPT which is set in the INCLUDE file param.inc.

Version 4.0.1 (March 1992) (Bug fixes)

1. In subprogram TSRATE, a line was added to allow for the correct calculation of anharmonicity when $LGS(5) \geq 21$ and $LGS(7) < 0$. Versions 2.7-4.0 gave correct rate constants and equilibrium constants only for the first temperature for this combination of LGS options.
2. FORMAT statements in various subprograms affecting the POLYRATE output were corrected. This does not affect any results of the calculation.

3. Minor modifications were made to allow the user to control the printing of the bond orders and charges calculated in MORATE. This modification affected the following subprograms: MAIN, PATH, POLYAT, RESTOR, and SADDLE.
4. In subprogram MAIN:
The statement

```
IF (LGS(1).EQ.0 . . . )
```

has been changed to

```
IF (LGS(30).LE.0.AND.LGS(35) = 0)
```

This modification removes the redundant check for $LGS(1) = 0$. This clean-up of the code is not a bug fix and does not affect any of the test runs.

Version 4.1 (April 1992)

1. In subprogram OVERLP:
Prior to version 4.1, when an LCT calculation was requested, the program re-ordered the generalized normal modes and related quantities, e.g., eigenvectors and eigenvalues, at the current save grid point so that the eigenvector of each mode at the current save grid point had the highest overlap with the eigenvector of the same mode at the previous point. If two modes at the current point each had their highest overlap with the same mode from the previous point, the program would print a message on the unit fu6 file and stop. This was modified in version 4.1. In version 4.1, once a mode at the previous point has been matched to a mode at the current point, it is no longer tested against further modes at the current point. Thus the conflict never occurs.

Version 4.1.1 (April 1992) (Bug fixes)

1. In subprogram REACT:
A bug in the computation of the zero of energy for a bimolecular reaction has been corrected. This bug was introduced in version 4.0.
2. In subprogram VTMUSN:
A call to subprogram TREPT, which computes a parabolic fit of free energy curves, contained one extra argument. This bug fix affects μVT rate constant calculations using a 3-point primary fit, which is selected by setting IFTMV1 to 3 in the input file on unit fu21. This bug was introduced in version 4.0. It does not affect the results of the test runs.
3. Certain non-ANSI statements in files dattim.sun, dattim.ibm, and dateclock.c have been corrected to ANSI-standard. Non-ANSI statements in subprogram MAIN and potential surface function files hni.f, h2ni.f, and cwmc.f are also corrected.
4. A bug has been corrected in OVERLP. In previous versions, the switching of the eigenvalues and eigenvectors was sometimes incorrect, especially for very low frequency vibrations. The new version corrects the error, and in addition it should be more robust because the matching is now carried out in order of decreasing frequency of the vibrational frequencies.

Version 4.2 (May 1992)

1. Subprogram FDIAG has been restructured. In this version of POLYRATE the generalized normal modes are now ordered according to the absolute values of their eigenvalues. In addition, for a save point along the minimum energy path the

matching of the generalized normal modes with those at the previous save point is carried out between the vibrational normal modes only. That is, the modes corresponding to the translational, rotational, and reaction coordinate degrees of freedom, which are assumed to be the modes with the smallest absolute eigenvalues, are not used in the matching process. This also involved changes in OVERLP.

2. In subprogram OVERLP:
The IF statement 'IF (NN.EQ.0) GO TO 200' has been modified to be 'IF (NN.LE.1) GO TO 200'.
3. FORMAT statements in several subroutines have been modified to make the output more clear. In particular, the output for the reaction energetics has been improved. The output has also been modified so that most of the information written to unit fu6 is lower case.
4. In subprogram INITZE:
A check for the value of LGS(4) when LGS(35) = 0 has been coded; this was necessary because in POLYRATE LGS(4) is allowed to have values of 0, 1, and 2, but in MORATE LGS(4) can also have a value of 3.
5. Test run oh3tr1 is modified to test the option LGS2(4) = 1. This option allows users to calculate the zero of energy by POLYRATE.

Version 4.2.1 (June 1992)

1. In the INCLUDE file common.inc:
The size of array LRP, which is only used in the subprogram RATE for calculating vibrational partition functions of reactants and products, is increased from N3TM = 3*NATOM to N6TM = 6*NATOM, since LRP(N3TM) may not be sufficiently large for some reactions.
2. In subprogram INITZE:
The array LRP, which is used to indicate whether or not a particular vibrational mode is thermal or not, is initialized. All modes are initialized to be thermal, as this is the case unless LGS(23) \neq 0.

Version 4.3 (June 1992)

1. In subprogram OPTION:
A check for the value of LGS(3) when |LGS(9)| > 1 has been coded. This check was added because the LCT calculation is not supported when LGS(3) < 0.
2. In subprogram PATH:
The calculation of the MEP and properties at points on the MEP will now be stopped if the number of save points reaches parameter NSDM, which defines the size of arrays for storing MEP information.

Version 4.3.1 (July 1992) (Bug fixes)

1. In subprogram LCPROB:
In versions 3.0-4.3, the LCT primitive probabilities were uniformized with respect to the primitive probability at the highest energy grid point whereas the correct

procedure is to use the value at the maximum energy of the adiabatic ground state potential. This is now corrected.

2. In subprogram LCG3:
In versions 3.0-4.3, the total number of accessible states in the LCT calculation was set equal to the largest value of n_p that satisfies eq. (11) in *Comp. Phys. Comm.* **71** (1992) 235, at the highest energy grid point. The calculation of the maximum value of n_p is now handled correctly, as discussed in *Comp. Phys. Comm.* **75** (1993) 143.
3. In subprograms LCG3TH, LCSTX, LCVCOR, and LCXTAU:
In versions 3.0-4.3, when quantities derived from the vibrational normal mode frequencies were calculated, the contribution from imaginary normal modes were included incorrectly. This is now corrected.
4. In subprogram LCG3TH:
For a given tunneling path, the program determines the boundary of the adiabatic region, which consists of two points, ξ_a (or x_{ia}) and ξ_b (or x_{ib}). When ξ_a (or x_{ia}) is greater than ξ_b (or x_{ib}), the integrand at ξ_b (or x_{ib}) was included twice in the calculation of the theta integral. This is now corrected.
5. In subprogram LCG3:
In versions 3.0-4.3, the numerical integration of $T_1^{LCG3}(E, n_p)$ [eq. (35) in *Comp. Phys. Comm.* **71** (1992) 235] was incorrect when $n_p > 0$ and the backward reaction is exoergic. The upper bound of the range of integration should be the reaction coordinate turning point calculated from the adiabatic excited-state potential. However this information was not stored and the turning point calculated from the adiabatic ground-state potential was used. This is now corrected.
6. In subprogram READ5:
An IF statement for checking the conflicts between the option for tunneling and the option for anharmonicity was not written properly. Due to this problem, users were not able to start a calculation if a mode was to be treated as a harmonic oscillator when $LGS(5) \geq 21$ and an LCG3 calculation was requested in the same run. This bug existed in versions 4.0-4.3.
7. In subprogram RESTOR:
Several WRITE statements were mistakenly coded as READ statements. This caused I/O conflicts when information from two restart files was to be merged into a new restart file. This bug existed in versions 2.0-4.3.

Version 4.4 (August 1992)

1. One of the criteria for determining whether a point on an LCG3 tunneling path is within the adiabatic region is to check if that point is within the turning points of each of the generalized normal modes. In subprogram LCZCRT, the generalized normal mode vibrational coordinate for each mode was compared to both turning points of that mode. The checking procedure is now made more efficient by using eqs. (30a) and (30b) of *Comp. Phys. Comm.* **71** (1992) 235, in which the absolute value of the turning point of each generalized normal mode is used in the comparison. Also the algorithm for calculating the vibrational turning points is

improved in subprograms LCG3TH and LCSTX. Prior to this version, vibrational turning points were calculated from the frequencies at each save grid point and then the vibrational turning points at various s values were obtained by interpolation from this grid. Starting with version 4.4, only the vibrational frequencies are interpolated, and vibrational turning points are always calculated from the interpolated frequencies. The two algorithms yield very similar results when the generalized normal mode frequencies are much greater than zero, but the new algorithm is more stable when any frequencies are small (near zero) and noisy. In addition, subprogram LCG3TH has been restructured. This modification reduces unnecessary calculation.

2. The LCT method is now generalized to cases involving a well on the product side of the MEP in the exoergic direction. The program first checks if the adiabatic ground-state potential at the last save grid point on the product side of the MEP (s_+) is less than the value of the product ground-state energy. If so, a new point, called the well boundary or s_{WB} , is identified as the farthest out s value for which $V_a^G(s)$ is greater than or equal to the product ground-state energy. Then the maximum number of accessible states is determined as discussed in *Comp. Phys. Comm.* **75** (1993) 143, using s_{WB} instead of s_+ . The contribution from tunneling paths with termini beyond the well boundary is not included.
3. In the calculation of the LCG3 tunneling amplitude, the theta integral, the sine of the angle between the tunneling path and the MEP, the vibrational period, and the excited state potential are obtained by interpolation at NGS0 points between each pair of adjacent energy quadrature points. Prior to version 4.4, 4-point interpolation was used to calculate these quantities. A new option that allows users to specify the order of interpolation is added in version 4.4 because sometimes a more stable, lower-order interpolation is needed for good accuracy. The default is 4-point interpolation and the desired order is set equal to LGS2(9). The input data files of test runs clhbrtr1 and clhbrtr2 are modified to use 2-point interpolation. Details of these two test runs can be found in the section on sample test runs. This involves modifications in subprogram LCSINT.
4. An option for extra output containing details of LCT calculations has been added. In particular, the extra output includes the theta integrand and the corresponding s value on the MEP at each point along the tunneling path, the boundary of each adiabatic region on the tunneling path, the length of the tunneling path, and the classical potential and the correction terms to the potential for points in the non-adiabatic region. This involves modifications in subprograms FIOPEN, FCLOSE, INITZE, LCG3, LCG3TH, LCZCRT, READ5. The C-shell scripts poly.inccr and poly.exec are also modified to accommodate the change in the file usage. The command files for the VMS operating system remain the same since none of the test runs uses this option.
5. In the input data files ch5j1tr1.dat, ch5j1tr2.dat, ch5j1tr5.dat, ch5j1tr6.dat, clhbrtr1.dat, clhbrtr2.dat, h2nitr1.dat, and h3tr2.dat: The numbers of quadrature points used in the calculation of tunneling probabilities, as specified by NQE, NSEG, NQTH, NSEG2, NG, and NGS0, are decreased to reduce the computation time while still converging the transmission coefficients within 1%. This is possible because the current algorithm is more stable due to modifications 1 and 3.

6. A new subprogram NSTATE has been added. This subprogram is used to calculate the total number of accessible states and to determine if an excited state is open for a given tunneling energy. This modification reduces redundant calculations.
7. In subprogram LCNWTN:
The call to subprogram INTRPL to interpolate the gradient vector is modified. The parameter N3TM is used in the argument of the call instead of the variable NDIMX. This is to clean up the program (not a bug fix).
8. In INCLUDE file percon.inc and subprograms BOLTZ, KAPVA, and LCG3:
The dimension of arrays related to the calculation of tunneling probabilities was declared as 243 in previous versions. It is now set to the parameter NSV, and NSV is set equal to 245 in the INCLUDE file percon.inc. The arrays that were modified are PE, PEC, PED, ESV, PLCG3, TRNPT1, and TRNPT2.
9. In versions 4.0-4.3.1, variables N3 and LSAVE were passed to subprograms LCSET and LCSTX as arguments in the CALL statements while they were also passed through common blocks. The program is now cleaned up such that these variables are now only passed through common blocks. Subprogram LCG3 is also modified due to this change.
10. In subprograms ANHARM, FITMAX, LCG3, LCVIB, MUBAR, PSAG, PSATP2, QQPOT:
Prior to version 4.4, certain array elements in these subprograms were passed to other subprograms as arguments in the CALL statements. However, they were not declared as one-dimensional arrays in the subprograms that are called. In version 4.4, dummy variables are set equal to these array elements, and then the dummy variables are passed to other subprograms as arguments in the CALL statements.
11. In subprogram READ8:
In versions 4.0-4.3.1, the variable NATOM and the array elements LGS(13) and LGS(23) were passed to READ8 as arguments in the CALL statements while they were also passed through common blocks. In addition, the array elements were not declared as one-dimensional arrays in READ8. In version 4.4, NATOM, LGS(13), and LGS(23) are no longer passed in the CALL statements. Due to this change, subprogram READ5 and MAIN are also modified.
12. In subprogram SGND3V:
The variable NEND that was passed to SGND3V as an argument in the CALL statement was not used in SGND3V. It is now removed. Subprogram ANHARM is also modified due to this change.
13. In subprogram TURNPT:
The variable NMODE that was passed to TURNPT as an argument in the CALL statement was not used in TURNPT. It is now removed. Subprogram MUBAR is also modified due to this change.
14. Subprogram FIOPEN has been restructured. With this modification, the names of any files that cause errors as they are opened or attempted to be opened will be given in the poly.fu6 file.
15. In subprogram READ5:

A check for the values read from unit fu5 for the temperatures has been coded. This check is only carried out if NTEMP is greater than 0, and it stops the calculations if any of the temperature values are not greater than zero.

Version 4.5 (September 1992)

1. In subprograms PATH and READ5:
In previous versions, when the cubic starting algorithm was requested, the third derivative of the potential with respect to the coordinates was evaluated using a step size equal to DLX, which is the step size for numerical second derivatives. In version 4.5, this step size is set to D3LX specified in the poly.fu5 input file. The INCLUDE file common.inc is also modified due to this change.
2. In INCLUDE files poly9995.inc, poly9911.inc, poly9111.inc, poly6775.inc, poly6711.inc, and common.inc:
The array SSPEC, which contains the special save grid points (see records 18 and 19 for fu5 input) is now dimensioned to NMSPEC, which is specified in the INCLUDE file param.inc. The subprogram READ5 is also modified so that a warning message will be written to the fu6 output if the user-input value of NSPEC is greater than NMSPEC.
3. In subprogram MAIN:
The statement "STOP 'MORATE MAIN 1'" is modified to "STOP 'MAIN 1'". Also several commented statements for opening files morate.fu5 and morate.fu6 are removed. This is to facilitate the procedure of interfacing POLYRATE with electronic structure packages.
4. In subprograms MAIN, COLSHF, MEPSRT, PATH, POLYAT, REACT, RESTOR, RESTRT, SADDLE, and SSAVE:
The statement "include 'morate.inc'" is modified to "include 'esp.inc'". This is to facilitate the procedure of interfacing POLYRATE with a variety of electronic structure packages.
5. In subprograms FIOPEN, FCLOSE, MAIN, POLYAT, REACT, and SADDLE:
The statements to open the files esp.fu31—fu35 are added in FIOPEN. These files correspond to FORTRAN units fu31—fu35 which are specified in the INCLUDE file param.inc. The sample INCLUDE files for param.inc, i.e., poly9995.inc, poly9911.inc, poly9111.inc, poly6775.inc, and poly6711.inc, and the C shell script poly.inccr are also modified due to this change. This is to facilitate the procedure of interfacing POLYRATE with a variety of electronic structure packages.
6. Subprogram HEADER is renamed as HEADR. This is to facilitate the procedure of interfacing POLYRATE with a variety of electronic structure packages.

Version 4.5.1 (September 1992) (Bug fix)

1. In subprogram REACT:
A bug in the computation of the zero of energy for a unimolecular reaction when LGS(35) = 0 has been corrected. This bug was introduced in version 4.1.1.

Version 4.5.2 (March 1993) (Bug fixes)

1. In subprogram READ5:
A bug in the following print statement
`WRITE (FU6,7200)`
has been corrected. The statement has been changed to
`WRITE (FU6,7200) NTEMP`
This bug did not affect any of the calculations; only the output was affected.
2. In subprogram READ5:
For $LGS(35) \neq 0$ (for MORATE calculations), a bug in the read statement for DLX and DERSTP has been corrected. This statement read DLX and DERSTP but it should have read DLX, DERSTP, and D3LX. The statement for setting D3LX equal to DLX if $D3LX = 0$ and $LGS(35) \neq 0$ has also been added to this subprogram.
3. In subprogram RPHRD2:
For $LGS(30) = 1$ and $LOPT(4) = 0$, the `FREQ` array which is used to store the input frequencies was not initialized before the subprogram read in the frequencies of a new species. This bug did not affect any of the calculations, but it caused some wrong output. This has been fixed by adding appropriate initialization statements.
4. In subprogram HINDRT:
The variable `NATOM` was reinitialized in the subprogram HINDRT as a local variable. This could cause problems when the vibrational modes of the reactants or products are treated as hindered internal rotators. This bug, which was introduced in version 4.0, has been corrected. None of the test runs in the distribution package are affected by this bug.

Version 5.0 (March 1993)

1. An algorithm to allow frequencies, potential energies, and moment-of-inertia-tensor determinants to be corrected by an interpolation scheme has been added. This method is called VTST-IC (i.e., VTST with interpolated corrections). In the present version, we have included zero-order interpolation, i.e., VTST-IC-0. At the zero-order level, corrections are interpolated by using the accurate values for the reactant or a pre-saddle minimum, the saddle point, and the product or a post-saddle minimum. Various types of functions are used for interpolation depending on the shape of MEP and on the relative magnitudes of the corrections at those points mentioned above. Details of this zero-order interpolation scheme are given in the subsection on IVTST in the sections on theoretical background for dynamics calculations and selected details of the calculations. The various options of this interpolation scheme can be selected by setting LGSIC options in the FORTRAN unit fu50 input. In addition, users should supply accurate values of quantities needed for interpolation on FORTRAN unit fu50. For this option, subprograms COG, COHBT, ECKART, HBT, READ50, UPDMI, VICLCG, ZOC3P, ZOCAST, ZOCFRE, ZOCMI, ZOCPAR, ZOCPRN, ZOCVCL, and ZOCUPD have been added. Subprograms LCG3TH, MUBAR, MUCDSC, PATH, READ5, RESTOR, RESTRT, ZEROPT, and MAIN, and the INCLUDE files common.inc and param.inc have been modified.
2. In version 5.0, if $LGS(30) < 1$, normal mode frequencies and generalized normal mode frequencies are ordered in descending order for all output and input. Related quantities, e.g., normal mode eigenvectors, zero point energies, individual vibrational partition functions, etc., are also ordered accordingly in all output.

Subprograms FINOUT, REACT, READ5, and REDGEO have been modified for this change.

3. The canonical optimized multidimensional tunneling (COMT) approximation and the microcanonical optimized multidimensional tunneling (μ OMT) approximation are now implemented. These methods are explained in detail in the subsection on optimized multidimensional tunneling calculations. These methods will be used in evaluating transmission coefficients whenever SCT and LCT calculations are both requested. (In version 5.0, SCT calculations are carried out whenever LCT calculations are carried out, so the critical distinction is whether LCT calculations are requested.) The new subprogram PBTSRT has been added. Subprograms KAPVA, FINOUT, and TABLE and the INCLUDE file common.inc have been modified.
4. A new option for calculating the vibrational partition functions using simple perturbation theory [Truhlar and Isaacson, *J. Chem. Phys.* **94** (1991) 357] for the case of no degenerate vibrational modes was implemented. (Corrections for Fermi or other resonance are not included yet.) In this method, vibrational partition functions are evaluated using the ground-state energy and the fundamentals, which are calculated using perturbation theory up to first order for the quartic terms and the Coriolis interaction, and 2nd order for the cubic terms. Subprograms ANCOEF, CHKFRE, CHKUP, CORTRM, GRAD, GRADDER, and PTQVIB have been added. The INCLUDE file common.inc and subprograms CENTER, COLSHF, INITZE, MEPSRT, NORMOD, PROJCT, RATE, RESTOR, RESTRT, SSAVE and TSRATE have been modified.
5. In subprogram FINOUT:
In version 5.0, the output of calculated transmission coefficients in the file linked to unit fu6 and rate constants in the files linked to units fu6 and fu15 has been cleaned up, and only the quantities that are requested to be calculated will be printed.
6. Various changes have been made to make the subprograms more modular. They are summarized as follows:
 - 6A. In subprogram TRANS:
The position and the gradient vectors are now passed as arguments in call statements rather than being passed through the common block POTCM. Subprograms ANHARM, ANHDER, INTFNC, INTPM, LCG3TH, NORMOD, PATH, PHSINT, QQPOT, REACT, TP, and WKBVIB have been modified.
 - 6B. In subprograms RESTRT and RESTOR:
Subprogram RESTRT now writes MEP information to unit IOUT which is passed from the calling subprograms as an argument in the call statements. Subprogram RESTOR has been restructured, and the writing of the information corresponding to the merged MEP to FORTRAN unit fu3 has been removed since this function is now carried out by subprogram RESTRT.
 - 6C. The calculations of quantities needed for the cubic-starting algorithm have been removed from subprogram PATH, and they are now carried out by a new subprogram CUBST.
 - 6D. The calculations related to the electronic structure input file option have been removed from subprogram PATH, and these are now carried out by a new subprogram ELRPH.

- 6E. The writing of reaction energetics has been removed from subprogram NOROUT, and these are now written by a new subprogram ENROUT.
- 6F. The writing of saddle point energetics has been removed from subprogram MAIN, and these are now written by a new subprogram SENOUT.
- 6G. The part of the code that prints MEP information, including V_{MEP} , V_a^G , μ_{CD-SC} , and generalized transition state frequencies has been taken out of the subprogram PATH and placed in a new subprogram MEPOUT. This subprogram is always called by PATH if the MEP is calculated.
- 6H. The contents of the subprogram MAIN have been grouped into 6 new subprograms, DOREST, DORPH, DOREPR, DOSAGE, and DOPNM. These subprograms are called by MAIN to control the VTST calculations.
- 6I. In subprogram RESTOR:
For a restart calculation, the variable EZERO is echoed to FORTRAN unit fu6. The write statements for writing the MEP information to FORTRAN unit fu6 have been replaced by a call to the subprogram MEPOUT. All remaining FORMAT statements in this subprogram have been modified to be case sensitive and to fit in 80 columns.
- 6J. In subprogram MAIN:
The open statements for the files linked to FORTRAN units fu5 and fu6 have been replaced by calls to the new subprogram FIOPEN.
- 6K. For $LGS(35) = 1$, i.e., an electronic structure input file calculation, the calculation, printing, and storage of the bond orders and charges have been removed from the code. This involved changes in the following subprograms: COLSHF, MAIN, MEPSRT, PATH, POLYAT, RESTOR, RESTRT, SADDLE, and SSAVE. This was done to make the link between POLYRATE and any electronic structure package more modular and more transparent.
7. Various changes have been made to clean up the program. They are summarized as follows:
- 7A. The matching of normal modes to those at previous save grid points has been changed. Prior to version 5.0, the usage of arrays was different when LCG3 calculations were requested from when they were not. In this version, the program is restructured so that the matching of normal modes is always performed in the same fashion regardless of which tunneling method is used. Subprograms FDIAG, INITZE, and OVERLP have been modified to accommodate this change.
- 7B. In subprogram ZEROPT:
The number of degrees of freedom corresponding to rotation, translation, and the reaction path (if GTST), and the number of degrees of freedom of a species are now evaluated in subprogram ZEROPT rather than being passed to ZEROPT. Subprograms RPHINT and RPHRD2 are also modified.
- 7C. The error-checking on the compatibility of ICODE(IOP) and LGS(34) has been moved to subprogram REDGEO.

- 7D. Extra blank lines in the long output (FORTRAN unit fu6) have been eliminated, and the layout and the clarity have also been improved. Most of the subprograms that echo information to the long output have been modified.
- 7E. All the unit conversion and variable initialization that were in the subprogram MAIN have been moved to the subprograms INITZE and READ5.
- 7F. The number of vibrational modes for each reactant and product species and for the generalized transition state is now only calculated in the subprogram REDGEO and is stored in the array NF. After that, these numbers are referenced through the array elements whenever needed. The calculation of these numbers in other parts of the program has been removed.
- 7G. In subprogram BCALC:
Two lines of FORTRAN which were commented out have been removed. These lines were used for debugging during the preparation of version 4.0.
- 7H. The option LGS(26) = 1, which allowed the user to force the points on the MEP to have a particular symmetry, has been removed in this version of the code. This involved changes in the following subprograms: ELRPH, OPTION, PATH, and READ5, and the removal of the dummy subprogram SYMCHK.
- 7I. As a result of item 6K, the option LGS(4) = 3 has been removed; this option controlled the printing of the bond orders and the charges which are no longer calculated in this version of POLYRATE. This involved changes in the following subprograms: INITZE, MAIN, PATH, POLYAT, SADDLE, and RESTOR.
- 7J. All C*M lines, except those for the statement "include 'esp.inc'", have been removed. This involved changes in the following subprograms: MAIN, POLYAT, REACT, READ5, and SADDLE. This will cause the program to perform a couple more 'if' checks, but it should not slow the code down appreciably.
8. In subprogram NORMOD:
The position vector of a GTS in the mass-scaled Cartesians is now saved in a temporary array before it is transformed to unscaled Cartesians to evaluate the second derivatives. The mass-scaled position vector is restored afterward by using the values in the temporary array. This procedure avoids the round-off error when the position vector is transformed back and forth between the scaled and unscaled Cartesians.
9. The SCSAG tunneling method has been removed because it is obsolete, being replaced by CD-SCSAG. As a consequence, in the unit fu5 input file, the options LGS2(3) and LGS2(6) on record 1A and the variable M0 on record 20 are no longer used. Subprograms BCALC0, ELRPH, FINOUT, KAPVA, MUCDSC, OPTION, PATH, and PSAG have been modified.
10. The call to subprogram MEPSRT by subprogram PATH has been replaced by a call to the new subprogram MEPINV for better performance. The subprogram MEPSRT is now only called by the subprogram RESTOR.
11. The common block POTCM in the INCLUDE file common.inc has been renamed PEFCOM. This has been done to avoid conflicts between this common block and the one used in the three-body analytic potential energy functions that are in the test suite.

12. The code has been modified so that all the files used in a calculation are no longer closed at the end of the calculation by a call to FCLOSE. Instead, the files in a calculation are closed after they are used; in fact, there are three points in the calculation when files are closed. First, after the call to READ5, all the input data files that have been used up to that point are closed by a call to the new subprogram FCINPT. Second, after the MEP calculation, all input and output files used during the MEP calculation are closed by a call to the new subprogram FCMEP. Finally, at the end of the calculation, all files, input and output, which have not been closed are closed by a call to the new subprogram FCRATE. The subprogram FCLOSE no longer exists. The three new subprograms, FCINPT, FCMEP, and FCRATE, call another new subprogram FICLSE; this subprogram closes a file linked to a FORTRAN unit NUNIT with the status FSTAT. This change was made to safeguard against loss of information in the event of a system or code crash.
13. Some changes have been made in the POLYRATE test runs. They are summarized as follows:
 - 13A. Five new test runs have been added to this version of POLYRATE to test the VTST-IC options. They are cmctr1, cmctr2, ho2tr4, ho2tr5, and nh3tr1.
 - 13B. The forward symmetry factors SIGMAF in the unit fu5 input data files of the ch5 test runs have been changed to the correct values. The forward rate constants are a factor of 4 smaller for the test runs ch5j1tr5 and ch5j1tr6, and a factor of 2 smaller for other ch5 test runs than they were in the previous versions.
 - 13C. Copies of the short output files linked to FORTRAN unit fu14 have been removed from the POLYRATE distribution.
 - 13D. For the clhbr, ho2 and oh3 test runs, the data for the potential energy surfaces have been placed inside the respective potential energy surface codes, so that no potential energy surface data need now be read.
14. The INCLUDE files poly9995.inc, poly9911.inc, poly6775.inc, and poly6711.inc have been renamed param4.inc, param3.inc, param2.inc, and param1.inc, respectively, and the include file poly9111.inc has been removed from the POLYRATE distribution.
15. The subroutine RST has been renamed RSPDRV to avoid the name conflict with the subroutine RST in Cray's science library.

Version 5.0.1 (May 1993) (Bug fixes)

1. For $LGS(30) = 1$ and $LOPT(4) = 0$, the imaginary frequency of the saddle point was not printed out correctly in the unit fu6 output file. This was because the program printed the frequencies before the imaginary frequency was read. This has been fixed by printing the frequencies after the imaginary frequency is read from the unit fu30 input file. The subprograms RPHRD1 and RPHRD2 have been modified. This bug was present since version 1.1.
2. For $LGS2(1) = 1$, and when the hindered-internal-rotator approximation was used, the program printed incorrect data in the unit fu6 output file because of a missing

argument in one of the WRITE statements in the subprogram READ5. This has been fixed by adding the appropriate argument. This bug was present since version 4.0.

3. In subprogram LCVMOD:
The variables EVIBM and VMD were not initialized in the subprogram LCVMOD. Due to this error, a non-zero value calculated by the subprogram LCVMOD was used when there should have been no contribution from a generalized normal mode to the correction term of the effective potential for LCG3 tunneling. This has been corrected by adding appropriate initialization statements. This bug was present since version 4.0.
4. The modification in the subprogram MEPSRT due to the simple perturbation theory treatment option introduced in version 5.0 was not put into the new subprogram MEPINV. (See version 5.0 revision history, items 4 and 10.) This has been fixed by adding appropriate statements in the subprogram MEPINV.
5. The forward symmetry factor SIGMAF in the unit fu5 input data files of the ch5 test runs has been corrected. This factor is now 2 in the test runs ch5j1tr5 and ch5j1tr6 and is 4 in the other ch5 test runs. As a result, the forward rate constants are a factor of 2 larger for the test runs ch5j1tr5 and ch5j1tr6, and a factor of 4 larger for other ch5 test runs than they were in version 5.0. Except for the test runs ch5j1tr5 and ch5j1tr6, this factor was correct in the unit fu5 input data files of the ch5 test runs before version 5.0.
6. This version and the previous version (5.0) of POLYRATE contain unpublished developers' versions of the VTST-IC algorithm which are not in its final form and may have bugs. So the VTST-IC feature should not be used in versions prior to 5.1.

Version 5.1 (December 1993)

1. The VTST-IC method, which was present in undocumented development version in version 5.0, has been included in the form documented in *Faraday Trans. Chem. Soc.* **90** (1994) 1715. It is now the recommended method if more accurate data are available for the reaction path information than the potential routine can provide. This is especially useful when an implicit semiempirical potential energy surface is used, e.g., in a MORATE calculation with version 5.1 or later of MORATE. Numerous changes have been made in several subroutines to implement the final version of VTST-IC, and the unit fu50 input formats have also been changed.
2. The input files for unit 50 for test runs cmctr2, nh3tr1, and ho2tr5 have been modified because of the changes mentioned in item 1. The calculated results are a little bit different from the previous versions.
3. The zero-order IVTST method described in *J. Chem. Phys.* **95** (1991) 8875 has been added to this version for bimolecular reactions. A new test run, ch4ohtr1, has been added in the distribution package to test this method. This option uses file fu50.
4. A new utility program, hrotor.f, which calculates the reduced moment of inertia of a hindered-rotor vibrational mode and the harmonic, free rotor, and hindered-rotor partition functions of this mode, has been added to the distribution package.
5. Two other new utility programs, findb.f and findl.f, which calculate range parameters for VTST-IC calculations, are also included in the distribution package.

6. Two other new test runs, ch5j1tr8 and oh3tr8, which are two of the test cases in *J. Chem. Phys.* **96** (1992) 5758, have been added to the distribution package.

Version 5.1.1 (March 1994) (Bug fixes)

1. In subprogram EXTRAP:
A bug in extrapolating the classical potential energy curve along the MEP on the product side has been corrected. This bug was introduced in or before version 1.5.
2. In subprogram DOREST:
A bug in printing the saddle point energetics for a restart calculation has been corrected. This doesn't effect any reaction rate constant calculations. The bug was introduced in version 5.0.1.
3. In subprogram ANHARM:
A bug in transforming between mass-scaled and unscaled Cartesian coordinates has been corrected. This only effects the calculations using an electronic structure input file (unit fu30), i.e., when LGS(30) = 1. This bug was introduced in version 5.0.
4. The printing of curvature components, which was reversed in the previous version, has been fixed.

Version 6.0 (June 1994)

1. POLYRATE 6.0 involved a complete change in the structure of the user input files previously numbered fu5, fu21, fu40, and fu50. The user-supplied data needed by the program is now all in FORTRAN units fu5 and fu50. The only other input file is unit fu30, which contains electronic structure information in lieu of a global analytic potential energy surface. The new unit fu5 and fu50 input files are in the keyword style.

For the unit fu5 input, the user options have been broken into nine sections: GENERAL, REACT1, REACT2, PROD1, PROD2, START, PATH, TUNNEL, and RATE. In each of these sections, one uses keywords to specify the parameters, calculation type, and variables for the calculation. (See the keyword section of the POLYRATE manual for complete details.) This restructuring of the input has added new capabilities to the program. One example is that it is now possible to converge the geometry in each of the stationary point calculations to a different tolerance and to use a different step size for each stationary point. This flexibility comes from changing the previously-used LGS options, which applied to the entire calculation, to keywords that are specified for each reactant, product, and saddle point in the system. This new type of input makes the program much easier to use. Most of the keywords have reasonable defaults. No input files, except those for unit fu30, from any previous version of POLYRATE will work with version 6.0 or later. The user should find that the inconvenience of converting existing input files to the new format will be offset by the ease of generation of future input files. All the test run input files, for both unit fu5 and unit fu50, in the distribution package have been converted to the new format.

2. Numerous new subprograms have been added in version 6.0 because of the new keyword input user interface. These new subroutines reside in the interface.f file in the source code directory. Subprograms MAIN, OPTION, OPT50, READ5, RESTOR, RESTRT, and ZOCPAR have been modified for the keyword input interface, and subprograms READ8, READ40 and READ50 have been removed.

3. The section of the unit fu6 output file that echoes the selected options has been completely rewritten. It now contains more details than earlier versions, and it is restricted to 80 columns. The part of unit fu6 output that echoes the VTST-IC calculation information has also been refined. This involves changes in the subprogram ZOCPRN.
4. The option [LGS(24)] for including the energy of product diatomics in endoergicity calculations has been removed. In addition, the program now has some built-in intelligence, and the user is no longer required to input quantities that POLYRATE can determine. For example, the number of frequencies, the mass of the individual atoms, the number of atoms in the system, the linearity of a species, the number of temperatures, and the number of activation energies are now all determined by the POLYRATE program.
5. The options LGS(18) and LGS(19) for special treatments for V_{MEP} and vibrational partition functions using M. Quack and J. Troe's method [*Ber. Bunsenges. Phys. Chem.* **81** (1977) 329] have been removed. Because of this, test run ho2tr2 and related files have also been removed.
6. New default values for the range parameters of the correction functions used in the VTST-IC calculation have been included in version 6.0. (See the subsection on the general description of unit fu50 input.) The previous method for calculating the default values is obsolete and is no longer supported. However, it is still strongly recommended that one use the utility programs findl.f and findb.f included in the distribution package to find appropriate values for these range parameters.
7. The normal mode analysis section in the unit fu6 output file for a diatomic Morse oscillator (ICODE = 2 in earlier versions or with the keyword DIATOM in this version) has been modified. The stretching vibration is now labeled as mode 1 instead of mode 6, which is consistent with the output for polyatomic systems.
8. The manual has been converted from an ASCII text format to Postscript format and fully updated to reflect the major changes in the input.
9. Machine-dependent scripts have been added for IRIS Indigo machines.

Version 6.2 (August 1994)

1. The POLYRATE long output file (unit fu6) has been significantly improved, and all the output lines have been restricted to a maximum of 80 columns. Numerous changes have been made in subroutines that contain output statements.
2. A new quantized-reactant-state tunneling method for unimolecular reactions has been introduced in this version of POLYRATE. This method can be used together with all the existing tunneling methods. The subroutines BOLTZ, INTAB, KAPVA, RATE, RTUNNL, and SETLGS have modified, and the new subroutines QRSENE, QTQVIB, RQRST, SIMPSN, and WKBENE have been added into the code. The INCLUDE files common.inc and param[1-4].inc have also be modified. The new keywords for this method can be specified in the TUNNEL section of the unit fu5 input file.
3. Three new output files have been added in this version; they are units fu25, fu26, and fu27. Unit fu25 contains the V_{MEP} and V_a^{G} information, and unit fu26 contains

generalized transition state vibrational frequency information; these new output files are especially useful for analyzing reaction-path data since they can easily be imported (with minimal editing) to a spreadsheet program for analysis or plotting. Unit fu27 contains the Cartesian coordinates of the generalized transition states in the XYZ input format of the XMOL program (which is *not* supplied with POLYRATE). The user can thus visualize the geometry changes along the reaction path by using the XMOL program, if available. The subroutines MAIN, FIOPEN, FCRATE, INTAB, RRATE, and SETLGS have been modified for this enhancement, and the new subroutines PRMEP and RPRMEP have been added to the code. The INCLUDE files common.inc and param[1-4].inc have also been modified for this enhancement. The new keywords for generating these new output files can be specified in the RATE section of the unit fu5 input file.

Version 6.5 (January 1995)

1. The POLYRATE input file (unit fu5) has been significantly altered in preparation to facilitate a more convenient incorporation of direct dynamics options. New sections, i.e., ENERGY, SECOND, and OPTIMIZATION, have been created. Numerous changes have been made throughout the program to localize all calls to the potential energy subroutine and calculations of first and second derivatives.

Version 6.5.1 (October 1995) (Bug fixes)

1. A bug in calculating the anharmonicity constant for the Morse model has been corrected. The local dissociation energy was calculated incorrectly when not using an analytic potential energy surface. This required modifications in the subroutine ANHARM. Also, when the dual-level dynamics method is used, the anharmonicity constant was not corrected using the higher-level data. This has been fixed by modifying in the subprogram ZOCUPD.
2. A subroutine call in subprogram RATE to the subprogram HRPART had an incorrect number of arguments. Also, one call to FITMAX in subprogram RATE had an incorrect number of arguments. These bugs have been fixed in this version.
3. A bug in subroutine DIATOM has been fixed so that the subprogram works correctly for unit fu30 input.
4. A bug in the extrapolation options in subroutine RPHINT has been fixed.

Version 6.6 (October 1995)

1. An option for performing generalized normal mode analyze in curvilinear coordinates has been added via the COORD keyword in the PATH section. For a linear or bent triatomic i-j-k, the curvilinear internal coordinate definitions are chosen automatically by the program; this triatomic option includes the i-j and j-k stretches and the i-j-k bends. For systems with more than 3 atoms, the INTDEF keyword can be used to define the curvilinear internal coordinates. Test run ho2tr2 was added for the $\text{OH} + \text{O} \rightarrow \text{HO}_2$ reaction. Test runs oh3tr9 and oh3tr10 were added for the $\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$ reaction. Details for these reactions are given in the section on sample test runs.

2. The zero- and first-order (the global approach) IVTST methods have been added, using fu29. (Recall that IVTST-0 has been available using fu50 since version 5.1, so now these are two ways to do IVTST-0.) Several test runs were added to the distributed test suite. The test runs ch5j2itr1 and ch5j2itr2 were added for the $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ reaction, the test run ch4ohtr2 was added for the $\text{OH} + \text{CH}_4 \rightarrow \text{H}_2\text{O} + \text{CH}_3$ reaction, and the test runs ch4cltr1 and ch4cltr2 were added for the $\text{CH}_4 + \text{Cl} \rightarrow \text{HCl} + \text{CH}_3$ reaction. Details for these reactions are given in the section on sample test runs.
3. The subroutine NEWT has been rewritten in order to support the quasi-Newton BFGS optimization method. This method can be used with both analytic surfaces and direct dynamics, and it is now the default for reactant and product optimization. Some changes related to this implementation have also been made in subroutines LINMN, SECCEN, and SECCEP. The keyword CONVERGENCE in the OPTIMIZATION section is no longer valid, and it has been replaced by the GCOMP and TSGCOMP keywords to emphasize that the convergence criterion for optimizations has changed.
4. A new sub-keyword, *nofreqcorr*, has been added to the unit fu50 data file for the keywords RCINFO and PCINFO. This sub-keyword can be convenient when higher-level data are unavailable for precursor or successor complexes. The subprograms READIC, OPT50, ZOCPAR, and ZOCUPD have been modified for this enhancement.
5. The code now runs on the Power Macintosh 7100/66 (Mac OS 7.5.1) computer using the LS FORTRAN compiler (version 1.1b1) available from Language Systems Corporation.

Version 6.6.1 (June-August 1996) (Bug fixes)

1. In file givtst.f:
Several errors are corrected in the IVTST calculations. In particular: the forward relative reduced mass is now calculated correctly for the evaluation of the relative translational partition function when unit fu29 is used for input. Also, the output of the zero- and first-order IVTST methods has been made more clear. The input energy of reaction is now converted to atomic units first, in order to prevent mistakes in the interpolation. The imaginary frequency of the transition state can be input as the last element of the TSFREQ section. The variables VAP and VAR are now used correctly in determining the lower limit of integration over energy in the transmission coefficient calculation. Most of the constants were moved to the INCLUDE file percon.inc for consistency. Finally, the CAG correction has been added. Information about evaluated curvatures has been added to the output.
2. In subroutine ENERG:
Version 6.6 had a bug in runs using the unit fu30 input file when one of the reactants or products is an atom. In such cases the subroutine SURF was called, but SURF should not be called in runs that use the fu30 input file. This has been fixed. Now, one can use the dumpot.f source code file as intended when using the unit fu30 electronic structure file input for a case when one of the reactants or products is an atom.

3. In subroutine RPHRD2:
We have corrected the order of fitting the gradients that are read from the unit fu30 input file. In the corrected version, the points along reaction path are always left-shifted even after the saddle point information has been inserted in the middle of the path. This bug had caused curvature calculations in the vicinity of the saddle point to be wrong.
4. In subroutine RFREQ:
This subroutine has been modified to allow the user to input the imaginary frequency for the ZCT calculation. In this case, the user must input the imaginary frequency as the last frequency of the saddle point.
5. In file main.f:
Anharmonicity is no longer allowed in TST-only calculations.
6. In test run ch4cl:
The data for the electronic partition function of the Cl atom has been changed to the correct values. The degeneracy of the ground state has been changed, and the first excited state has been included with correct atomic spectral data.
7. In test runs ch4ohtr1 and ch4ohtr2:
A comment is added for the reduced moment of inertia in the unit fu50 input file in the ch4ohtr1 test run. A more detailed description of keywords RMITS and RMIRP has been added to the manual with an example of how to include the symmetry number for the hindered-rotor approximation. The symmetry number SIGMAF in ch4ohtr2 has been changed to 12. (Notice: the hindered-rotor approximation is not supported when using the unit fu29 input file.)
8. In subroutine TPCDSC:
This subroutine was fixed to prevent the occurrence of a zero-divided-by-zero case.
9. In subroutine DEFSTA:
MODER(i,j) was changed to MODER(j,i) for correct dimensioning. This change will initialize the array as intended. At the beginning of the subroutine, the elements of this array are set to zero.
10. In test runs oh3tr6 and oh3tr10:
Unused data sections were removed from the input file.
11. In files dattim.ibm and dattim.hp:
These files have been modified to remove the NULL characters which generated unsightly hardcopy.
12. In subroutine ZOCPRN:
The title line for the frequency output of the lower-level and higher-level methods for the VTST-IC method has been clarified. The columns were labeled 'Point 1' and 'Point 2' even though MEPTYPEP or MEPTYPEP was set to *two* or *one*; however, the code was printing a sorted list of the combined frequencies. Now the title has been changed to 'Sorted Reactant' and 'Sorted Product' when there are no wells.
13. In subroutine NOROUT:
The variable SPRNT has been replaced with S in order to make sure the keyword SPRNT works properly.

14. In subroutines TSRATE and RATE:
The unused variable LGS(19) inside these routines has been removed since LGS(19) is not used in the current version of POLYRATE.
15. In file common.inc:
The common block AMASCM has been added for consistency with other blocks.

Version 7.0 (September 1996)

1. The keywords ICR and ICL were added for new interpolated correction methods. The following changes were made: (1) A new variable IVICM has been added in common.inc. (2) The subroutine MUCDSC in polymq.f, subroutines ZOCAST, ZOCPAR, and ZOCUPD in polysz.f, and subroutines INTAB, RICOPT, SURF, RVAR6, SETLGS, WVAR6 in interface.f were modified for the new methods. Also, LGS2(10) is used internally to store the choice between the ICA, ICL, and ICR options.
2. A new option for electronic structure input data has been added (POTENTIAL = *unit40*). This option reads the data from unit fu40 in keyword form rather than from unit fu30 in formatted records.
3. In subroutine RPHSET:
In version 7.0, the saddle point geometry is no longer read from both unit fu5 and unit fu30; it is only read from unit fu5. All the unit fu30 input files distributed with the present version of the code have no saddle point geometry. A warning message is printed when the NEXTPT option is used with an option code different from 0, and a stop message is printed after the next point is found.
4. In file interface.f:
Options *quad* and *page* for the keyword CURV have been changed to *dgrad* and *dhess* respectively. The PAGE keyword for activating the Page-McIver integrator has been changed to PAGEM. All these changes require that the user modify the input files prepared since version 6.0. All the examples included in the test run suite whose unit fu5 input files make use of these options have been modified accordingly.
5. In the testrun/ch5 directory:
New test runs are added (ch5fu30tr1, ch5fu30tr2, ch5fu30tr3, ch5fu30tr4, ch5fu30tr5, ch5fu30tr6, ch5fu29tr1, ch5fu29tr2, ch5fu40tr1, ch5fu40tr2, ch5fu40tr3, ch5fu40tr4, ch5fu40tr5, ch5fu40tr6, ch5icfu30tr1, ch5icfu30tr2, ch5icfu40tr1, ch5icfu40tr2, ch5fu50tr1 and ch5fu50tr2) to demonstrate the different input methods allowed in the current version of POLYRATE for electronic structure data.
6. In the testrun/oh3 directory:
The names of some of the existing test runs have been changed (oh3tr6 and oh3tr10 have become oh3fu30tr3 and oh3fu30tr2, respectively; oh3tr7, oh3tr8 and oh3tr9 have become oh3tr6, oh3tr7 and oh3tr8, respectively). New test runs to illustrate the use of direct dynamics calculations with unit fu30 and unit fu40 input files have been added (oh3fu30tr1, oh3fu30tr4, oh3fu30tr5, oh3fu40tr1, oh3fu40tr2, oh3fu40tr3, oh3fu40tr4, oh3fu40tr5).

7. The manual has been improved with the following major changes: (1) A new section describing the reaction-path curvature calculations has been added. (2) The section describing the reaction rates including tunneling calculations has been improved. (3) A better description of the VTST-IC method has been included in the sections on theoretical background for dynamical calculations and selected details of the calculations. (4) The description of extrapolation methods has been improved. (5) The description of the unit fu30 input file has been improved. (6) A new section has been added to explain the use of the unit fu40 input file for direct dynamics calculations. (7) A summary has been added to the sample test run section. (8) The section summarizing subprograms has been updated. (9) New errata sections have been added for published papers referenced in the footnotes to the text and bibliography sections. (10) More consistent font choices have been used to distinguish subprogram names, variables, keywords, and keyword values.
8. The MDMOVIE keyword has been added in the GENERAL section to generate a file (as unit fu65) for the MDMOVIE program. This change involved changes in the following subroutines: DEFGEN, INTAB, RGENER, RVAR6, WVAR6, and DEFPAT. A new variable, IMDMOV, was added in the file common.inc.
9. In file set_machine.jc:
This job control file has been modified for more choices of machines. The compiler options for most of the platforms are for the O3 optimization level, and a new set of scripts for the IBM RS/6000 Model 590 machine is available.
10. The output of POLYRATE has been cleaned up so that no line exceeds 80 columns, the respective units are included for each value printed, clearer headings are printed together with the numerical information, timing routine prints times to 0.01 seconds instead of 0.0001 seconds and with a more clear format, and the debug information in the VTST-IC section is removed. To print the debug information, remove the comment from the CALL ZOCASST line in the ZOCUPD routine.
11. In file givtst.f:
Improved output is provided for several parts of the IVTST calculations. A table containing the reaction energetics is printed. The parameters for the interpolating functions match the notation used in *J. Chem. Phys.* **95** (1991) 8875. The transmission probabilities are better labeled and the reaction-path curvature results are printed in first-order IVTST calculations. Formulas used in subroutines BCALP and MUSCCD have been modified to follow the notation used in the references. The DELTAH keyword has been removed from the unit fu29 input file.
12. In subroutine BANGLE:
This subroutine has been renamed BANGLE1 to avoid using the same name as is used in the MOPAC program.
13. In subroutines RPHRD2 and RPHDXN :
The option choices $LOPT(3) = 99$, $LOPT(3) = 100$, and $LOPT(4) = 99$ have been added for the unit fu30 input file.
14. In subroutine KAPVA:
A new table is printed in the long output file to display the values of the transmission coefficients obtained with all the possible combinations of the number of quadrature points used in the evaluation of the theta and Boltzmann integrals.

Version 7.0.1 (February 1997) (Bug fixes)

1. In file `poten/ch5.f`:
The data parameter `DIJ` has been changed from 6.27396 to 3.1190 to correctly correspond to *J. Chem. Phys.* **87** (1987) 7024.
2. In subroutine `REACT`:
All occurrences of `LGS(24)` are changed to `LGS2(4)`. This was an error.

Version 7.1 (February 1997)

1. In files `interface.f` and `polyag.f`:
A keyword option `REORDER` has been added to the `PATH` section. The default is set to `NOREORDER`, which forces all calculated frequencies to be in canonical order along the `MEP`. Previous versions of `POLYRATE` have the default set as `REORDER`, where the frequencies are ordered according to overlap calculations. The `DEFPAT`, `INTAB`, `RPATH`, `RVAR6`, `WVAR6`, `NOROUT`, `RFREQ`, `GETFREQ`, `RPHRD2`, `IRFGT0`, and `FDIAG` subroutines were modified, and a new variable, `ireord`, was defined in file `common.f`. Note: All calculated frequencies in the `FREQ` array are stored in canonical order (if `NOREORDER` is used) except at the saddle point, where we store the imaginary frequency in position 1.
2. In file `poly40.f`:
The default unit for frequencies read from unit `fu40` has been changed from atomic units to wavenumbers. The keyword `FREQUNIT` controls the units of the frequencies; values are *waven* for wavenumbers (default value) and *au* for atomic units.
3. The file `dattim.ibm` has been revised to obtain the date and time without the need for `dateclock.ibm`.
4. In file `interface.f`:
In the subroutine `RMUVT`, the variables `FIT1` and `FIT2` were changed to `IFIT1` and `IFIT2`.
5. A change was made to subroutine `SADDLE` to correct the zero of energy in calculations where an electronic structure program is being used in place of an analytic surface or a file of data.
6. Other changes were made to the `NORMOD`, `POLYAT`, `REACT`, `DORPH`, `FCMEP`, `CENTER`, `DOREPR`, `RATE`, `SADENG`, and `HEADR` subroutines. In general, these changes were to make the linking of `POLYRATE` with various electronic structure programs more seamless or to clean up old code.
7. In subroutine `FINOUT`:
Lines containing `LGS(13)` are removed because this option is no longer used.
8. In addition, several changes were made that will be supported by later versions of `POLYRATE` or in `ACESRATE` or both. These changes should not affect users of `POLYRATE`-version 7.1 who do not have the `ACESRATE` interface, but they are listed here for completeness.

- 8.1. Specialized ACESRATE hooks were added to subroutines ENERG, FIRST, GEOMOP, PREP, and SECDER and to files common.inc and param*.inc, and the following subroutines were modified to read and process ACESRATE keywords: DEFENG, RACES, READ5, RENERG, RGEOM, RGSAD, RGZMAT, RSECND, RZVAR, DEFOPT, DEFPAT, DEFSTA, INTAB, INTABS, RGSZM, RICOPT, ROPT, RSTAT, RVAR6, SETLGS, WVAR6, FIOPEN, and DOREST.
- 8.2. The restart option was enhanced. In earlier versions of POLYRATE for a two-level (VTST-IC) calculation any change in the higher-level potential energy surface required the recomputation of the entire lower-level minimum energy path. When this restart option is supported, it will be possible to read the lower-level path from a unit fu1 restart file. MAIN was also restructured to make restart and direct dynamics calculations easier and more flexible.
- 8.3. The CALCMUFF keyword option was added. When this is supported the reduced mass will be calculated when the reaction path is computed. In previous versions of POLYRATE, the effective reduced mass was only computed if tunneling was also selected. This means that in a restart calculation, if the original run did not include a tunneling calculation, tunneling could not be used in any later calculation without recomputing the entire reaction path.

Version 7.1.0.5 (March 1997)

1. A fourth switch keyword, GAUSSIAN, is added to the OPTIMIZATION and SECOND sections of the unit fu5 input file. The four switch keywords POLYRATE, MOPAC, ACES, and GAUSSIAN are used by GEOMOP and SECDER. The ROPT and RSECND subroutines are modified for this update. This is a temporary measure so the first version of GAUSSRATE can be implemented with minimal changes to the structure of version 7.1.

Version 7.1.1 (March 1997) (Bug Fixes)

1. In files energetics.f, common.inc, and interface.f:
The keyword SADDLE is introduced to indicate whether the reaction has a saddle point or not. The following subroutines are modified: SADDLE, SADENG, DEFPAT, INTAB, OPTION, RPATH, and SETLGS. To perform the calculations with NOSADDLE correctly, a new *gradi* option of the keyword FIRSTSTEP has to be used to obtain the first point by following the gradient direction. Input files of the test runs ho2tr1, ho2tr2, and ho2tr3 were changed due to this update. In versions 6.0-7.1.0.5, the NOOPT keyword was correct only for reactions without a saddle point. Thus the NOOPT keyword was equivalent to NOOPT, NOSADDLE and FIRSTSTEP = *gradi*.
2. In subroutine SECDER:
The subroutine calls for evaluating the second derivatives of the potential with *quadratic* and *fourth* are corrected for the right number of frequencies. This corrects a mistake that was introduced in version 6.5.

Version 7.1.2 (March 1997) (Bug Fixes)

1. In subroutines RPHDXN, RPHRD2, CVCOOR, and file poly40.f:
The transformation from rectilinear to curvilinear internal coordinates requires the unnormalized gradients. If normalized gradients are given in the unit fu30 and unit fu40 input files, there is no way to perform the transformation from rectilinear to curvilinear internal coordinates. Incorrect statements in the subroutine CVCOOR were removed.
2. In files ch5fu30tr*.fu30, ch5fu40tr*.fu40, ch5icfu30*.fu30, and ch5icfu40*.fu40:
The electronic structure program GAUSSIAN94 gives the Cartesian gradients instead of Cartesian forces in the formatted checkpoint file (even though the output says Cartesian forces). Test runs prepared based on the forces taken from GAUSSIAN94 are corrected. Also, the SIGN of unbound eigenvector at the saddle point is set to *product* instead of *reactant*.
3. In files percon.inc and givtst.f:
The constant CALAU is removed from POLYRATE to prevent confusion with the parameter of the same name in MOPAC.
4. In the ch4cl* test runs:
The geometries of the stationary points in the input file have been changed. The values have been wrong since version 6.6.
5. In subroutine NEWT:
The output statements for the optimization results are corrected.

Version 7.2 (April 1997)

1. In subroutines REACT and SETLGS:
A portion of the program that calculates the zero of energy has been rewritten. The option *reactants* no longer exists in the current version of program. The zero of energy is determined automatically after the reactants are evaluated unless the *read* option is chosen for the EZERO keyword. Changes were made in test runs ch5j1tr4.dat, clhbrtr1.dat, cwmctr2.dat, cmctr1.dat, cmctr2.dat, ho2tr4.dat, ho3tr5.dat, nh3tr1.dat, oh3tr1.dat, oh3tr7.dat, oh3tr8.dat, and oh3tr3.dat.
2. In file intbsv2.f:
The transformation from rectilinear coordinates to curvilinear internal coordinates of doubly degenerate linear bends has been added. Changes were made in subroutines CVCOOR, CONOUT, ICODAL, BMAT, BIMAT, and BTENS, and in the common.inc file. To use this option, the '=' symbol is used to denote the linear bends in the INTDEF keyword option. The input files of the ch5j1tr4, ch5fu30tr6, and ch5fu40tr6 test runs are modified to illustrate this new curvilinear internal coordinate option.

3. In files `interface.f` and `polysz.f`:
New interpolated corrections options have been implemented. First, a difference of two Eckart function fits is used to interpolate the correction to $V_{\text{MEP}}(s)$. The new keyword option is called ECKART. The following subroutines have been modified: DEFPAT, INTAB, RICOPT, ZOCPAR, and ZOCVCL. Second, two improved options are available for interpolating corrections to the frequencies. The original IC method is denoted as ICA, and the two new options are called ICL and ICR. The combination of the ICL option with the ECKART option is taken as the new default. To perform calculations with the original IC method, the NOECKART and ICA keywords are required. The `ch5icfu30tr1` and `ch5icfu40tr1` test runs are used as examples of the new dual-level ICL/ECKART option. The input files for the `ch4ohtr1`, `cmctr2`, `ho2tr5`, `ch5fu50tr2`, and `nh3tr1` test runs are changed to accommodate the new default setting.
4. In files `polysz.f` and `comp_src.*`:
The BLAS library routines are moved into a separate file, `fromblas.f`, from the main POLYRATE program. If their machine has a pre-installed BLAS library, users can modify the script `comp_src.jc` to include the machine library which yields the best performance.
5. In subroutine FINDL:
The subroutine FINDL, previously distributed only as a stand-alone utility, is now also added to the `polyag.f` file as a subroutine. This subroutine will automatically determine the L parameter for VTST-IC calculations (the value of the keyword RPL in the unit `fu50` input file) in any single-level reaction path calculation with a positive barrier.
6. In files `interface.f`, `hooks.f`, and `polyag.f`:
The keyword WRITEFU30 is added to the GENERAL section of the unit `fu5` input file to create a `*.fu30` file from a calculation using the hooks. (This option is mainly useful to developers for creating test runs.) Test run `oh3tr7` is modified to demonstrate this option. The OPENFI and RPHWRT subroutines have been modified.
7. In file `givtst.f`:
The RHESS, RGRAD, and GIVTST subroutines have been changed to allow free formatted geometry, gradients, and Hessian input from the unit `fu29` file.
8. In file `interface.f`:
A new keyword, EZUNIT, is added to allow the zero of energy to be input in kcal/mol. A new subroutine, REZERU, and a variable, IEZERU, are added, and the input file for `clhbrtr2` is changed for this new feature.
9. The keywords SIGMAF and SIGMAR are no longer saved to file `fu1` for restarts. This allows the user to change their values in the RATE section of the restart run.

10. Subroutines ENERG, FIRST, SECDER, and GEOMOP have been renamed EHOOK, GHOOK, HHOOK, and OHOOK, respectively. The options for the potential keyword were changed and the following subroutines were modified: DERIV2, REACT, DIATOM, DOSAGE, F1DIM, LCG3TH, PHSINT, QQPOT, WKBVIB, DERV24, INTPM, NEWT, NORMOD, SECCEN, SECCEP, ANHARM, ANHER, GRAD, TP, INTFNC, CUBST, INTPM, NEWT, NORMOD, POLYAT, REACT, SADDLE, and DOREST. A new keyword, HESSCAL, was added to provide a choice between calculating second derivatives with the GHOOK or HHOOK subprograms. In the SECOND section, HESSCAL = *ghook* replaces the POLYRATE keyword, and HESSCAL = *hhook* replaces the MOPAC and GAUSSIAN keywords. The keywords POLYRATE, MOPAC, GAUSSIAN, BFGS, and TSBFGS in the OPTIMIZATION section were replaced by OPTMIN and OPTTS. Subroutines RHESSC and ROPTOP were added to parse the options.
11. The option *analytic* is replaced by *hooks* for the keyword POTENTIAL in the ENERGETICS section of the unit fu5 input file.
12. In subroutine RPHINT:
The interpolation of the curvature components from the unit fu30 or the unit fu40 basic input grid to the save grid (see the section on methods for interpolation and extrapolation of electronic structure data for a description of these grids) is now based on the absolute value instead of signed values in order to avoid problems related to the continuity of the eigenvector sign convention.
13. In subroutine GFDIAG:
When using curvilinear coordinates, a check is added to set the eigenvalue to zero if a value corresponding to a vibration is smaller than 10^{-18} .

Version 7.3 (July 1997)

1. Three new keywords are added to increase modularity, and as a consequence three new variables are added to the file common.inc. The variable ISUP is used to toggle the keyword SUPERMOL (in the GENERAL section), the variable INITG is used to indicate the option chosen by the INITGEO keyword (in the REACT1, REACT2, PROD1, PROD2, and START sections), and the variable INTMU (for the INTMU keyword in the PATH section) is used to specify the number of points used for the interpolation of the reduced mass at the saddle point. All three variables replace the LGS(35) value of the previous version. For ISUP, the following subroutines are modified: DEFGEN, INTAB, RGENER, SETLGS, NORMOD, POLYAT, REACT, SADENG, and CENTER. For INTMU: DEFPAT, INTAB, RPATH, BCALC0, ELRPH, and PATH are modified. For INITG: REACT, DEFOPT, INTAB, RGEOM, RGSAD, ROPT, DOREST, FCMEP, and FIOPEN are modified. Also, the IPOT values are modified as follows:

value	old meaning	new meaning
0	<i>hooks</i>	<i>hooks</i>
1	<i>mopac</i>	<i>unit30/unit40</i>
2	<i>aces</i>	<i>aces</i>
3	<i>gamess</i>	<i>unit29</i>

A new switch keyword, SUPERMOL (and its negation, NOSUPERMOL), has been added to the GENERAL section to indicate that the supermolecule mode is used for reactants and products. The default is on (SUPERMOL). The choice has no effect if there is only one reactant and one product. The following subroutines were modified: CENTER, DEFGEN, INTAB, NORMOD, POLYAT, REACT, REDGEO, RGENER, and SETLGS.

A new variable keyword, INTMU, has been added to the PATH section to indicate the number of points on each side of the saddle point used for the interpolation of the effective reduced mass at the saddle point. The available options are 1 and 3. Thus, values of 1 and 3 indicate that interpolation is based on 2 and 6 points, respectively. The default is 1. The following subroutines were modified: BCALC0, DEFPAT, ELRPH, INTAB, PATH, and RPATH.

A new variable keyword, INITGEO, has been added to the REACT1, REACT2, PROD1, PROD2, and START sections to indicate the location of the initial geometry. The available options are *geom* and *hooks*, the default being *geom*. The following subroutines were modified: DEFOPT, DOREST, FCMEP, FIOPEN, INTABS, POLYAT, REACT, RGEOM, RGSAD, and RSTAT.

2. The switch keywords NONLINEAR and LINEAR and the variable keyword PHASE have been incorporated into the single variable keyword SPECIES. The default has been set to *nonlinrp*. The following subroutines were modified: DEFSTA, INTABS, REDGEO, RSTAT, RVAR6, and WVAR6.
3. Subroutine REACT:
If the option *read* of the EZERO keyword in the potential section is used, only the geometries of the reactants are obtained. In the case of INITGEO = *hooks*, optimizations are required because the initial geometries are obtained through the OHOOK subroutine. Since MORATE-version 7.3 requires using INITGEO = *hooks*, the NOOPT option is not implemented with MORATE.
4. A new keyword is added to the REACT1, REACT2, PROD1, PROD2, and START sections. The LINAXIS keyword is used to indicate that a molecule is linear and to specify its orientation. There are three options: *x-axis*, *y-axis*, and *z-axis*. The default option is *z-axis*. A new array, IOLIN, is added to the file common.inc. This array is used in subroutines DEFSTA, RSTAT, CENTER, RVAR6, INTABS and WVAR6.
5. An option to print the distance matrix was added, and a new variable keyword, PRDISTMX, has been added to the PATH section to indicate the interval along the save grid for the printing of interatomic distance matrices. The default is to print the interatomic distance matrix at every 20th point along the save grid. The following subroutines were modified: DEFPAT, INTAB, NOROUT, and RPATH.
6. In subroutine ZOCPAR:
The default value of the keyword RPL was calculated from the lower-level surface inversion 7.2; from version 7.3 on, the default is set to the value calculated from the higher-level calculations.
7. An option to calculate the vibrational partition functions with vibrations treated classically has been added, and a new switch keyword, CLASSVIB, and its negation, NOCLASSVIB, have been added to the GENERAL section to allow the user to choose between classical and quantal vibrations. The default is to use quantized vibrations. The following subroutines were modified: DEFGEN, INTAB, OPTION, and RGENER.
8. In subroutines ICFDIAG and PROJF:
An unnecessary calculation of the **G** matrix has been removed in order to obtain better performance for the transformation from rectilinear to curvilinear internal coordinates.

9. In subroutines NORMOD and HHOOK:
Two GHOOK subroutine calls with an HHOOK call between them have been removed to eliminate unnecessary work. The HHOOK subprogram is modified to return the gradient at the same time as the Hessian, which is now the expected way for all HHOOK subprograms to be written.
10. The FORTRAN file units fu31-fu36 have been moved to fu71-fu76, and the FORTRAN file units fu80-fu89 have been reserved for use with interface codes.

Version 7.3.1 (July 1997) (Bug fixes)

1. The line “if (icode(jtype).lt.0) lgs(34) = 1” was missing from the interface.f subprogram in version 7.3. This only affects solid-state runs.
2. In the oh3fu30tr3 testrun the value of the SPECIES keyword for the saddle point in version 7.3 is *lints*, and this has been corrected to *nonlints*.

Version 7.4 (August 1997)

1. The option *curv3* is added by making modifications to the file common.inc, and to the OPTION, SETLGS, VECCON, CVCOOR, and READINT subroutines, plus adding the new subroutines BIMAT2, MTINV2, GFDIA2, SVBKSB, and SVDCMP. This option is designed for the transformation from Cartesian coordinates to redundant internal coordinates for frequency evaluations at generalized transition states. The maximum number of internal coordinates is $3N + 6$. This new option has been used in the test runs ch5j1tr4, ch5fu30tr6, ch5fu40tr6, and ch5j1tr4. Furthermore the following test runs have been changed to use the *curv1* option: clhbrtr1, ho2tr2, and h3tr3.
2. The unit fu61 file has been created to save the stationary point information so it can be used in a subsequent run, if desired. This effectively creates a new restart capability, as discussed in the section on restart options. After the calculations are done on a given reactant or product, the geometry, energy, and frequencies of the reactant or product are written to the unit fu61. Similarly, for a saddle point, the geometry, energy, frequencies, Hessian, and eigenvectors are written. A new variable keyword, STATUS, is added in the *REACT1, *REACT2, *PROD1, *PROD2, and *START sections with several options.

Other keywords added to support this new option are ENERGY, FREQ, and EIGENVECTOR. The subroutines modified are DEFSTA, INTABG, INTABS, OPTION, RSTAT, RVAR6, STVARJ, WVAR6, DOREPR, FCRATE, FDIAG, FIOPEN, DOSAFR, NORMOD, POLYAT, REACT, SADDLE, and SADENG. A new subroutine WRTHOK is added. The keyword option OPT is removed, and FREQ is changed to a list keyword.

The subroutines SETLGS, DORPER, and MAIN have been modified to allow POLYRATE to stop correctly after calculating one or more stationary points.

3. The keywords ICA, ICL, and ICR are replaced by options for the new keyword option FREQ in the ICOPT list. A new option, *none*, is added for no frequency corrections in the VTST-IC method. Modifications were made in the ZOCAST, ZOCPAR, ZOC3P, INTAB, and RICOPT subroutines.

4. In the file hooks.f:
Duplicate copies of the subroutines DERIV2, DERV24, NEWT, SECCEP, SECCEN, and TRANS were renamed YDERV2, YDER24, YNEWT, YSECEP, YSECEN, and YTRANS respectively, and they were added to the hooks.f file to increase modularity.
5. The default for the FREQMAT keyword in the unit fu50 file has been modified for all cases. From now on, the default is to match the frequencies of a unimolecular reaction with wells in descending order. Modifications were made in the OPT50 and ZOC3P subroutines.
6. The switch keywords PRSAVEMODE and PRSAVERP have been changed into variable keywords with default values set to 9999. This means that after every 9999 save grid points, the information on the save grid will be printed instead of after every save grid point. Modifications were made in the file common.inc and in the NORMOD, RPATH, INTAB, DEFPAT, and CVCOOR subroutines.
7. The two extra gradients given in the unit fu30 files for computing the curvature were placed before the Hessian matrix in previous versions of POLYRATE, but have been moved after the Hessian matrix in version 7.4. Modifications were made to the RPHRD2 subroutine.

Also, in previous versions of POLYRATE the two extra gradients were always printed out in mass-scaled coordinates. From now on, the two extra gradients will be printed out in the same coordinates as the gradient. The subroutines PATH, RPHDXN, RPHWRT, and RPHRD2 have been modified to allow for this change.

8. The subroutine FIOPEN has been modified to allow POLYRATE to write the unit fu25, fu26, fu27, and fu30 files at the same time. This modification allows the PRPATH and WRITEFU30 options to be used together.
9. When PRPATH is used, the numerical values of the CD-SCSAG effective reduced mass and of kappa and the determinant of the moment of inertia tensor are written to the unit fu25 file. The following subroutines have been modified: MEPINV, MEPSRT, PATH, SSAVE, BCALC, COLSHF, EXTRAP, and TSRATE.
10. The SGI Power Challenge option in the set_machine.jc script has been divided into two options, one for the SGI R8000 chip set and one for the SGI R10000 chip set.
11. New scripts have been added to support the Linux operating system.

Version 7.4.1 (September 1997) (Bug fix)

1. In files polyrr.f, poly40.f, and common.inc:
When using input units fu30 or fu40 with hindered rotor anharmonicity, the reduced moment of inertia for hindered rotations along the reaction path was set to a fixed value for all the points along the path. Now, the value is interpolated using Lagrangian interpolation and extrapolated using exponential functions in the same fashion as the moment of inertia. The following subroutines have been modified: GENSEC40, IRFGT0, RPHFIT, RPHINTM RPHRD2, RPHSET, and SHFTSAD.

Version 7.4.2 (October 1997) (Bug fix)

1. Subroutine ZOPAR and ZOCPRN:
An error in the ICL algorithm has been corrected. The error involved using the energy ratio of frequencies (0 instead of 1) for the case where both the high- and lower-level frequencies go to zero at reactants or products.

Version 7.4.3 (October 1997) (Bug fix)

1. Subroutine ROPTOP:
The dummy argument ISTRT in the subroutine's formal-argument-list has been misspelled, ISTRTI, since version 7.2.

Version 7.4.4 (December 1997) (Bug fixes)

1. When the geometries in files fu30 and fu40 were given in mass-scaled coordinates, the saddle point geometry, given in file fu5 in unscaled coordinates, was not mass-scaled when it was used with units fu30 and fu40 and therefore, the saddle point geometry was incorrectly used as mass-scaled. This leads to the wrong moment of inertia and the wrong TST rates. The subroutines RPHRD2 and SADSEC40 have been modified to fix this bug.
2. When using the WRITEFU30 option with LOPT(6) = 3 or 4, the extra gradient before the Hessian point was being printed out incorrectly. The extra gradient was un-normalized using the wrong norm. This has been corrected by modifying the PATH subroutine.

Version 7.8 (December 1997)

1. The IVTST-M algorithm has been implemented. For this purpose, the module ivtstm.f has been added, the variables N3S31 and NPT31 have been added into the percon.inc INCLUDE file, and several variables or arrays have been added to the common.inc INCLUDE file. The subprograms added or modified are SPL31, SPLM31, SPLV31, MUCDSC, PSAG, IMOM40, SADSEC40, RPHINT, and RPHRD2.
2. A new input unit, named fu31, has been added. The keyword WRITEFU31 in the GENERAL section and the option *unit31* for the POTENTIAL keyword in the ENERGETICS section of the fu5 input file have been added. The module poly31.f has been added for this change, and the INCLUDE files param.inc, percon.inc and common.inc and the PATH, DOPNM, DOSAFR, FCMEP, FIOPEN, POLYAT, REACT, DEFGEN, INTAB, RLINE, RGENER, RPOTET, SETLGS, RPHINT, and RPHWRT subprograms have been modified.
3. The original extrapolation algorithm for the information in the input units fu30 and fu40 has been removed. The subprograms EXTRP40, RD40EXTR, RD40GEN, RPH40, and RPHRD1 have been modified to reflect these changes.
4. The code has been modified in order to calculate wells in the reactant and/or product channels. The sections WELLR and WELLP have been added in the unit fu5 and WR40 and WP40 in unit fu40. Several arrays in the INCLUDE file common.inc and the variable NARRS in percon.inc have been modified, as well as the subprograms WRTHOK, PATH, DOREPR, DORPH, REACT, DEFT40, EXTRP40,

IMOM40, IRFGT0, PTSEC40, RD40EXTR, RD40RP, RPH40, RPSEC40, SADSEC40, DEFSTA, INTAB, INTABG, INTABS, OPTION, READ5, REDGEO, RGEOM, RSTAT, RVAR6, RVARY, SETANH, WVAR6, RPHINT, RPHRD1, RPHRD2, RPHSET, and RPHWRT. The subprogram WRTWEL has been added.

5. The switch keyword RODS has been added to the PATH section of file fu5 to determine if the RODS algorithm is used. The DEFPAT, DOSAFR, INTAB, NORMOD, NOST31, PATH, RPATH, RPHRD2, RVAR6, and WVAR6 subroutines have been modified, the IRODS, VTT, and XNOU variables have been added to common.inc, and a new subroutine RRPM has been added to interface.f.
6. The keywords INM and SSAVE have been removed from the PATH section of file fu5. This effectively makes the Hessian grid and the save grid the same. Therefore, when POLYRATE calculates the reaction path, the Hessian is saved whenever it is calculated. The Hessian grid step size is now controlled by the INH keyword in the path section. The DEFPAT, ELRPH, INTAB, IVTST0, NEXTPT, OPTION, PATH, QRSENE, RESTOR, RESTRT, and RPHWRT subroutines have been modified accordingly.
7. The CALCS keyword has been added to the path section of file fu5. This new keyword determines how the value of s is calculated when using the RODS algorithm. The INTAB, RPATH, RVAR6, and WVAR6 subroutines have been modified, and the ICALCS variable has been added to common.inc.
8. A new keyword CUS has been added into the RATE section for evaluating the reaction rate constants with the canonical unified statistical model. The file COMMON.INC and the following subroutines were changed: RATE, FINOUT, DEFRAT, INTAB, and RRATE. In addition, three new subroutines were added: CUSSPL, ISWAP, and DPSWAP.
9. The keyword GEOMUNIT has been introduced in the GENERAL section of the fu5 input file to specify the units for the geometries of the stationary points read from the fu5 input file. The DEFGEN, INTAB, RGENER, RGEOM, and RGSAD subroutines have been modified and the RRGU5 subprogram has been added. The IGEOU5 variable has been added to the common.inc file.
10. The subroutine TABLE has been modified in order to print the ZCT results in the fu15 output file.
11. The subroutine RPHWRT has been modified in order to write a cleaner fu30 input file.
12. Subroutine CENTER:
This subroutine has been modified to clarify the output and reduce the amount of the output. The POLYAT, SADENG, IMOM40, and RPHRD2 subroutines have also been modified for this purpose.
13. The frequency index for files fu30 and fu40 has been changed so that index 1 corresponds to the largest frequency at the saddle point, index 2 to the second largest frequency, and so on. The subroutines DEFLT40, RD40RP, and RPHRD2 have been modified accordingly. In addition, the subroutine RPHWRT has been modified so that when the writefu30 keyword is specified the generated unit fu30 file is consistent with this frequency numbering.

14. The following test runs have been added: ch5fu31tr1, ch5icfu31tr1, oh3tr9, oh3tr10, oh3fu30tr6, oh3fu31tr1, oh3fu40tr6, and ch4otr1.
15. A large number of minor corrections and clarifications have been made in the manual.

Version 7.8.1 (February 1998) (Bug fixes)

1. When using the INTMUEFF option with IVTST-M, the values of μ_{eff} at the first point on the reactant side were incorrectly stored as the values of μ_{eff} at the first point on the product side and vice versa. This has been corrected in subroutine ARMUEF.
2. The moment of inertia and the reduced moment of inertia were both incorrectly interpolated asymptotically to zero instead of asymptotically to infinity when using IVTST-M. The subroutines SPLRMI, SPLNM, and SPLM31 have been modified to reflect this correction.
3. The reduced moment of inertia of the saddle point was incorrectly stored as the product well when using unit fu30. This has been corrected in subroutine RPHRD2.
4. In runs which WRITEFU30 or WRITEFU31 is selected, along with the option for printing the Hessian in unscaled coordinates, the unscaled Hessian was correctly written out but then it was incorrectly substituted for the mass-scaled Hessian in the rest of the run. Subroutine RPHWRT has been corrected to eliminate this error.
5. In the subroutine MEPINV the variable SBKAP was incorrectly specified as SBAP. This typo caused the wrong curvature to be printed to the fu26 file.
6. The check_test.jc script has been modified to avoid checking both ch4o and ch4oh test run files when the ch4o test run files are to be checked.
7. The compiling script comp_src.linux has been corrected to compile the dateclock.c subroutine.
8. In subroutine RPHRD1, a change was made in the calculation of the reaction-path curvature at the points adjacent to the saddle point. This was later found to be incorrect and was removed in version 7.9.3.

Version 7.8.2 (March 1998) (Bug fixes)

1. The subroutine NORMOD has been modified to fix a run-time error which occurs on SUN workstations.
2. The hindered rotor option has been enabled for the reactants, products, and wells. The subroutines modified are SETANH, NORMOD, REACT, HINDRT, and ANHARM and new arrays were added to common.inc.
3. When the hindered rotor approximation was used with unit fu31 the transformation from Cartesian coordinates to mass-scaled coordinates was incorrect. As a consequence, the reduced moment of inertia of the saddle point was miscalculated.

The subroutines RPNT31, STAT31, and NOST31 have been modified to account for this correction.

4. The subroutines IRFGT0, LOPT40, NOST31, PATH, and RPHRD2 have been modified to allow SCT tunneling approximations to be calculated when using the RODS algorithm with units fu30, fu31, and fu40, and the hooks.

Version 7.9 (March 1998)

1. A new keyword LOWFREQ has been added to the PATH section. The subroutines RLOWFR and REPFL have been added and the subroutine NORMOD has been modified to account for this new option.
2. To remove conflicts with AMSOL, the subroutines INVRT1 and OVERLP have been changed to INVRT and OVRLP, respectively.
3. The CALCS keyword has been removed from the PATH section because that option is not recommended. The subroutines DORODS, DEFPAT, INTAB, RPATH, RVAR6, and WVAR6 have been modified accordingly.

Version 7.9.1 (April 1998) (Bug fix)

1. Some extraneous print statements have been removed from the PATH subroutine.

Version 7.9.2 (June 1998) (Bug fixes)

1. A typo in the subroutine ROPTOP has been fixed. This typo caused an error in the OPTMIN and OPTTS keywords in the OPTIMIZATION section.
2. Subroutines ZOCPAR and ZOCPRN:
Modifications were made to correct errors in the frequency interpolation with the *ic1* option.
3. The bug introduced in item 8 of Version 7.8.1 was removed. Now the curvature at the saddle point is correctly calculated by eq. 98 of the GTST book chapter. Notice that even though the curvature at the saddle point itself is obtained by interpolation (see discussion of the INTMU keyword), when INTMU = 1 the imaginary frequency eigenvector at the saddle point is used to obtain the curvature at the points on the save grid or either side of the saddle point. Furthermore the code has been modified so that this calculation is carried out correctly when fu31 is used (which was not true before) as well as when the hooks or other electronic structure input file options are used. Subroutines STAT31, NOST31, RPHRD1, and SHTSAD were modified.

Version 8.0 (August 1998)

1. The mapped interpolation (MI) algorithm, which was added in version 7.8 for use with files fu30, fu31, and fu40, now also works with the hooks. The mapped interpolation option has been used to replace the two-point exponential extrapolation for the $V_a^G(s)$ and $\mu_{eff}(s)$ functions for reactions with a saddle point. For reactions without a saddle point the original two-point extrapolation is still used. Modifications were made to the subroutines ZOCUPD, BCALC, ELRPH,

MEPSRT, and PATH. New subroutines IVTMH, IVTMA, IVTMD, and ARMUEF1 have been added. The keywords EXFIRST and EXSECOND in the PATH section are no longer used with the electronic structure input files fu30, fu31, and fu40, and their meaning is changed when they are used with the hooks. Keywords UNIMOL and LBEXP were moved from file fu31 to the PATH section of file unit fu5.

2. The keyword LBEXP has been moved from file fu31 and is now an option for the keyword IVTSTMOPT in the PATH section of file fu5. In addition, the UNIMOL keyword has been removed from file fu31, and its options are now options for the keyword IVTSTMOPT in the PATH section of file fu5. These changes were made to use mapped interpolation with the hooks.
3. In subroutine REACT:
A new array WEW is added to store the frequencies of the wells for interpolation. Energies of the reactant well and product well are stored in the variables EWR and EWP, respectively. The geometries of the stationary points are stored in the corresponding XR arrays. These changes were made to use MI with the hooks.
4. In subroutine DEFPAT:
A new variable keyword INI has been added to indicate the interval of the gradient interpolation steps in the MI algorithm. A check has been added to the subroutine OPTION to make sure that $INI \leq INH$. These changes were made to use MI with the hooks.
5. The option *vrpe* has been added to the RPM keyword to allow the user to use the variational reaction path (VRP) algorithm based on the Euler steepest-descents (ESD) method to follow the reaction path. This algorithm is called VRP(ESD) or just VRP. The subroutines DEFPAT, DORODS, ENDRODS, INTAB, NORMOD, NOST31, PATH, RPHRD2, RRPMP, RVAR6, and WVAR6 have been modified.
6. The IC keyword has been changed to an option of the DL keyword and is renamed *ioc*. The ICOPT keyword has been renamed ICOPT and the options *eckart* and *noeckart* have been changed to *deckart* and *seckart*, respectively. The default for interpolating $V_{MEP}(s)$ is changed from *deckart* to *seckart*. The *zero* option has been removed from the ICOPT keyword.
7. A new option for correcting energies along the reaction path by the dual-level single-point energies (ISPE) method has been added. The new option *ispe* is available under the DL keyword. The file unit fu51 has been reserved for use with the new *ispe* option. Note that the ISPE option uses the mapped interpolation algorithm and so the formulas differ from an IOC calculation in which only the energy is corrected. Furthermore an IOC calculation involves corrections based on an optimization at a higher level. The IC keyword has been changed to an option of the DL keyword and is renamed *ioc*.
8. The option *coord* has been added to the PRPATH keyword in the PATH section to print the bond lengths and bond angles along the reaction path. A new subroutine PRCORD has been added and the subroutines INTAB, RPRMEP, NORMOD, FCRATE, FIOPEN, and RPHRD2 have been modified.
9. The keywords SPT and SPTOPT have been removed, and the associated has been commented out.

10. The NONADIB keyword has been renamed LCTCOR, the option previously called *quadratic* has been renamed *quadlin*, and the default has been changed from *no* to *quadlin*.
11. The LOWFREQ keyword in the PATH section of the fu5 input file has been renamed IVTST0FREQ.
12. The default value for the INTMU keyword of the PATH section of the fu5 input file was changed from 1 to 3.
13. The parameter NSDIM has been moved from percon.inc to the param*.inc files.
14. The RPM keyword option *euler* has been renamed *esd*; subroutines DEFPAT, INTAB, and RRPMP have been modified.
15. In subroutine MAIN:
Modifications were made to allow anharmonicity options to be used with TST-only calculations.
16. In function VPART:
The vibrational partition function is set to 10^{10} when using the WKB method with an imaginary frequency normal mode. This value was previously set to 1.
17. In subroutine NOROUT:
Modifications were made to print the ground-state vibrational energies for the saddle point to file unit fu6.
18. Modifications have been made to print 'VTST-IOC' and 'VTST-ISPE' to the output files instead of 'VTST/IC'.
19. In subroutines REACT and NOROUT:
Modifications were made to print 'Harmonic Frequencies' to the output files instead of 'Frequencies'.
20. The common block OUTPUT was renamed OUTCOM in all the subroutines contained in the givtst.f subprogram.
21. The subprograms SORT and UPCASE were renamed SORTG and UPCSE, respectively. In addition the function TORS was renamed PTORS.
22. The IBM PowerPC 604e option has been added to the set_machine.jc script and the necessary script files, *.ibm604, have been added to the script directory.
23. The test runs oh3fu31tr2, ch5fu51tr1, ch5fu50tr3, and cmtr3 have been added to the test suite.
24. The WRITEFU31 keyword has been added to the oh3tr10 test run.
25. The oh3tr1 test run has been modified to use the *vrpe* option of the RPM keyword.

1. A new geometry optimization option using the eigenvector following (EF) algorithm was added for the reactants, products, reactant well, product well, and saddle point. The FORTRAN source code is taken from Frank Jensen's program in the MOPAC5.07MN software package (this EF code is originally from Ref. 2 in Sect. 22) and stored in the file ef.f. The subroutines OHOOK, DEFOPT, ROPT, ROPTOP, and INTAB were modified, a new subroutine REFOPT was added, and the compilation scripts comp_src.* are modified to accomplish these changes.
2. A new option for calculating the reduced moment of inertia of an asymmetric rotator using Pitzer's curvilinear algorithm was added. The subroutines RTOR and RTOROPT were added to replace the RHINRT and RHINRTS subroutines in the interface.f file. Subroutines DEFSTAI, INTAB, INTABS, RSTAT, SETANH, ANHARM, CENTER, HRPART, RATE, RPHINT, and TSRATE were modified for this new option. Three new subroutines, HRSET, ASYMRMI and ASYMENT, were added. One can now estimate the effective vibrational frequency based on the rotational barrier height. A new testrun cwmctr3 is added to illustrate the curvilinear option. Also, the hindered rotor treatment has been generalized so that it no longer implicitly assumes that $P = 1$. The old list keyword HINROT was removed and is replaced by new list keywords TOR and TOROPT.
3. The subroutines ICFDIAG, ICSAVE, and CVCOOR were modified to carry out normal mode analysis using curvilinear internal coordinates (with or without redundancy) at the saddle point for the CURV2 and CURV3 options.
4. The PRMODE keyword was added to the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections for printing movie clips of the vibrational modes for the XMOL program. The subroutines NOROUT, INTABS, DEFSTA, and RSTAT were modified, and a new function DIGISTR was added.
5. The option DVMEP is added to file fu51 for use with the IVTST-ISPE option. By using this option, the point at the higher level doesn't need to be a save-grid point at the lower level. The subroutines READIC and ZSPMEP were modified.
6. The subroutine WRTHOK has been modified for printing out the vibrational frequencies and Hessians for calculations with the SOLID option.
7. In the previous versions of POLYRATE, the FREQ list keyword is used when a list of harmonic frequencies is read in as a restart option when the STATUS keyword has the value 6, and the NOFREQ option is used to switch off the normal mode analysis. In this version, the functions of the old keyword FREQ in the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections have been separated, and they are now associated with the keywords VIB and FREQ in order to clarify these two distinct functions. The VIB list keyword is used to indicate a list of harmonic frequencies of the corresponding species when the STATUS keyword has the value

of 6. The `FREQ` keyword is used as a switch keyword to determine whether the normal mode analysis is carried out in the calculation.

8. The subroutines `IVTMD`, `IVTMH`, and `RPHINT` were modified to support the mapped interpolation `VTST` algorithm for cases without information about either the product or a product well. The method used in such calls is fully explained in Section 6.M. of this manual.
9. The keyword `FIXMUEF` has been added to the `IVTSTMOPT` list keyword in the `PATH` section. This keyword is used to apply a global interpolation to estimate the effective reduced mass (μ_{eff}) of the `SCT` algorithm at the saddle point when the `IVTST-M` algorithm is used with hooks. The default option is to use local Lagrangian interpolation for the effective reduced mass at the saddle point with either one point or three points (see `INTMU` keyword option) on each side of the saddle point. The global interpolation can be used to obtain a smoother value of μ_{eff} at the saddle point; it should be used if and only if the points close to the saddle point suffer from numerical instability.
10. The keywords `RMITS`, `RMIRP`, and `HRRMI` were renamed to `MITS`, `MIRP`, and `HRMI` for input of the moment of inertia of the internal rotation mode in file `fu50` instead of the reduced moment of inertia as the previous versions. The subroutine `ROPT50` has been modified.
11. A check has been implemented in the subroutine `REMOVEC` to ensure that the total number of internal coordinates is greater or equal to the total number of vibrational modes.
12. The subroutine `WRTHOK` has been modified in order to print out correct energies for the reactants, products, wells, and saddle point when using the restart option controlled by the `STATUS` keyword.
13. The subroutine `SSAVE` has been modified to ensure that the array `BKAP` will contain the correct value of the curvature along the reaction path.
14. The `IVTST0FREQ` keyword option has been extended the case of unimolecular reactants and/or products. The subroutine `REPFL` has been modified and a new subroutine `rprfl` has been added to `polytr.f`. The test run `ch5j1tr7` is used as an example for this keyword option.
15. The compilation and linking scripts for IBM-SP with PowerPC 604e processor is upgraded to O3 with 64-bit computation.

Version 8.1.1 (July 1999) (Bug fixes)

1. The `BLAS` subroutines in the `ef.f` file were moved to the file `fromblas.f` in order to keep all the routines from the mathematical libraries in the same file.

2. The default values of the SR and SP options in the IVTST0FREQ keyword have been set to -1.0 bohr and $+1.0$ bohr.
3. The keyword option PROJECT in the *START section has been corrected. In previous versions, the translational and rotational modes are always projected. We recommend to project out the translational and rotational modes in order to obtain better vibrational frequencies; however, this is not necessary. The subroutine CENTER has been modified.

Version 8.1.2 (August 1999) (Bug fix)

1. The *.jc scripts in the testrun directory were updated to work with the current version of POLYRATE.

Version 8.2 (August 1999)

1. New keyword option BATH and its options FRICTION, BATHTEMP, DIFFUSION, and CUPLCNST are added to the PATH section to perform nonequilibrium solvation calculations. There are two limitations for using this new option, 1) the system must be molecule, not solid state (because the redundant internal coordinate system is used) and 2) the equilibrium free energy of solvation is included in the potential function. In this version, there are no test cases for carrying out nonequilibrium solvation calculations using POLYRATE directly, however, new test cases are added to AMSOLRATE and GAMESOLRATE where the free energy of solvation is approximated using a continuum solvation model. Three new subroutines are added to POLYRATE: RBATH, RDIFFU, and EFFBATH. The NEWT, SADENG, EHOOK, GHOOK, HHOOK, CVCOOR, IVTMH, DOSAFR, INITZE, NOROUT, PATH, PROJCT, VPART, WRTHOK, and ZOC3P subroutines are modified. Since BATHTEMP is a variable keyword, the nonequilibrium solvation calculation can only be carried out at one temperature.

Version 8.2.1 (November 1999) (Bug fix)

1. The function subprogram UPCASE has been renamed UPCSE. This should have been done in version 8.0 (see item 21 of the version 8.0 revision history).

Version 8.4 (December 1999)

1. The subroutines RATE and TSRATE have been modified to print out the internal free energy of reactant.
2. The nh3tr2 testrun has been created.

Version 8.4.1 (March 2000) (Bug fixes)

1. The file /testo/cwmctr3.fu15 has been replaced with the correct version for this testrun.
2. FORMAT statements were changed to prevent compiler warnings.

Version 8.4.2 (October 2000) (Bug fix)

1. The calculation of the partition function was incorrect when calculated off the saddle point when using curvilinear internal coordinates. The bug was due to the PROJCT subroutine being commented out in versions 8.1-8.4.1. This affects all runs using the options CURV2 and CURV3 in versions 8.1-8.4.1. CURV1 and rectilinear-vibration runs were unaffected by this bug.

Version 8.5 (October 2000)

1. The LCG4 option has been added. The implementation of the LCG4 method in POLYRATE involved the modification of the following subroutines:
 - a. LCXINT: function that calculates the imaginary action integral at every quadrature point of the adiabatic region.
 - b. LCGTH (old LCG3TH): this subroutine calculates the imaginary action integral between the outer turning points of the vibrational adiabatic potential. Part of the code that was previously in this subroutine has been incorporated into two new subroutines called LCNAD and LCVEF.
 - c. LCNAD: returns the value of the imaginary action integral at every quadrature point of the nonadiabatic region.
 - d. LCVEF: function that calculates the effective potential of the nonadiabatic region.

The output of units fu41–fu44 was modified in the following ways (see page 250 of the POLYRATE–version 8.4.1 manual) and two new units (fu48 and fu49) were added:

- a. poly.fu41: contains the theta integral and the weighting factor of each point on each tunneling path.
- b. poly.fu42: contains the length of the tunneling path, the classical energy, the potential correction terms (zero-point energy and anharmonic), the interpolated correction if any, and the effective potential for each point in the nonadiabatic region of the tunneling path.
- c. poly.fu43: contains the theta integrals and the boundaries of the nonadiabatic region for each tunneling path.
- d. poly.fu44: contains the length of the tunneling path, the maximum of the effective potential, and its location on the path for each tunneling energy.
- e. poly.fu45-47: same as before.
- f. The unit poly.fu48 is used to restart a LCG calculation. It contains the grid number of a given tunneling energy, the grid number of a point along the

linear path and the value of the calculated potential in the nonadiabatic region (the expensive part of the LCG calculations).

- g. The unit poly.fu49 stores the information that can be read directly from unit poly.fu48 afterwards.
2. We added an option to scale the frequencies for CVT calculations with harmonic vibrations. This implementation involved changes in the main program, and the subroutines REACT, SADENG, ENROUT, SENOUT, NOROUT, RATE, TSRATE, TABLE, TSOUT, VPART, WRTWEL, and several subroutines in the interface.f block as well as changes in common.inc.
3. We added an option to compute energy levels by WKB-on-the-true-potential for excited-state calculations. This implementation involved changes in subroutines RATE, VPART, STAUUV, VTMUSN, and EVIB.
4. We added 6 new test runs, ohcltr1-4, ch5jltr9, and ch5jltr10.

Version 8.5.1 (November 2000) (Bug fix)

1. An error was made in the hindered rotor subroutine, HRSET, in version 8.5. This bug only affects version 8.5.

Version 8.6 (February 2001)

1. The curvilinear internal coordinate bound-mod analysis for reactants and products is added.
2. The scaling force constant keyword FCSCALE is added into PATH section.
3. The keyword SCLPT and CARTRP are added. SCLPT is used to specify the effect of force constant scaling. CARTRP is used to declare explicitly that the normal mode analysis for reactants and products is carried out in Cartesian coordinates.
4. The switch subkeyword RTPJAC is added to the list keyword LCTOPT to specify the MEP and large curvature representative tunneling termini in mass-scaled Jacobian coordinates are printed in fu24.

Version 8.6.1 (August 2001, Bug fixes)

1. The behavior of the keyword FREQSCALE is changed:

Previously the keyword FREQSCALE was used to scale all harmonic vibrational frequencies uniformly but only for partition function calculations. However, the scaled frequencies were not used in calculation of zero point energies or transmission coefficients. The usage of FREQSCALE has been corrected to remove

this error. If FREQSCALE is used, the scaling will be applied to all stationary point frequencies and generalized normal mode frequencies for generalized transition states along the reaction path. Furthermore, the vibrationally adiabatic potential energy curve V_a^G will be obtained by adding the scaled vibrational ZPE to V_{MEP} , and all tunneling calculations will be based on this new V_a^G .

The above changes involving FREQSCALE are implemented in subroutine NORMOD (in energetics.f) by scaling all frequencies right after they are obtained from force constant matrix diagonalizing routines FDIAG, ICFDIAG, or ICFDRP.

2. The bug is fixed for curvature calculations when fu31 is supplied as electronic structure input file, and RODS keyword is used.

When RODS is specified, the reaction path curvature vector calculation should be based on the vector normal to the reoriented dividing surface optimized by RODS algorithm. It involves a derivative calculation of this vector normal to the dividing surface with respect to the reaction coordinate s . Before the numerical derivative routine is called, the above vectors should be converted to mass-scaled and should be normalized. However, the code in old version did not handle this correctly. We fixed this bug in the current version in the subroutine NOST31 (poly31.f).

Version 8.7 (September, 2001)

1. The normal mode analysis of reactants and products is set to Cartesian coordinates as the default, i.e., CARTRP is set as default in PATH section. One can specify that the generalized normal mode analysis for reactants and products is applied in curvilinear coordinates by turning CARTRP off (NOCARTRP). Note that NOCARTRP must be used when the force constant scaling keyword FCSCALE is specified, and if incompatible options are chosen, a message is printed and the program stops.
2. The default for EZUNIT keyword in the ENERGETICS section had not been set in previous versions. This bug is fixed, and the default for EZUNIT is set to be *au* in the subroutine DEFENG.
3. The default setting of INITGEO keyword for reactants, products, and saddle points is moved from the subroutine DEFOPT to the appropriate subroutine DEFSTA because the keyword INITGEO is defined under REACT1, REACT, PROD1, PROD2, and START sections instead of OPTIMIZATION section.
4. In subroutine INTAB, the writing of FPRINT input option in SECOND section is taken from the ghook "if" statement, so that status of the FPRINT switch will always be printed out whether or not HESSCAL is set to *ghook* or *hhook*.

Version 8.7.1 (January 2002, Bug fix)

1. Array declarations in rtpjac.f and intbsv3.f are modified to be compatible with the '-static' compiling option in compiling scripts.

Version 8.7.2 (February 2002, Bug fixes)

1. A bug in subroutine HRSET was fixed that affected hindered rotor calculations. This bug only affects version later than 8.5.
2. A bug was fixed in subroutine REACT. The geometries for each species should be copied to a local array no matter the initial geometries were obtained from fu5 or hooks. See comments in the source code for details.
3. A bug was fixed in subroutine DEFSTA. The initialization of array INITG(I) should be done from I=1 to 8 not only from I=1 to 5. See comments in the source code for details.

Version 8.7.3 (May 2002, Bug fixes)

1. The initialization of logical variable LUPD in subroutine EFMAIN that controls the use of an updated Hessian for an optimization was not done. This logical variable is now initialized to .TRUE.
2. The initialization of the variable IPRNT that controls some printing options in subroutine EFMAIN was not done. This variable is now passed through a common block.
3. A bug in subroutine KAPVA was causing the wrong header to be printed for LCG3 or LCG4 calculations when using the quantized-reactant-state tunneling method was fixed. This has now been fixed.
4. A bug was fixed in subroutine HRSET that calculates information needed for evaluating the hindered rotor partition function. A wrong index for a vibrational frequency mode had been used.
5. A compilation error with Linux has been fixed. This bug fix involved modifications in the TABLE, FINOUT, and KAPVA subroutines.

Version 8.9 (May 2002)

1. The ILCG option has been added. This is done though two new keywords, ILCT and NILCTPT, that have been added to the list keyword LCTOPT. The implementation of the ILCG method in POLYRATE involved modifications of the following subroutines:
 - a. RLCT: two new variables, ILCGIT and NSPLIC, have been added. These two variables are also included in common.inc.
 - b. INTAB: the two new keywords, ILCT and NILCTPT, have been included and are now tabulated.
 - c. LCG34 and LCGTH: the modifications have been made to incorporate the interpolated LCG algorithm.
 - d. LCGIT: this is a new subroutine that chooses the points at which the ILCG interpolation is made.
 - e. LCVEF: the modifications allow performing the spline-under-tension interpolation.

- f. SPL: the dimensionalities of arrays IPAS and SAVARR have been increased.
2. An option for determining the MEP limits based on energetics criteria has been added. This is done through the new keyword SPECSTOP. The implementation of this new feature required addition of a new subroutine, RSSTOP, and modifications to the following subroutines: DEFPAT, ELRPH, INTAB, OPTION, PATH, and RPATH.
3. The generalized free energy of activation can now be calculated in subroutines RATE and TSRATE either through direct use of the vibrational partition functions or through the use of the logarithms of the vibrational partition functions. Using logarithms enables the calculation of the free energy of activation when the vibrational partition functions (or their product) are susceptible to underflow. The procedure used for calculating generalized free energy of activation is determined by the GTLOG switch in the RATE section. The implementation of this feature required modifications to the following subroutines: DEFRAT, INTAB, RATE, RRATE, SETLGC, and TSRATE.
4. The tunneling contributions obtained using the quantized tunneling energy method are now calculated, in the BOLTZ subroutine, by using an exponential of an energy difference rather than a product of exponentials. In this way, the results are precisely the same as before, and the calculation is more stable.
5. The STATES subkeyword is introduced under the QRST keyword to limit the number of quantized reactant states that will be included in the tunneling calculation. The implementation of this feature required modifications to the following subroutines: DEFRAT, QRSENE, RQRST, SETLGS, and WKBENE.
6. The format of the output files is changed so that the results obtained with ZCT (MEPSAG) method are labeled as ZCT instead of MEPSAG, and the results obtained with SCT (CD-SCSAG) method are labeled as SCT instead of CD-SCSAG. The modified subroutines are BOLTZ, GIVTST, KAPA, KAPPAS, KAPVA, and TABLE.
7. The set_machine.jc script has been upgraded to include options for SGI Origin R12000 processors, for Sun UltraSparc workstations, and for Compaq AlphaStations.
8. New scripts have been added for SGI Origin R12000 processors, for Sun UltraSparc workstations, and for Compaq AlphaStations.
9. The script comp_src.linux was modified to allow compilation with the fort77 compiler in Mandrake Linux.
10. Two new timing routines have been added. One of them, dattim.compaq, is used for timings on Compaq workstations, and the other, dattim.f90, uses the DATE_AND_TIME routine that is the standard timing routine in Fortran 90.
11. Three new test runs, ch5j1tr11, clhbrtr3, and clhbrtr4, have been added, and two test runs, ch5j2tr and oh3tr8, have been slightly modified to include the SPECSTOP keyword. The running, checking, and cleaning scripts have been updated.

12. A few old variables that were not used anymore have been removed from the code.

Version 8.9.1 (May 2002, Bug fixes)

1. The default setting of STATES subkeyword in QRST keyword is moved from the subroutine DEFRAT to the appropriate subroutine DEFTUN because the keyword QRST is defined under the TUNNEL section instead of the RATE section.
2. The interpolation for determining the tunneling probability at certain energies was done wrongly when using quantized tunneling energy method with a low frequency mode for the reaction coordinate. This bug was fixed in subroutine BOLTZ.
3. A bug was fixed in subroutine QRSENE that was not allowing using quantized tunneling energy method with a given number of quantized states.

Version 9.0 (May 2002)

1. The keyword WRITE62 has been added to the GENERAL section of the unit fu5 input file to create an fu62 file that contains a subset of the stationary point information presented in fu61. The implementation of this feature required modifications to the following subroutines: DEFGEN, INTAB, RGENER, FIOPEN, FCRATE, and WRTHOK.
2. The STQVIB subkeyword is introduced under the QRST keyword to limit the number of quantized reactant vibrational states that will be included in the reactant vibrational partition function. The implementation of this feature required modifications to the following subroutines: DEFTUN, QRSENE, RQRST, KAPVA, and WKBENE.
3. The format of the long output file, fu6, has been slightly modified when using quantized tunneling energy method. Similarly, the format of the output file fu15 has been slightly modified for IVTST calculations.
4. Test run nh3tr2 has been slightly modified to include the GTLOG keyword.
5. A few common blocks have been renamed and are now ended in either COM or CM.
6. A few old variables that were not used anymore have been removed from the code.

Version 9.1 (July 2002)

1. The long output file, fu6, has been modified so it contains the rate constants with non-tunneling contributions (i.e., contributions from energies above the top of the effective barrier). The implementation of this feature required modifications to the following subroutines: BOLTZ, FINOUT, and KAPVA.
2. The default of the CARTRP switch keyword has been set to on. The option of changing the default based on the values of other keywords has been removed. The implementation of this feature required modifications to the following subroutines: DEFPAT, INTAB, and SETLGS.

Version 9.1.1 (not released)

1. One bug affecting the evaluation of the correction potential, V_{corr} , when tunneling occurs into final vibrational excited states was corrected.
2. The restart option of the LCT method, which under some conditions, gave wrong results for tunneling into vibrationally excited states was corrected.

Version 9.2 (July 2003)

1. Modification of subroutine LCG34.
2. The old LCGTH subroutine is now subroutine LCGNAD. It evaluates the LCT3 and LCT4 nonadiabatic regions and stores all the needed information to evaluate the tunneling amplitude.
3. The new subroutine LCGTHE evaluates the theta integral.
4. Old subroutine LCGNAD is now subroutine LCG4V. It searches for a LCT4 nonadiabatic region within a LCG3 adiabatic region.
5. Two common blocks GRDDX and GRDDX2 were added to store the information from the LCGNAD subroutine.
6. Several subroutines called by SPL2D, which incorporate the 2D spline under tension, were added. They are based in the code developed by R. Renka.
7. The rectangular grid needed by the 2D spline under tension is built in LCGRD and printed in unit fu42.
8. The LCGTR subroutine was added.
9. The LCVEF subroutine was slightly modified.
10. Keywords **ILCT1D**, **ILCT2D** and **LCTGRID** were added and the two keywords ILCT and NILCTPT were removed.
11. Subroutine RLCTGRD was created in order to activate the new keywords.
12. Two new test runs, clhbrtr5 and ch5j1tr12, were added.
13. The Linux version of POLYRATE must be compiled with Portland Fortran 77.

Version 9.3 (November 2003)

1. Version 9.3 runs on Windows 95/98/Me/2000/XP.
2. Version 9.3 can be compiled with the efc compiler and run on the SGI Altix.

Version 9.3.1 (June 2005, Bug fixes)

1. Previous versions had errors in the calculation of the reduced moment of inertia for a torsion using the curvilinear scheme of Pitzer and Gwinn. Version 9.3.1 corrects the errors in the alpha matrix, and uses all right-handed coordinate systems. The CO and CW torsion schemes now work correctly.
2. Version 9.3.1 no longer prints out the table of torsion information before it is calculated, and the value labeled "Total Moment of inertia" is now correct.
3. The ISPE and FREQSCALE options are now compatible with fu31 files. The IVTST0FREQ option remains a known incompatibility.

Version 9.3.2 (not released)

1. Program will now compile and work properly with the gnu and ifort compilers. Since the gnu compiler is available in Linux and for most other operating systems, this greatly increases the portability.

Version 9.4 (May 2006)

1. The 'configure' script is added to automatically detect compilers, and a Makefile is made for easy compilation of Version 9.4.
2. Version 9.4 allows parallel processing using MPI. The code has been rearranged to accommodate this, where the file poly_mpi is used for multiple processors, and the file dummy_mpi is required for serial jobs. This is done in the Makefile.

Version 9.4.1 (June 2006)

1. The 'configure' script has been modified to only print the parallel options if an MPI compiler is found.

23.B. ASCII manual revision history

From version 1.6 through version 5.1, the manual was distributed as an ASCII text file.

23.C. PostScript and PDF manuals

The manuals for versions 6.0 to 7.2 were produced with Microsoft Word 5.0 or 5.1a for the Macintosh and distributed as a PostScript file. The manuals for versions 7.3 to 8.4.1 were produced with Microsoft Word 7.0 for Windows and distributed in PostScript and

Portable Document Format (PDF) formats. Beginning with version 8.5 the manual is distributed only in Portable Document Format (PDF) format.

23.D. Incompatibilities of POLYRATE–version 9.4.1 with previous versions

1. In version 3.9, the default SCT tunneling method was changed from SCSAG to CD-SCSAG. Thus, $LGS(9) \geq 1$ in all previous versions [and $LGS(9) \leq -2$ in versions 3.0-3.07] requested the SCSAG method, but from version 3.9 to 5.1.1, these options request CD-SCSAG. In versions 6.0 and later, where this kind of choice is controlled by keywords, the CD-SCSAG method remains the default. In fact the original SCSAG method was removed in version 5.0.
2. In the FORTRAN unit fu5 input data file, the input order for the normal modes and generalized normal modes in records 10A, 15, and 15B was changed in passing from version 4.5.1 to version 5.0. In versions 5.0 and later, the modes should be ordered from the highest to the lowest frequency, which is consistent with the universal convention in the spectroscopic literature.
3. The Quack-Troe method for interpolating energies along the reaction path was removed for versions 6.0 and later.
4. No FORTRAN unit fu5 input data files from versions earlier than 6.0 will work in POLYRATE version 6.0 or later.
5. The input keyword CONVERGENCE in the OPTIMIZATION section was removed as of version 6.6.
6. In version 7.0 and later, variables *quad* and *page* for the keyword CURVE have been changed to *dgrad* and *dhess*, respectively. Furthermore, the PAGE keyword for activating the Page-McIver integrator has been changed to PAGEM.
7. In versions 7.0 and later, the saddle point geometry is no longer read from both the unit fu5 file and the unit fu30 file; it is only read from the unit fu5 file. Additionally, the DELTAH keyword has been removed from the unit fu29 input file.
8. In versions 7.0 and later, for the unit fu30 file, the meaning of $LOPT(3) > 1$ and $LOPT(4) > 2$ have been changed.
9. In versions 7.0 and later, the frequencies along the MEP are ordered in canonical order by default, whereas previous versions ordered the frequencies according to overlap calculations.
10. In versions 7.1 and later, when using the unit fu40 file, the default unit for frequencies is cm^{-1} , whereas in version 7.0, these frequencies were in atomic units. Note: For the unit fu30 file, frequencies are still in atomic units, but the unit fu40 file has the full functionality of the unit fu30 file (except for $LOPT(7)$ and $LOPT(8)$ not equal to zero) and is more convenient, so we recommend using the unit fu40 file instead of the unit fu30 file.
11. In versions 6.0-7.1, one could carry out calculations on a reaction without a saddle point simply by using the NOOPT keyword. Starting with version 7.1.1 one must

use NOOPT, NOSADDLE, and FIRSTSTEP = *gradi*. The reason we changed this is so that the combination of NOOPT and SADDLE could be allowed.

12. In version 7.2, the default values for the Interpolated Correction option (ICOPT) are set to ICL and ECKART, whereas previous versions used ICA and NOECKART. See also item 32 where the default is changed again.
13. In version 7.2, the allowed values for the keyword POTENTIAL were changed; the option *analytic* is renamed *hooks*.
14. In version 7.2, the keyword POLYRATE in the SECOND section was changed to HESSCAL = *ghook*.
15. Starting with version 7.2, the keywords POLYRATE, MOPAC, GAUSSIAN, BFGS, and TSBFGS in the OPTIMIZATION section of the unit fu5 input file have been replaced by two keywords OPTMIN and OPTTS. GAUSSIAN and MOPAC are now selected using OPTMIN = *ohook* and OPTTS = *ohook*. The POLYRATE and BFGS combination of keywords is now selected by setting OPTMIN = *bfgs*. POLYRATE and NOBFGS are selected by setting OPTMIN = *nr*.
16. Starting with version 7.2, the values of the SIGMAF and SIGMAR keywords are no longer read from a restart file. The correct values must be read from the unit fu5 input file; otherwise a default value of 1.0 will be used.
17. Starting with version 7.2, the option *reactants* of the EZERO keyword of the ENERGETICS section of the unit fu5 input file was removed. POLYRATE will automatically determine the zero of energy (denoted by the option *calculate*) unless the option *read* of the EZERO keyword is used.
18. In version 7.3, the meanings of the values of the variable IPOT have been modified as follows:

value	old meaning	new meaning
0	<i>hooks</i>	<i>hooks</i>
1	<i>mopac</i>	<i>unit30/unit40</i>
2	<i>aces</i>	<i>aces</i>
3	<i>gamess</i>	<i>unit29</i>
19. In subroutine ZOCPAR:
The default value of the keyword RPL was calculated from the lower-level surface in version 7.2; from version 7.3 on, the default is set to the value calculated from the higher-level calculations.
20. In version 7.3, units fu31-36 were renamed fu71-76.
21. In version 7.3, the switch keywords NONLINEAR and LINEAR and the variable keyword PHASE were incorporated into the single variable keyword SPECIES. The default has been set to *nonlinrp*.
22. In version 7.4, the keywords ICA, ICL, and ICR are replaced by options for the new keyword option FREQ in the ICOPT list.
23. In version 7.4, the default of the FREQMAT keyword on unit fu50 has been modified for all cases. From now on, the frequencies of a unimolecular reaction or reactions with wells are matched in descending order by default.

24. In version 7.4, the keywords PRSAVERP and PRSAVEMODE were changed from switch keywords to variable keywords with default values of 9999.
25. The two extra gradients given in the unit fu30 files for computing the curvature were placed before the Hessian matrix in previous versions of POLYRATE, but have been moved after the Hessian matrix in version 7.4.
26. In version 7.8, the INM and SSAVE keywords have been removed. The save grid information is now saved every INHth step. Therefore, the Hessian grid and the save grid are now the same for all calculations.
27. In version 7.8, the EULER, ES1, and PAGEM keywords have been combined into options for the new RPM keyword.
28. In version 7.8, the default for the INH keyword has been changed from 1 in previous versions to 9.
29. In version 7.9, the CALCS keyword has been removed from the PATH section.
30. In version 8.0, the mapped interpolation (MI) algorithm, which was added in version 7.8 for use with fu30, fu31, and fu40, now also works with the hooks. The MI option has been used to replace the two-point exponential extrapolation for the $V_a^G(s)$ and $\mu_{eff}(s)$ functions for reactions with a saddle point. For reactions without a saddle point the original two-point extrapolation is still used. The keywords EXFIRST and EXSECOND in the PATH section are no longer used with the electronic structure input files fu30, fu31, and fu40. See also item 31.
31. In version 8.0, the keyword LBEXP has been moved from file fu31 and is now an option for the keyword IVTSTMOPT in the PATH section of file fu5. In addition, the UNIMOL keyword has been removed from file fu31, and its options are now options for the keyword IVTSTMOPT in the PATH section of file fu5. These changes were made to use MI with the hooks.
32. In version 8.0, the IC keyword has been changed to an option of the DL keyword and is renamed *ioc*. The ICOPT keyword has been renamed ICOPT and the options *eckart* and *noeckart* have been changed to *deckart* and *seckart*, respectively. The default for interpolating $V_{MEP}(s)$ is changed from *deckart* to *seckart*. The *zero* option has been removed from the ICOPT keyword.
33. In version 8.0, the keywords SPT and SPTOPT have been removed.
34. In version 8.0, the RPM keyword option *euler* has been renamed *esd*.
35. In version 8.0, the NONADIB keyword has been renamed LCTCOR, the *quadratic* option has been renamed *quadlin*, and the default has been changed from *no* to *quadlin*.
36. In version 8.0, the LOWFREQ keyword in the PATH section of the fu5 input file has been renamed IVTSTOFREQ.

37. In version 8.0, the default value for the INTMU keyword of the PATH section of the fu5 input file has been changed from 1 to 3.
38. In version 8.0, the vibrational partition function has been set to 10^{10} when using the WKB method with an imaginary frequency normal mode. This value was previously set to 1.
39. In version 8.1, the keyword FREQ has been separated into two keywords, VIB and FREQ. The keyword VIB is used to specify a list of harmonic vibrational frequencies when STATUS is 6 for the reactants, products, wells, and saddle point. The keyword FREQ is used as a switch keyword to indicate whether normal mode analysis is to be carried out.
40. In version 8.1, the old hindered-rotor list keyword HINROT is removed; it is replaced by new list keywords TOR and TOROPT.
41. In version 8.1, the keywords RMITS, RMIRP, and HRRMI were renamed to MITS, MIRP, and HRMI to input the moment of inertia of the internal rotation mode instead of the reduced moment of inertia in a fu50 file. Notice that the difference between the reduced moment of inertia and the moment of inertia is that the reduced moment of inertia is divided by the square root of the symmetry number.
42. In version 8.1, the input format for the keyword IVTST0FREQ has been changed to include new options INDEX, SR, and SP. And the algorithm used for interpolating the vibrational mode also differs from the previous version. See section 10.M for a detailed explanation for the algorithm.
44. In version 8.1.1, the NOPROJECT keyword option in the START section has been fixed in order to work in the way that is described in the manual. In previous versions, the translational and rotational modes of the saddle point were always projected.

23.E. Program distribution history

23.E.1. Program libraries

Version number	Date sent to library	Library or libraries sent to	Program number
1.1	1987	CPC	ABBD
2.5	Dec. 1990	QCPE	601-version 2.5
4.0	Mar. 1992	QCPE	601-version 4.0
4.0.1	Mar. 1992	CPC	ACJC
4.0.1	Apr. 1992	QCPE	601-version 4.0.1
5.0	Apr. 1993	QCPE	601-version 5.0
5.0.1	May 1993	QCPE	601-version 5.0.1
6.5	Apr. 1995	QCPE	601-version 6.5
6.5	Apr. 1995	CPC	ADBN
6.6	Nov. 1995	QCPE	601-version 6.6
7.3.1	July 1997	QCPE	601-version 7.3.1

23.E.2. Anonymous ftp

Version number	Ftp site	Date available
6.5	SDSC	Mar. 1995
6.6	SDSC	Nov. 1995
7.0	SDSC	September 1996

23.E.3. World Wide Web

Version number	Web sites	Date available
7.0	U of M and SDSC	September 1996
7.1	U of M	February 1997
7.1.1	U of M	March 1997
7.2	U of M	April 1997
7.3	U of M	July 1997
7.3.1	U of M	July 1997
7.4	U of M	August 1997
7.4.1	U of M	September 1997
7.4.2	U of M	October 1997
7.4.3	U of M	October 1997
7.8	U of M	December 1997
7.8.1	U of M	February 1998
7.9	U of M	March 1998
7.9.1	U of M	April 1998
8.0	U of M	July 1998
8.1	U of M	July 1999
8.1.1	U of M	July 1999
8.1.2	U of M	August 1999
8.2	U of M	August 1999
8.2.1	U of M	November 1999
8.4	U of M	December 1999
8.4.1	U of M	March 2000
8.5	U of M	October 2000
8.5.1	U of M	November 2000
8.6	U of M	February 2001
8.6.1	U of M	August 2001
8.7	U of M	September 2001
8.7.1	U of M	January 2002
8.7.2	U of M	February 2002
8.7.3	U of M	May 2002
8.9	U of M	May 2002
8.9.1	U of M	May 2002
9.0	U of M	May 2002
9.1	U of M	July 2002
9.3	U of M	November 2003
9.3.1	U of M	June 2005
9.4	U of M	May 2006
9.4.1	U of M	June 2006

The URL for the U of M website is <http://comp.chem.umn.edu/polyrate>

23.E.4. Other

A few other versions have been distributed on a limited basis.

23.E.5. Current

Current availability is delineated in Section 3 on availability and licensing.

24. ACKNOWLEDGMENTS

The authors are grateful to Da-hong Lu, Thanh N. Truong, Sachchida N. Rai, Gene C. Hancock, Jack Lauderdale, Tomi Joseph, Michael Zhen Gu, and Steven Clayton for contributions to earlier revisions of POLYRATE. The development of all versions of POLYRATE has been supported in part by grants from the U. S. Department of Energy, Office of Basic Energy Sciences (principal investigator: Donald G. Truhlar, University of Minnesota). The development of POLYRATE has also been supported in part by a grant from the National Science Foundation, Department of Advanced Scientific Computing (principal investigator: Rozeanne Steckler, San Diego Supercomputer Center).

25. INDEX

0IVTST KEYWORD		BOLTZ SUBPROGRAM	393
DEFINITION	250	BOTHK KEYWORD	
ABSORD SUBPROGRAM	391	DEFINITION	200
ACALC SUBPROGRAM	391	BRENT SUBPROGRAM	394
ACESET SUBPROGRAM	391	BRNULI SUBPROGRAM	394
AITKEN SUBPROGRAM	391	BTEN1 SUBPROGRAM	393
AITKF2 SUBPROGRAM	391	BTEN2 SUBPROGRAM	393
AITKNF SUBPROGRAM	391	BTEN5 SUBPROGRAM	393
ALFCT SUBPROGRAM	391	BTEN5X SUBPROGRAM	393
ALLEXCIT KEYWORD		BTENS SUBPROGRAM	393
DEFINITION	190	BTENTR SUBPROGRAM	393
TESTRUN EXAMPLE	292, 297, 330	BTORSN SUBPROGRAM	392
AMSOLRATE	15	CALCS SUBPROGRAM	394
ANALYSIS KEYWORD		CALCUB SUBPROGRAM	394
DEFINITION	200	Canonical order	44
ANCOEF SUBPROGRAM	391	Canonical Variational Theory	12
ANGL SUBPROGRAM	391	CARTRP KEYWORD	
ANGLCS SUBPROGRAM	391	DEFINITION	169
ANGLV SUBPROGRAM	391	CASE SUBPROGRAM	394
ANHARM SUBPROGRAM	67, 392	CD-SCSAG	55
Anharmonicity	45, 46, 98, 215	CENTER SUBPROGRAM	67, 394
Anharmonicity keywords	147	CENTR1 SUBPROGRAM	394
ANHDER SUBPROGRAM	392	CENTRAL SUBPROGRAM	394
ANTLR KEYWORD		CFLOAT SUBPROGRAM	394
DEFINITION	148	ch4ctr1 test run	381
ARMUEF SUBPROGRAM	392	ch4ctr2 test run	382
ARMUEF1 SUBPROGRAM	392	ch4ohtr1 test run	379
ARRSM SUBPROGRAM	392	ch4ohtr2 test run	380
ARRSV SUBPROGRAM	392	ch4otr1 test run	383
ASYMALP SUBPROGRAM	392	ch5fu29tr1 test run	305
ASYMCAL SUBPROGRAM	392	ch5fu29tr2 test run	306
ASYMENT SUBPROGRAM	392	ch5fu30tr1 test run	307
ASYMNZD SUBPROGRAM	392	ch5fu30tr2 test run	308
ASYMRMI SUBPROGRAM	392	ch5fu30tr3 test run	309
ATOMS KEYWORD	77, 124	ch5fu30tr4 test run	310
DEFINITION	127	ch5fu30tr5 test run	311
BANGLE1 SUBPROGRAM	392	ch5fu30tr6 test run	312
BARRI KEYWORD		ch5fu31tr1 test run	313
DEFINITION	211	ch5fu40tr1 test run	314
BATH KEYWORD		ch5fu40tr2 test run	315
DEFINITION	169	ch5fu40tr3 test run	316
BCALC SUBPROGRAM	392	ch5fu40tr4 test run	317
BCALCO SUBPROGRAM	392	ch5fu40tr5 test run	318
BCALCP SUBPROGRAM	392	ch5fu40tr6 test run	320
BIMAT SUBPROGRAM	393	ch5fu50tr1 test run	321
BMAT SUBPROGRAM	393	ch5fu50tr2 test run	322
BMAT1 SUBPROGRAM	393	ch5fu50tr3 test run	323
BMAT2 SUBPROGRAM	393	ch5fu51tr1 test run	324
BMAT5 SUBPROGRAM	393	ch5icfu30tr1 test run	325
BMAT5X SUBPROGRAM	393	ch5icfu30tr2 test run	326

ch5icfu31tr1 test run	327	Rectilinear	49
ch5icfu40tr1 test run	328	CORTRM SUBPROGRAM	395
ch5icfu40tr2 test run	329	COSINE SUBPROGRAM	395
ch5j1tr1 test run	285	CRVVEC SUBPROGRAM	395
ch5j1tr10 test run	297	CUBIC SUBPROGRAM	395
ch5j1tr11 test run	298	CUBIC2 SUBPROGRAM	395
ch5j1tr2 test run	287	CUBST SUBPROGRAM	395
ch5j1tr3 test run	288	CURV KEYWORD	
ch5j1tr4 test run	289	DEFINITION	170
ch5j1tr5 test run	290	DGRAD OPTION	104
ch5j1tr6 test run	292	TESTRUN EXAMPLE	285, 290, 295, 330,
ch5j1tr7 test run	293	332, 340, 341	
ch5j1tr8 test run	294	DHESS OPTION	104
ch5j1tr9 test run	295	TESTRUN EXAMPLE	293
ch5j2itr1 test run	303	TESTRUN EXAMPLE	289
ch5j2itr2 test run	304	ONESIDE OPTION	104
ch5j2tr1 test run	299	TESTRUN EXAMPLE	354
ch5j2tr2 test run	301	CUS KEYWORD	
ch5j2tr3 test run	302	DEFINITION	200
CHAIN UTILITY SUBPROGRAMS	81	CUSSPL SUBPROGRAM	395
CHECK KEYWORD		CVCOOR SUBPROGRAM	395
DEFINITION	127	CVT	34
CHKFRE SUBPROGRAM	394	TESTRUN EXAMPLE	290
CHKUP SUBPROGRAM	394	CVT KEYWORD	
CLASS SUBPROGRAM	394	DEFINITION	200
CLASSVIB KEYWORD		TEST RUN EXAMPLE	285, 295
DEFINITION	127	TESTRUN EXAMPLE	289, 294
clhbrtr1 test run	330	cwmctr1 test run	339
clhbrtr2 test run	332	cwmctr2 test run	340
clhbrtr3 test run	333	cwmctr3 test run	341
clhbrtr4 test run	334	D parameter	148
cmctr1 test run	336	DATTIM SUBPROGRAM	29, 395
cmctr2 test run	337	DAXPY SUBPROGRAM	395
cmctr3 test run	338	DDOT SUBPROGRAM	397
CMS SUBPROGRAM	394	DEFAULT SUBPROGRAM	396
COG SUBPROGRAM	395	DEFENG SUBPROGRAM	396
COHBT1 SUBPROGRAM	395	DEFGEN SUBPROGRAM	396
COHBT2 SUBPROGRAM	395	DEFLT40 SUBPROGRAM	396
COLSHF SUBPROGRAM	395	DEFOPT SUBPROGRAM	396
common blocks	30, 78	DEFPAT SUBPROGRAM	396
comp_pot.jc	441	DEFRAT SUBPROGRAM	396
comp_src.jc	441	DEFSEC SUBPROGRAM	396
COMT	56, 58	DEFSTA SUBPROGRAM	396
CONOUT SUBPROGRAM	395	DEFTUN SUBPROGRAM	396
CONSTANT KEYWORD		DELEX1 KEYWORD	
DEFINITION	148	DEFINITION	212
CONV40 SUBPROGRAM	395	DELTA2 KEYWORD	48, 102
Conventional Transition State Theory	12	DELTA2 PARAMETER	171
COORD KEYWORD		DELTAE KEYWORD	
DEFINITION	169	DEFINITION	211
COORD UTILITY SUBPROGRAMS	80	DEMIN KEYWORD	
Coordinates		DEFINITION	148
Cartesian	49	DERIV2 SUBPROGRAM	396
Curvilinear	49	DERV SUBPROGRAM	396

DERV24 SUBPROGRAM	396	EDGEOK KEYWORD	
DERVAG SUBPROGRAM	396	DEFINITION	201
DETM KEYWORD	431	EFFBATH SUBPROGRAM	398
DEFINITION	250	EFMAIN SUBPROGRAM	398
TESTRUN EXAMPLE	350	EFOPT KEYWORD	
detmi utility program	431	DEFINITION	141
DFGN31 SUBPROGRAM	396	EFORMD SUBPROGRAM	398
DGAMMA SUBPROGRAM	397	EFOVLP SUBPROGRAM	398
DGEDI SUBPROGRAM	397	EFPHES SUBPROGRAM	398
DGEFA SUBPROGRAM	397	EHOOK SUBPROGRAM	85, 398
DIATOM KEYWORD	67	EIGENVECTOR KEYWORD	
DEFINITION	149	DEFINITION	150
DIATOM SUBPROGRAM	67, 397	ELEC KEYWORD	
DIFFD KEYWORD	48, 102	DEFINITION	150
DIFFD PARAMETER	171	electronic structure input	13, 68, 365, 373
DIGISTR SUBPROGRAM	397	Electronic structure input file	
DIST SUBPROGRAM	397	anharmonicities	90
DL KEYWORD	105	data interpolation	92
DEFINITION	128	Dummy potential energy surface	77, 88
DLX2 KEYWORD		frequencies	89
DEFINITION	141	moment of inertia	89
DLX3 KEYWORD		unit fu30	88, 213
DEFINITION	170	unit fu40	88
DOGLEG SUBPROGRAM	397	user-supplied information	90
DOPNM SUBPROGRAM	397	ELLIP SUBPROGRAM	398
DOREPR SUBPROGRAM	397	ELRPH SUBPROGRAM	399
DOREST SUBPROGRAM	397	EMIN KEYWORD	
DORODS SUBPROGRAM	397	DEFINITION	190
DORPH SUBPROGRAM	397	ENDRODS SUBPROGRAM	399
DOSAFR SUBPROGRAM	397	ENDSLP SUBPROGRAM	399
DOSAGE SUBPROGRAM	397	ENERGETICS SECTION	133
DOTPRD SUBPROGRAM	398	ENERGY KEYWORD	
DPMPAR SUBPROGRAM	397	DEFINITION	151
DPR123 SUBPROGRAM	398	ENERXN KEYWORD	
DPSWAP SUBPROGRAM	398	DEFINITION	250
DQQP KEYWORD		ENESAD KEYWORD	
DEFINITION	149	DEFINITION	250
RELATED KEYWORDS	154	ENORM SUBPROGRAM	399
DSCAL SUBPROGRAM	398	ENROUT SUBPROGRAM	399
DSWAP SUBPROGRAM	398	EPART SUBPROGRAM	399
dumpot.f	77, 88	EPARTI SUBPROGRAM	399
E1GEOM SECTION		Equilibrium constant	57
DEFINITION	209	ERF SUBPROGRAM	399
E1GRAD SECTION		ES1 KEYWORD	
DEFINITION	209	DIFFD PARAMETER	
E1HESS SECTION		TESTRUN EXAMPLE	294
DEFINITION	210	ES1 KEYWORD	
EACT KEYWORD		DELTA2 PARAMETER	
DEFINITION	201	TESTRUN EXAMPLE	294
TESTRUN EXAMPLE	285, 295	METHOD DESCRIPTION	102
EBND SUBPROGRAM	398	METHOD DESCRIPTION	48
ECKARS SUBPROGRAM	398	TESTRUN EXAMPLE	294, 299, 355
ECKART SUBPROGRAM	398	ES1OPT KEYWORD	
ECKRT SUBPROGRAM	398	DEFINITION	171

ESD KEYWORD	67, 102	TESTRUN EXAMPLE	353, 355, 358
TESTRUN EXAMPLE	289, 302, 348	DEFINITION	174
Euler stabilization method	48	RELATED KEYWORD	170
Euler steepest-descents (ESD)method	47	FITMAX SUBPROGRAM	401
EVIB SUBPROGRAM	399	FITMX SUBPROGRAM	401
EWKB SUBPROGRAM	399	FIVFT SUBPROGRAM	401
EXCITED KEYWORD		FIVPT SUBPROGRAM	401
DEFINITION	190	FORMA SUBPROGRAM	401
EXFIRST KEYWORD	96	FORMBT SUBPROGRAM	401
DEFINITION	171	FORMF SUBPROGRAM	401
TESTRUN EXAMPLE	342, 353	FORMF2 SUBPROGRAM	401
Expert users	276, 278	FORMG SUBPROGRAM	401
EXPND SUBPROGRAM	399	FORWARDK KEYWORD	
EXSECOND KEYWORD	96	DEFINITION	201
DEFINITION	173	TESTRUN EXAMPLE	342, 343
TESTRUN EXAMPLE	342	FPRINT KEYWORD	
EXTRAP SUBPROGRAM	399	DEFINITION	137
EXTRP SUBPROGRAM	399	FREQ KEYWORD	
EXTRP40 SUBPROGRAM	399	DEFINITION	151
EZERO KEYWORD	75	FREQIMAG KEYWORD	
CALCULATE OPTION		DEFINITION	251
TESTRUN EXAMPLE	330, 353	FREQMAT KEYWORD	106
DEFINITION	134	DEFINITION	251
READ OPTION		TESTRUN EXAMPLE	337, 350
TESTRUN EXAMPLE	332	FREQSCALE KEYWORD	
EZUNIT KEYWORD		DEFINITION	174
DEFINITION	134	FREQUNIT KEYWORD	
F1DI SUBPROGRAM	399	DEFINITION	151
F1DIM SUBPROGRAM	399	FRPRMN SUBPROGRAM	401
FACORD SUBPROGRAM	399	FRSORT SUBPROGRAM	401
FCINPT SUBPROGRAM	400	FU parameters	
FCMEP SUBPROGRAM	400	GAMESOLRATE	15
FCN SUBPROGRAM	400	GAUSQD SUBPROGRAM	401
FCRATE SUBPROGRAM	400	GAUSSQ SUBPROGRAM	401
FCSCALE KEYWORD		GAUSSRATE	15
DEFINITION	173	GBSLVE SUBPROGRAM	401
FCSCL SUBPROGRAM	400	GBTQL2 SUBPROGRAM	402
FDIAG SUBPROGRAM	400	<i>GCOMP KEYWORD</i>	
FDIAG2 SUBPROGRAM	400	DEFINITION	141
FDJAC1 SUBPROGRAM	400	GEMM SUBPROGRAM	402
FICLSE SUBPROGRAM	400	GENERAL SECTION	124
File fu30 input		Generalized transition state theory	34
lopt options	213	Canonical variational theory (CVT)	34, 36
File usage	443	Improved canonical variational theory (ICVT)	34
findb utility program	436	Microcanonical variational theory (μ VT)	34
testrun example	337	GENSEC40 SUBPROGRAM	402
FINDL SUBPROGRAM	400	GEOM KEYWORD	75
findl utility program	436	DEFINITION	151
testrun example	337	Geometry optimization keywords	
FINDLV SUBPROGRAM	400	polyrate	140
FINOUT SUBPROGRAM	57, 69, 400	GEOMUNIT KEYWORD	
FIOPEN SUBPROGRAM	401	DEFINITION	128
FIRSTSTEP KEYWORD	103	GETFRQ SUBPROGRAM	402
CUBIC OPTION			

GFDIAG SUBPROGRAM	402	ICINT SUBPROGRAM	403
GHOOK SUBPROGRAM	86, 402	ICL KEYWORD	106
GIVTST SUBPROGRAM	402	ICODAL SUBPROGRAM	403
GRAD SUBPROGRAM	402	ICOPOT KEYWORD	105
GRADDR SUBPROGRAM	402	ICOUTRP SUBPROGRAM	403
GSETA SUBPROGRAM	402	ICR KEYWORD	106
GSETM SUBPROGRAM	402	ICSAVE SUBPROGRAM	404
GSETP SUBPROGRAM	402	ICSVRP SUBPROGRAM	404
GSPEC KEYWORD		ICVT	34
DEFINITION	201	TESTRUN EXAMPLE	289, 290, 302
TESTRUN EXAMPLE	357	ICVT KEYWORD	
GTEMP KEYWORD		DEFINITION	202
DEFINITION	201	TEST RUN EXAMPLE	285, 295
TESTRUN EXAMPLE	354	ICVT/MEPSAG	56
GTLOG KEYWORD		IDAMAX SUBPROGRAM	404
DEFINITION	202	IDIRECT KEYWORD	
GTST Book Chapter	32	DEFINITION	175
h2nitr1 test run	378	ILCT	64
h3tr1 test run	342	IMOM40 SUBPROGRAM	404
h3tr2 test run	343	Improved Canonical Variational Theory	12
h3tr3 test run	344	INCLUDE FILES	30, 390, 429
h3tr4 test run	345	COMMON.INC	30, 429, 495
HARMONIC KEYWORD		PARAM.INC	30, 279, 429, 443
DEFINITION	152	PERCON.INC	30, 429
HBT SUBPROGRAM	403	INERTA SUBPROGRAM	404
HEADR SUBPROGRAM	403	INH KEYWORD	67
HESS KEYWORD		DEFINITION	175
DEFINITION	175	TESTRUN EXAMPLE	294, 342
HESSCAL KEYWORD		INI KEYWORD	
DEFINITION	137	DEFINITION	175
HESSIAN KEYWORD		INITGEO KEYWORD	
DEFINITION	152	DEFINITION	152
HHOOK SUBPROGRAM	86, 403	INITZE SUBPROGRAM	404
HINDRT SUBPROGRAM	403	Input	
HINDRT1 SUBPROGRAM	403	file fu29	208
hnitr1 test run	377	file fu30	88, 213
ho2tr1 test run	346	file fu31	222
ho2tr2 test run	347	file fu40	231
ho2tr3 test run	348	file fu5	121
ho2tr4 test run	349	section descriptions	123
ho2tr5 test run	350	file fu50	248
HQSC SUBPROGRAM	403	file fu51	259
HREC KEYWORD		INTAB SUBPROGRAM	404
DEFINITION	142	INTABS SUBPROGRAM	404
HRMI KEYWORD	433	INTCOR SUBPROGRAM	404
DEFINITION	251	INTDEF KEYWORD	
hrotor utility program	433	DEFINITION	175
HRPART SUBPROGRAM	403	INTEGR SUBPROGRAM	405
HRSET SUBPROGRAM	403	INTERPOLATED OPTIMIZED CORRECTIONS	
HVAL SUBPROGRAM	403	(VTST-IOC)	62, 105
HYBRD SUBPROGRAM	403	INTERPOLATED VTST	58
HYBRD1 SUBPROGRAM	403	INTERSC SUBPROGRAM	404
ICFDIAG SUBPROGRAM	403	INTEUL SUBPROGRAM	405
ICFDRP SUBPROGRAM	403	INTF31 SUBPROGRAM	404

INTFNC SUBPROGRAM	405	LCNWTN SUBPROGRAM	406
INTMU KEYWORD		LCPATH SUBPROGRAM	406
DEFINITION	176	LCPROB SUBPROGRAM	407
INTPM SUBPROGRAM	405	LCPROJ SUBPROGRAM	407
INTPM2 SUBPROGRAM	405	LCSET SUBPROGRAM	407
INTRPL SUBPROGRAM	405	LCSINT SUBPROGRAM	407
INTRVL SUBPROGRAM	404	LCSTX SUBPROGRAM	407
INVRT1 SUBPROGRAM	404	LCT3 KEYWORD	
INVRT2 SUBPROGRAM	404	DEFINITION	190
INVRT3 SUBPROGRAM	404	LCT4	119
IOCOPT KEYWORD		LCT4 KEYWORD	
DEFINITION	129	DEFINITION	190
TESTRUN EXAMPLE	337	LCTCOR KEYWORD	
ZERO OPTION		DEFINITION	252
TESTRUN EXAMPLE	379	LCTDETAIL KEYWORD	
IRCX SUBPROGRAM	405	DEFINITION	191
IRFTG0 SUBPROGRAM	405	OUTPUT DETAILS	265
IRPH SUBPROGRAM	405	LCTOPT KEYWORD	
IRPHZ SUBPROGRAM	405	DEFINITION	192
Isoinertial coordinates	44, 67	INTERPOLATE OPTION	
ISWAP SUBPROGRAM	405	TESTRUN EXAMPLE	332
ISWP KEYWORD		TESTRUN EXAMPLE 285, 290, 292, 295, 297	
DEFINITION	176	LCTP SUBPROGRAM	407
ISWR KEYWORD		LCVCOR SUBPROGRAM	407
DEFINITION	176	LCVIB SUBPROGRAM	407
IVTMA SUBPROGRAM	405	LCVMOD SUBPROGRAM	407
IVTMD SUBPROGRAM	405	LCXINT SUBPROGRAM	407
IVTMH SUBPROGRAM	405	LCXTAU SUBPROGRAM	407
IVTST SECTION		LCZCRT SUBPROGRAM	407
DEFINITION	210	LIN SUBPROGRAM	408
IVTST0 KEYWORD	59	LIN2 SUBPROGRAM	408
DEFINITION	129	LINAXIS KEYWORD	
IVTST0 SUBPROGRAM	405	DEFINITION	153
IVTST0FREQ KEYWORD		LININT SUBPROGRAM	408
DEFINITION	178	link_exe.jc	442
INCOMPATABILITY	178	LINMN SUBPROGRAM	408
IVTST1 KEYWORD	59	LINMR SUBPROGRAM	408
DEFINITION	129	LOCATE SUBPROGRAM	408
IVTST-M	60, 112	LOCS SUBPROGRAM	408
IVTSTMOPT KEYWORD		LOPT OPTIONS	213
DEFINITION	177	LOPT40 SUBPROGRAM	408
JACSCF SUBPROGRAM	405	LOWFREQ KEYWORD	
KAPA SUBPROGRAM	405	DEFINITION	252
KAPPAS SUBPROGRAM	406	LSAME SUBPROGRAM	408
KAPVA SUBPROGRAM	69, 406	LUBKSB SUBPROGRAM	408
KG1 SUBPROGRAM	406	LUDCMP SUBPROGRAM	408
Large-curvature Tunneling	12, 41	MAIN SUBPROGRAM	408
LCG3	41, 55	Manuals	32
LCG3 SUBPROGRAM	406	MATINV SUBPROGRAM	408, 409
LCG4	41, 55	MATX SUBPROGRAM	408
LCGIT SUBPROGRAM	406	MDMOVIE KEYWORD	
LCGSRT SUBPROGRAM	406	DEFINITION	129
LCGTH SUBPROGRAM	406	Memory usage	446
LCNORM SUBPROGRAM	406	MEPINV SUBPROGRAM	408

MEPOUT SUBPROGRAM	408	NEXTPT KEYWORD	
MEPSAG	41, 54	DEFINITION	179
MEPSRT SUBPROGRAM	409	NEXTPT SUBPROGRAM	409
MEPTYPEP KEYWORD	105	nh3tr1 test run	351
DEFINITION	253	nh3tr2 test run	352
TESTRUN EXAMPLE	337, 350	NITER KEYWORD	
MEPTYPER KEYWORD	105	DEFINITION	142
DEFINITION	253	NOALLEXCIT KEYWORD	
TESTRUN EXAMPLE	337, 350	TESTRUN EXAMPLE	343
microcanonical optimized multidimensional		NOCVT KEYWORD	
tunneling	12	TESTRUN EXAMPLE	339
Microcanonical Variational Theory	12	NONLINEAR KEYWORD	
MINIMUM ENERGY PATH (MEP)		TESTRUN EXAMPLE	342, 384
CALCULATING	47	NOOPT KEYWORD	
MIRP KEYWORD		TESTRUN EXAMPLE	346, 347, 348
DEFINITION	254	NOPROJECT KEYWORD	
MIT S KEYWORD		TESTRUN EXAMPLE	355
DEFINITION	254	NORMOD SUBPROGRAM	67, 68, 409
MNBRAK SUBPROGRAM	409	NOROUT SUBPROGRAM	409
MOPAC program	15	NOSADDLE KEYWORD	50
MOPEG SUBPROGRAM	409	NOST31 SUBPROGRAM	409
MOPHES SUBPROGRAM	409	NQE PARAMETER	194
MOPOPT SUBPROGRAM	409	NQTH	55
MOPSET SUBPROGRAM	409	NQTH PARAMETER	194
MORATE	15	NSEG2	55
MORMODEL KEYWORD	98	NSEGBOLTZ KEYWORD	
DEFINITION	153	TESTRUN EXAMPLE	342
MORSEI OPTION		NSEGHETA KEYWORD	
TEST RUN EXAMPLE	384, 385, 386, 387	TESTRUN EXAMPLE	342
TESTRUN EXAMPLE	346, 347, 349, 354	NSTATE SUBPROGRAM	409
MORSEIIIA OPTION		NSTEPS KEYWORD	
TEST RUN EXAMPLE	342	DEFINITION	179
RELATED KEYWORDS	153, 154	NUMFRQ SUBPROGRAM	410
MORSE KEYWORD	98	NUMSTEP KEYWORD	101
DEFINITION	153	DEFINITION	137
RELATED KEYWORDS	153	RELATED KEYWORDS	138
TESTRUN EXAMPLE	342, 346, 347, 349, 384, 385, 386, 387	NUMTYP KEYWORD	101
MORSEQQ KEYWORD	98	DEFINITION	138
DEFINITION	154	OBTSRT SUBPROGRAM	410
RELATED KEYWORDS	148, 153	oh3fu30tr1 testrun	363
TESTRUN EXAMPLE	354	oh3fu30tr2 testrun	364
MUCDSC SUBPROGRAM	409	oh3fu30tr3 testrun	365
MUSC SUBPROGRAM	409	oh3fu30tr4 testrun	366
MUSCCD SUBPROGRAM	409	oh3fu30tr5 testrun	367
MUVT KEYWORD		oh3fu30tr6 test run	368
DEFINITION	202	oh3fu31tr1 test run	369
MUVTOPT KEYWORD		oh3fu31tr2 test run	370
DEFINITION	203	oh3fu40tr1 testrun	371
TESTRUN EXAMPLE	346, 347	oh3fu40tr2 testrun	372
MXLNEQ SUBPROGRAM	409	oh3fu40tr3 testrun	373
NATOMS VARIABLE	76	oh3fu40tr4 testrun	374
NEARW SUBPROGRAM	409	oh3fu40tr5 testrun	375
NEWT SUBPROGRAM	67, 409	oh3fu40tr6 test run	376
		oh3tr1 testrun	353

oh3tr10 testrun	362	PCFREQS KEYWORD	
oh3tr2 testrun	354	DEFINITION	255
oh3tr3 test run	100	PCINFO KEYWORD	
oh3tr3 testrun	355	DEFINITION	255
oh3tr4 testrun	356	PHASE KEYWORD	
oh3tr5 testrun	357	SOLID OPTION	
oh3tr6 testrun	358	TESTRUN EXAMPLE	377, 378
oh3tr7 testrun	359	PHID SUBPROGRAM	410
oh3tr8 testrun	360	PHSINT SUBPROGRAM	410
oh3tr9 test run	361	POLINT SUBPROGRAM	410
ohcltr1 test run	384	POLYAT SUBPROGRAM	67, 411
ohcltr2 test run	385	POLYRATE	
ohcltr3 test run	386	DESCRIPTION OF SUBPROGRAMS	391
ohcltr4 test run	387	DIRECTORY STRUCTURE	266
OHOOK SUBPROGRAM	87, 410	HIDDEN FILE	440
OPENFI SUBPROGRAM	410	INCLUDE FILES	429
OPT50 SUBPROGRAM	410	INSTALLATION PROCEDURES	266
OPT50C SUBPROGRAM	410	INSTRUCTIONS FOR GENERAL USE	273
OPTIMIZATION SECTION	139	LICENSING	22
OPTION SUBPROGRAM	65, 410	MEMORY REQUIREMENTS	446
OPTMIN KEYWORD		PROGRAM STRUCTURE	65
OPTIMIZATION		REVISION HISTORY	470
DEFINITION	142	RUNNING TEST RUNS	272
OPTTS KEYWORD		TESTING PROCEDURES	271
OPTIMIZATION		potcomp.jc	441
DEFINITION	143	Potential energy surface	13, 390
ORD31 SUBPROGRAM	410	POTENTIAL KEYWORD	88
Orientation	91	DEFINITION	135
Output		UNIT30 OPTION	
files fu41-fu47	265	TESTRUN EXAMPLE	363, 364, 365, 366, 367, 371, 372
OVLRLP SUBPROGRAM	410	UNIT40 OPTION	
P1FREQ SECTION		TESTRUN EXAMPLE	373, 374, 375
DEFINITION	212	POTINF KEYWORD	78
P1HESS SECTION		DEFINITION	180
DEFINITION	210	PRCORD SUBPROGRAM	411
P2FREQ SECTION		PRDELG KEYWORD	
DEFINITION	212	DEFINITION	203
P2HESS SECTION		PRDISTMX KEYWORD	
DEFINITION	210	DEFINITION	180
PAGEM KEYWORD	102	PREGPT SUBPROGRAM	411
TESTRUN EXAMPLE	285, 293, 295, 342, 356	PREP SUBPROGRAM	84, 411
Partition function		PREPJ SUBPROGRAM	85, 411
electronic	39	PRGIGT KEYWORD	
rotational	34, 40	DEFINITION	203
vibrational	34, 40, 54	PRINT KEYWORD	
PATH SECTION	162	DEFINITION	143, 212
EXTRAPOLATION KEYWORDS	167	PRINTSTEP KEYWORD	
INTEGRATOR KEYWORDS	166	DEFINITION	180
NES KEYWORDS	168	PRMEP SUBPROGRAM	411
PRINT KEYWORDS	168	PRMODE KEYWORD	
STEP SIZE KEYWORDS	166	DEFINITION	154
PATH SUBPROGRAM	410	PRNTBT SUBPROGRAM	411
PCFREQA KEYWORD		PRNTFC SUBPROGRAM	411
DEFINITION	254		

PROD1 SECTION	145	PSATX SUBPROGRAM	411
PROD2 SECTION	145	PSATX2 SUBPROGRAM	412
PRODFREQ KEYWORD	106	PTORS SUBPROGRAM	412
DEFINITION	256	PTQVIB SUBPROGRAM	412
Product		PTSEC40 SUBPROGRAM	412
force constant matrix	45	PYC1 SUBPROGRAM	412
geometry optimization	43	PYC1P SUBPROGRAM	412
zero point energy	46	PYC2 SUBPROGRAM	412
PROGRAM FILES		PYC2P SUBPROGRAM	412
ACESPOLY	429	PYCOEF SUBPROGRAM	412
EF 430		QFORM SUBPROGRAM	413
ENERGETICS	430	QQPOT SUBPROGRAM	413
FROMBLAS	430	QQSEMI KEYWORD	98
GIVTST	430	DEFINITION	154
HEADR	430	RELATED KEYWORDS	149
HOOKS	430	TESTRUN EXAMPLE	302, 342
INTBSV1	430	QQWKB KEYWORD	98
INTBSV2	430	DEFINITION	154
INTBSV3	430	RELATED KEYWORDS	149
INTERFACE	430	TESTRUN EXAMPLE	348
IVTSTM	430	QRFAC SUBPROGRAM	413
MAIN	430	QRSENE SUBPROGRAM	413
MOPOLY	430	QRST	
POLY31	431	DIMENSION REQUIREMENTS	31
POLY40	431	KEYWORD DEFINITION	193
POLYAG	431	QSLVE SUBPROGRAM	413
POLYHL	431	QTQVIB SUBPROGRAM	413
POLYMQ	431	QUAD KEYWORD	
POLYRR	431	DEFINITION	194
POLYSZ	431	QUADFT SUBPROGRAM	413
RTPJAC	431	QUADR SUBPROGRAM	413
PROGRAM SUBPROGRAMS		QUADTW SUBPROGRAM	413
SETUP	429	R1FREQ SECTION	
SURF	429	DEFINITION	212
PROJECT SUBPROGRAM	68, 411	R1HESS SECTION	
PROJECT KEYWORD		DEFINITION	210
DEFINITION	154	R1MPYQ SUBPROGRAM	413
TESTRUN EXAMPLE	343	R1UPDT SUBPROGRAM	413
PROJF SUBPROGRAM	411	R2FREQ SECTION	
PROJTI SUBPROGRAM	411	DEFINITION	212
PRPART KEYWORD		R2HESS SECTION	
DEFINITION	204	DEFINITION	210
PRPATH KEYWORD		RACES SUBPROGRAM	413
DEFINITION	180	RALL31 SUBPROGRAM	413
PRSAVEMODE KEYWORD		RANALY SUBPROGRAM	413
DEFINITION	181	RANGEIC KEYWORD	
PRSAVERP KEYWORD	77	DEFINITION	257
DEFINITION	181	RATE SECTION	196
PRSQ SUBPROGRAM	411	print keywords	199
PRVIB KEYWORD		RATE SUBPROGRAM	57, 69, 413
DEFINITION	204	RATOMS SUBPROGRAM	414
PSAG SUBPROGRAM	411	RBASIS SUBPROGRAM	413
PSATP SUBPROGRAM	411	RBATH SUBPROGRAM	413
PSATP2 SUBPROGRAM	411	RCFREQA KEYWORD	

DEFINITION	256	REORDER KEYWORD	
RCFREQS KEYWORD		DEFINITION	181
DEFINITION	256	REPFL SUBPROGRAM	415, 419
RCINFO KEYWORD		RES1 SUBPROGRAM	415
DEFINITION	257	RESTART KEYWORD	72, 109
RCNST SUBPROGRAM	414	DEFINITION	130
RD40EXTR SUBPROGRAM	414	TESTRUN EXAMPLE	285, 287, 290, 340, 341
RD40GEN SUBPROGRAM	414	RESTOR SUBPROGRAM	65, 415
RD40RP SUBPROGRAM	414	RESTRT SUBPROGRAM	415
RDETIL SUBPROGRAM	414	RETRY KEYWORD	
RDHESS SUBPROGRAM	414	DEFINITION	143
RDIATM SUBPROGRAM	414	Reverse rate constant	36, 56, 106
RDIFFU SUBPROGRAM	414	revision history	470
RDLN SUBPROGRAM	414	REVKEXP KEYWORD	
REACFREQ KEYWORD		DEFINITION	204
DEFINITION	258	TESTRUN EXAMPLE	346, 347, 348
REACT SUBPROGRAM	67, 414	REXTRP SUBPROGRAM	415
REACT1 SECTION	145	REZERO SUBPROGRAM	415
REACT2 SECTION	145	REZERU SUBPROGRAM	415
Reactant		RFCFAC SUBPROGRAM	415
force constant matrix	45	RFREQ SUBPROGRAM	415
geometry optimization	43	RGEM31 SUBPROGRAM	415
properties	43	RGEN SUBPROGRAM	416
zero point energy	46	RGEN31 SUBPROGRAM	416
Reaction coordinate	35, 45	RGENER SUBPROGRAM	416
determining the sign of	71	RGEOM SUBPROGRAM	416
Reaction path	35, 46, 47, 56	RGRD SUBPROGRAM	416
Euler stabilization method	48	RGRD31 SUBPROGRAM	416
negative generalized normal mode eigenvalue	56	RGSAD SUBPROGRAM	416
projected force constant matrix	49	RGSPEC SUBPROGRAM	416
save grid	49, 68	RGSZM SUBPROGRAM	416
step sizes	103	RGTEMP SUBPROGRAM	416
Reaction rate methods keywords	198	RGZMAT SUBPROGRAM	416
Reaction rates	54	RHES31 SUBPROGRAM	416
generalized free energy of activation	54	RHESS SUBPROGRAM	416
transmission coefficients	54	RHESSC SUBPROGRAM	416
transmission probabilities	54	RHFSEC SUBPROGRAM	416
Boltzmann average	52	RHSS31 SUBPROGRAM	416
READ5 SUBPROGRAM	65, 414	RICOPT SUBPROGRAM	417
READIC SUBPROGRAM	414	RLCT SUBPROGRAM	417
READINT SUBPROGRAM	414	RLINE SUBPROGRAM	417
READLN SUBPROGRAM	414	RLOWFR SUBPROGRAM	417
REARG1 SUBPROGRAM	414	RMUVT SUBPROGRAM	417
RECS31 SUBPROGRAM	414	RODS KEYWORD	
REDGEO SUBPROGRAM	65, 415	DEFINITION	181
REDMU SUBPROGRAM	415	ROPT SUBPROGRAM	417
References		ROPT31 SUBPROGRAM	417
bibliography	455	ROPTOP SUBPROGRAM	417
convention explained	33	rotate utility program	436
footnotes to text	452	RPART SUBPROGRAM	417
REFOPT SUBPROGRAM	415	RPARTI SUBPROGRAM	417
RELEC SUBPROGRAM	415	RPATH SUBPROGRAM	417
RENERG SUBPROGRAM	415	RPH31 SUBPROGRAM	417
		RPH40 SUBPROGRAM	417

RPHAIT SUBPROGRAM	417	RVARY SUBPROGRAM	421
RPHB SUBPROGRAM	417	RVPERT SUBPROGRAM	421
RPHDXN SUBPROGRAM	417	RVRANG SUBPROGRAM	421
RPHEXP SUBPROGRAM	418	RWKB SUBPROGRAM	421
RPHFIT SUBPROGRAM	418	RWORD SUBPROGRAM	421
RPHFT2 SUBPROGRAM	418	RZVAR SUBPROGRAM	421
RPHFT3 SUBPROGRAM	418	SADDLE KEYWORD	47
RPHG1 SUBPROGRAM	418	DEFINITION	182
RPHG2 SUBPROGRAM	418	SADDLE SUBPROGRAM	47, 421
RPHG5 SUBPROGRAM	418	SADENG SUBPROGRAM	421
RPHINT SUBPROGRAM	418	SADFREQ KEYWORD	
RPHMEF SUBPROGRAM	418	DEFINITION	258
RPHNWT SUBPROGRAM	418	SADSEC40 SUBPROGRAM	421
RPHRD1 SUBPROGRAM	418	Sample test runs	281
RPHRD2 SUBPROGRAM	418	Save grid	49, 68
RPHSET SUBPROGRAM	418	SCALE KEYWORD	
RPHTRF SUBPROGRAM	418	DEFINITION	143
RPHTRX SUBPROGRAM	419	SCALEMASS KEYWORD	68
RPHWRT SUBPROGRAM	419	DEFINITION	182
RPL KEYWORD	109	TESTRUN EXAMPLE	288
<i>RPM KEYWORD</i>		SCLPT KEYWORD	
DEFINITION	182	DEFINITION	182
RPNT31 SUBPROGRAM	418	Scripts	
RPOTET SUBPROGRAM	419	comp_pot.jc	441
RPOTGE SUBPROGRAM	419	comp_src.jc	441
RPRMEP SUBPROGRAM	419	link_exe.jc	442
RPRVIB SUBPROGRAM	419	potcomp.jc	441
RPSEC40 SUBPROGRAM	419	SCT KEYWORD	
RQCOM SUBPROGRAM	419	DEFINITION	194
RQRST SUBPROGRAM	419	TESTRUN EXAMPLE	285, 295
RQUAD SUBPROGRAM	419	SCTOPT KEYWORD	
RRANGE SUBPROGRAM	419	DEFINITION	194
RRATE SUBPROGRAM	419	LAGRANGE OPTION	103
RRPM SUBPROGRAM	419	TESTRUN EXAMPLE	293, 299
RRSTRT SUBPROGRAM	419	TESTRUN EXAMPLE	288
RSCT SUBPROGRAM	419	TESTRUN EXAMPLE	
RSECND SUBPROGRAM	419	SPLINE OPTION	289
RSFRST SUBPROGRAM	420	<i>SDEB1 KEYWORD</i>	
RSP SUBPROGRAM	420	DEFINITION	183
RSPDRV SUBPROGRAM	420	<i>SDEB2 KEYWORD</i>	
RSPECL SUBPROGRAM	420	DEFINITION	183
RSPP SUBPROGRAM	420	SDSTART KEYWORD	
RSSTOP SUBPROGRAM	420	DEFINITION	144
RST SUBPROGRAM	420	SECCEN SUBPROGRAM	421
RSTAT SUBPROGRAM	420	SECCEP SUBPROGRAM	421
RSTATE SUBPROGRAM	420	Second derivatives	
RTEMP SUBPROGRAM	420	central differences	104
RTITLE SUBPROGRAM	420	numerical	101
RTOR SUBPROGRAM	420	numerical fourth order polynomial	101
RTOROPT SUBPROGRAM	420	quadratic fit	101
RTPJAC SUBPROGRAM	420	SECOND SECTION	136
RTUNNL SUBPROGRAM	420	SECOND SECTION KEYWORDS	
RUNM31 SUBPROGRAM	420	POLYRATE	136
RVAR6 SUBPROGRAM	421	SENOUT SUBPROGRAM	421

set_machine.jc	440	DEFINITION	185
SETANH SUBPROGRAM	421	TESTRUN EXAMPLE	290, 294, 342
SETLGS SUBPROGRAM	421	SSX SUBPROGRAM	423
SETUP SUBPROGRAM	75, 82, 422	START SECTION	145
SETVAR SUBPROGRAM	422	STAT31 SUBPROGRAM	423
SEX1 KEYWORD		STATE KEYWORD	
DEFINITION	212	ADIAB OPTION	
SFIRST KEYWORD	103	TESTRUN EXAMPLE	356
DEFINITION	183	DEFINITION	205
TESTRUN EXAMPLE	299, 353	DIAB OPTION	
SGND3V SUBPROGRAM	422	TESTRUN EXAMPLE	357
SHFTSAD SUBPROGRAM	422	STATEOPT KEYWORD	
SIGMAF KEYWORD		DEFINITION	206
DEFINITION	204	STATUS KEYWORD	
SIGMAR KEYWORD		DEFINITION	155
DEFINITION	205	STAUV SUBPROGRAM	424
SIGN KEYWORD	71	STLGSJ SUBPROGRAM	424
DEFINITION	183	STOR SUBPROGRAM	424
SIGS SUBPROGRAM	422	<i>STPTOL KEYWORD</i>	
SIMPSN SUBPROGRAM	422	<i>DEFINITION</i>	144
SINE SUBPROGRAM	422	STVARJ SUBPROGRAM	424
Small-curvature Tunneling	12, 41	SUPERMOL KEYWORD	
SNHCSH SUBPROGRAM	422	DEFINITION	130
SORT31 SUBPROGRAM	422	SURF SUBPROGRAM	75, 77, 79, 82, 424
SORTG SUBPROGRAM	422	SVBCMP SUBPROGRAM	424
SPECIES KEYWORD		SVBKSJ SUBPROGRAM	424
DEFINITION	155	SYMMETRY KEYWORD	
SPECSTOP KEYWORD		DEFINITION	186
DEFINITION	184	TESTRUN EXAMPLE	344, 345
SPL1B1 SUBPROGRAM	422	TABLE SUBPROGRAM	424
SPL1B2 SUBPROGRAM	422	TAUV SUBPROGRAM	424
SPL1D1 SUBPROGRAM	422	TEMP KEYWORD	
SPL31 SUBPROGRAM	423	DEFINITION	206
SPLM31 SUBPROGRAM	423	RELATED KEYWORDS	200, 201
SPLNB SUBPROGRAM	423	TESTRUN EXAMPLE	285, 295, 346, 347
SPLNF SUBPROGRAM	423	THETA2 SUBPROGRAM	424
SPLNM SUBPROGRAM	423	THETA3 SUBPROGRAM	424
SPLNMF SUBPROGRAM	423	TITLE KEYWORD	
SPLNV SUBPROGRAM	423	DEFINITION	131
SPLRMI SUBPROGRAM	423	TITLE SUBPROGRAM	424
SPLSPV SUBPROGRAM	423	TOR KEYWORD	
SPLV31 SUBPROGRAM	423	DEFINITION	156
SPRNT KEYWORD		TESTRUN EXAMPLE	340, 341, 379
DEFINITION	185	TOROPT KEYWORD	
SRANGE KEYWORD		DEFINITION	157
DEFINITION	185	TP SUBPROGRAM	424
TESTRUN EXAMPLE	349, 351, 352	TPCDSC SUBPROGRAM	424
SSAVE SUBPROGRAM	423	TQL2 SUBPROGRAM	425
SSPECIAL KEYWORD		TQLRAT SUBPROGRAM	425
DEFINITION	185	TRANFC SUBPROGRAM	425
TESTRUN EXAMPLE	348	TRANG SUBPROGRAM	425
SSPECPR KEYWORD		TRANLF SUBPROGRAM	425
DEFINITION	185	TRANS SUBPROGRAM	425
SSTEP KEYWORD	48, 67, 102	TRANS2 SUBPROGRAM	425

TRANSF SUBPROGRAM	425	VICLCG SUBPROGRAM	426
Transmission coefficients	37, 41, 54, 55	VIVT SUBPROGRAM	426
TRBAK3 SUBPROGRAM	425	VPART SUBPROGRAM	426
TRED3 SUBPROGRAM	425	VPARTI SUBPROGRAM	426
TREPT SUBPROGRAM	425	VPFEXP KEYWORD	
TRLFRP SUBPROGRAM	425	DEFINITION	207
TSC SUBPROGRAM	425	TESTRUN EXAMPLE	288, 293
TSFREQ SECTION		VRANGE KEYWORD	98
DEFINITION	212	DEFINITION	160
<i>TSGCOMP</i> KEYWORD		VSCALE KEYWORD	
DEFINITION	141	DEFINITION	186
TSHESS SECTION		VSPLI2 SUBPROGRAM	427
DEFINITION	210	VSPLIN SUBPROGRAM	427
TSHREC KEYWORD		VTMUSN SUBPROGRAM	427
DEFINITION	142	VTST rate constants	12
TSOUT SUBPROGRAM	57, 425	VTST-IOC	62
TSPSI SUBPROGRAM	425	VTST-ISPE	109
TSRATE SUBPROGRAM	57, 426	WELLP SECTION	145
TST		WELLR SECTION	145
TESTRUN EXAMPLE	290	WIGNER KEYWORD	
TST KEYWORD		DEFINITION	195
DEFINITION	207	Wigner tunneling	45, 54
TESTRUN EXAMPLE	339	WKB KEYWORD	98, 99
TUNNEL SECTION	187	DEFINITION	161
Tunneling		TESTRUN EXAMPLE	355
canonical optimized multidimensional	58	WKB METHOD	
interpolation of effective reduced mass	103	based on true potential	99
microcanonical optimized multidimensional	58	PRIMITATIVE	99
		UNIFORM	99
TURNPT SUBPROGRAM	426	WKBENE SUBPROGRAM	427
UNITMI KEYWORD		WKBPOT SUBPROGRAM	427
DEFINITION	258	WKBVIB SUBPROGRAM	427
TESTRUN EXAMPLE	350	WRITE62 KEYWORD	
UNITS KEYWORD		DEFINITION	131
DEFINITION	212	WRITEFU30 KEYWORD	
UPCAS SUBPROGRAM	426	DEFINITION	131
UPCSE SUBPROGRAM	426	WRITEFU31 KEYWORD	
UPDHR SUBPROGRAM	426	DEFINITION	132
UPDMI SUBPROGRAM	426	WRTHOK SUBPROGRAM	427
US KEYWORD		WRTWEL SUBPROGRAM	427
DEFINITION	207	WTITLE SUBPROGRAM	427
TESTRUN EXAMPLE	346, 347	WVAR6 SUBPROGRAM	427
Utility routines		XDOTP SUBPROGRAM	427
chain	81	XERBLA SUBPROGRAM	427
coord	80	XPROD SUBPROGRAM	427
VALVAG SUBPROGRAM	426	YDER24 SUBPROGRAM	427
VANHAR KEYWORD		YDERV2 SUBPROGRAM	427
DEFINITION	159	YNEWT SUBPROGRAM	428
TESTRUN EXAMPLE	302, 340, 341, 342, 348	YSECEN SUBPROGRAM	428
VARIABLE KEYWORD	98	YSECEP SUBPROGRAM	428
VECCON SUBPROGRAM	426	YTRANS SUBPROGRAM	428
VECMAG SUBPROGRAM	426	ZCT KEYWORD	
VIB KEYWORD		DEFINITION	195
DEFINITION	160		

TESTRUN EXAMPLE 285, 289, 290, 292, 295, 297, 379		ZOCPAR SUBPROGRAM	428
Zero of energy	74	ZOCPRN SUBPROGRAM	428
Zero-curvature Tunneling	12, 41	ZOCUPD SUBPROGRAM	428
ZEROPT SUBPROGRAM	428	ZOCVCL SUBPROGRAM	428
ZERP SUBPROGRAM	428	ZRIDDR SUBPROGRAM	428
ZIGAMA SUBPROGRAM	428	ZROUT SUBPROGRAM	429
ZOC3P SUBPROGRAM	428	ZRPTE SUBPROGRAM	429
ZOCAST SUBPROGRAM	428	ZSPMEP SUBPROGRAM	429
ZOCFRE SUBPROGRAM	428	μ OMT	58
		μ VT	34