Polyrate 17-C Manual

Jingjing Zheng, Junwei Lucas Bao, Rubén Meana-Pañeda, Shuxia Zhang,
Benjamin J. Lynch, 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, Vasilios S. Melissas, Jordi Villà,
Ivan Rossi, Elena. L. Coitiño, Jingzhi Pu, Titus V. Albu
*Department of Chemistry, Chemical Theory Center, and Minnesota Supercomputing
Institute, University of Minnesota, Minneapolis, Minnesota 55455*

Artur Ratkiewicz
Institute of Chemistry, University of Bialystok, Poland

Rozeanne Steckler
*Northwest Alliance for Computational Science & Engineering,
Oregon State University, Corvallis, Oregon 97331*

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, Chemical Theory Center, and Minnesota Supercomputing
Institute, University of Minnesota, Minneapolis, Minnesota 55455*

Most recent changes to program:
    June 16, 2018 for the part excluding the SS-QRRK utility;
    Dec. 22, 2018 for SS-QRRK utility.
Most recent update to manual: Jan 12, 2022

Polyrate 17-C is available from the **Truhlar Research Group Software** site at
    https://comp.chem.umn.edu/polyrate/
Polyrate 17-C is being submitted to the **Computer Physics Communications Program
Library**, and after acceptance it will also be available from the CPC Program Library at
    https://data.mendeley.com/journal/00104655?
The code is made available free of charge.

Executive summary:

Polyrate 17-C is a suite of computer programs for the calculation of chemical reaction rates of polyatomic species (including atoms and diatoms as special cases) by variational transition state theory (VTST); conventional transition state theory is also supported. *Polyrate 17-C* (called *Polyrate* for short) consists of the main program, called *Polyrate*, and a set of utility codes,

Polyrate can calculate the rate constants for both bimolecular reactions and unimolecular reactions, and it can be applied to reactions in the gas-phase, liquid-solution phase, or solid-state and to gas-solid-interface reactions. *Polyrate* can perform variational transition state theory (VTST) calculations on gas-phase reactions with both tight and loose transition states. For tight transition states it uses the ***reaction-path (RP) variational transition state theory*** developed by Garrett and Truhlar, and for loose transition states it uses ***variable-reaction-coordinate (VRC) variational transition state theory*** developed by Georgievskii and Klippenstein.

The ***RP methods*** used for tight transition states are conventional transition state theory, canonical variational transition state theory (CVT), and microcanonical variational transition state theory (μVT) with multidimensional semiclassical approximations for tunneling and nonclassical reflection. The tunneling approximations available are zero-curvature tunneling (ZCT), small-curvature tunneling (SSCT), large-curvature- tunneling (LCT), and optimized multidimensional tunneling (OMT). The SCT option is the centrifugal dominant semiclassical adiabatic ground-state tunneling, and the LCT options include both LC3 and LC4 including tunneling into excited states. One may also treat specific vibrational states of selected modes with translational, rotational, and other vibrational modes treated thermally.

For ***RP calculations***, several options are available for reaction paths, vibrations transverse to the reaction path, and transition state dividing surfaces. Generalized-transition-state dividing surfaces may be defined on the basis of gradient directions in isoinertial coordinates or by the re-orientation of the dividing surface algorithm. Reaction paths may be calculated by the Euler steepest-descent, Euler stabilization, Page-McIver, or variational-reaction-path algorithms. Vibrations away from the reaction path may be defined by rectilinear (not recommended), nonredundant curvilinear, or redundant curvilinear coordinates. Vibrational frequencies may be unscaled (not recommended) or scaled.

For ***VRC calculations***, rate constants may be calculated for canonical or microcanonical ensembles or energy-and-total-angular-momentum resolved microcanonical ensembles. VRC calculations are most appropriate for barrierless association reactions. Both single-faceted and multifaceted dividing surfaces are supported.

Pressure-dependent rate constants for elementary reactions can be computed using system-specific quantum RRK theory (SS-QRRK) with the information obtained from high-pressure-limit VTST calculation as input by using the *SS-QRRK* utility code. The *SS-QRRK* utility program is part of the *Polyrate* distribution, and its usage is described in a separate manual.

Potential energy surfaces may be analytic functions evaluated by subroutines, or they may be implicit surfaces defined by electronic structure input files or interface subroutines containing energies, gradients, and force constants (Hessians). 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.

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 112 ***test runs***, and 46 of these are for direct dynamics calculations; 85 of the test runs are single-level runs, and 27 are dual-level calculations.

Polyrate is designed to be used in conjunction with i***nterfaces to electronic structure programs*** for direct dynamics. Currently the following interfaces are available (note that some of the interfaces were created with older versions of *Polyrate*, but they should work with *Polyrate* as well):

Electronic structure package	Interface
CHARMM with CHARMMRATE	CRATE
GAMESS (or with GAMESSPLUS)	GAMESSPLUSRATE
GAUSSIAN 16/GAUSSIAN 09	GAUSSRATE
JAGUAR	JAGUARATE
MC-TINKER	MC-TINKERATE
MORATE	MOPAC
MULTILEVEL	MULTILEVELRATE
NWCHEM	NWCHEMRATE

CONTENTS

Part I General Information

1. INTRODUCTION

Polyrate 17-C (to be called *Polyrate* for short) is a suite of programs 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 program suite consists of the main program, called POLYRATE, and a number of utility programs. The program performs two qualitatively different kinds of VTST calculations:

- RP-VTST: reaction–path VTST for arbitrary kinds of reactions with tight transition states.
- VRC-VTST: variable-reaction-coordinate VTST only for loose transition states of bimolecular association reactions or their reverse dissociation reactions.

and it also can calculate pressure-dependent rate constants by using system-specific quantum Ramsperger–Rice–Kassel theory:

- SS-QRRK: system-specific quantum RRK theory

RP-VTST and VRC-VTST are included in *Polyrate* proper, and SS-QRRK is performed in a separate utility code that is included as part of the *Polyrate* package. The *SS-QRRK* utility code is described in a separate manual, which is also included as part of the *Polyrate* package.

The following articles contain summaries of the technical background for the methods in *Polyrate*:

- "Variational Transition-State Theory," D. G. Truhlar and B. C. Garrett, Accounts of Chemical Research 13, 440-448 (1980). doi.org/10.1021/ar50156a00220.
- "Generalized Transition State Theory," D. G. Truhlar, A. D. Isaacson, and B. C. Garrett, in *Theory of Chemical Reaction Dynamics*, edited by M. Baer (CRC Press, Boca Raton, FL, 1985), Vol. 4, pp. 65–137. comp.chem.umn.edu/Truhlar/docs/C20.pdf
- "Variational Transition State Theory," D. G. Truhlar and B. C. Garrett, in *Annual Review of Physical Chemistry*, Vol. 35, edited by B. S. Rabinovitch, J. M. Schurr, and H. L. Strauss (Annual Reviews, Inc., Palo Alto, California, 1984), pp. 159–189. doi.org/10.1146/annurev.pc.35.100184.001111 comp.chem.umn.edu/Truhlar/docs/C21.pdf

- "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, Computer Physics Communications **47**, 91-102 (1987).
doi.org/10.1016/0010-4655(87)90069-5

- "Dynamical Formulation of Transition State Theory: Variational Transition States and Semiclassical Tunneling," S. C. Tucker and D. G. Truhlar, in *New Theoretical Concepts for Understanding Organic Reactions*, edited by J. Bertrán and I. G. Csizmadia (Kluwer, Dordrecht, The Netherlands, 1989), pp. 291–346. [NATO ASI Ser. C **267**, 291–346 (1989)]
doi.org/10.1007/978-94-009-2313-3_11
comp.chem.umn.edu/Truhlar/docs/UMSI1988-90.pdf

- "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. C. Hancock, J. G. Lauderdale, T. Joseph, and D. G. Truhlar, Computer Physics Communications **71**, 235-262 (1992).
doi.org/10.1016/0010-4655(92)90012-N

- "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**, 341-343 (1995). [new version announcement] doi.org/10.1016/0010-4655(95)00037-G

- "Transition State Theory," B. C. Garrett and D. G. Truhlar, in *Encyclopedia of Computational Chemistry*, edited by P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, and H. F. Schaefer III (John Wiley & Sons, Chichester, UK, 1998), Volume 5, pp. 3094-3104.
comp.chem.umn.edu/Truhlar/docs/C58.pdf

- "Variational Transition State Theory and Multidimensional Tunneling for Simple and Complex Reactions in the Gas Phase, Solids, Liquids, and Enzymes," D. G. Truhlar, in *Isotope Effects in Chemistry and Biology*, edited by A. Kohen and H.-H. Limbach (Marcel Dekker, Inc., New York, 2006), pp. 579-619.
doi.org/10.1201/9781420028027.ch22

- "Variational Transition State Theory in the Treatment of Hydrogen Transfer Reactions, " D. G. Truhlar and B. C. Garrett, in *Hydrogen-Transfer Reactions*, edited by J. T. Hynes, J. P. Klinman, H.-H. Limbach, and R. L. Schowen (Wiley-VCH, Weinheim, Germany, 2007), Vol. 2, pp. 833-874.
doi.org/10.1002/9783527611546.ch27
comp.chem.umn.edu/Truhlar/docs/C73.pdf

- "Variational Transition State Theory," B. C. Garrett and D. G. Truhlar, in *Theory and Applications of Computational Chemistry: The First Forty Year*s, edited by C. E. Dykstra, G. Frenking, K. S. Kim, and G. E. Scuseria (Elsevier, Amsterdam, 2005), pp. 67-87. CODEN: 69HSJX
  doi.org/10.1016/B978-044451719-7/50048-2
  comp.chem.umn.edu/Truhlar/docs/C71.pdf

- "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*, Vol. 23, edited by K. B. Lipkowitz and T. R. Cundari (Wiley-VCH, Hoboken, NJ, 2007), pp. 125-232.
  doi.org/10.1002/9780470116449.ch3
  comp.chem.umn.edu/Truhlar/docs/C76.pdf

- "Mechanistic Analysis of the Base-Catalyzed HF Elimination from 4-Fluoro-4-(4'-nitrophenyl)butane-2-one Based on Liquid-Phase Kinetic Isotope Effects Calculated by Dynamics Modeling with Multidimensional Tunneling," Y. Kim, A. V. Marenich, J. Zheng, K. H. Kim, M. Kołodziejska-Huben, M. Rostkowski, P. Paneth, and D. G. Truhlar, Journal of Chemical Theory and Computation **5**, 59-67 (2009).
  doi.org/10.1021/ct800345j

- "Multi-path Variational Transition State Theory for Chemical Reaction Rates of Complex Polyatomic Species: Ethanol + OH Reactions," J. Zheng and D. G. Truhlar, Faraday Discussions **157**, 59-88 (2012).
  doi.org/10.1039/C2FD20012K.

- "Association of CF_2 and Dissociation of C_2F_4 by Variational Transition State Theory and System-Specific Quantum RRK Theory," J. L. Bao, X. Zhang, and D. G. Truhlar, Proceedings of the National Academy of Sciences U.S.A. **113**, 13606-13611 (2016).
  doi.org/10.1073/pnas.1616208113

- "Variational Transition State Theory: Theoretical Framework and Recent Developments," J. L. Bao and D. G. Truhlar, Chemical Society Reviews **46**, 7548-7596 (2017).
  doi.org/10.1039/C7CS00602K

- "Semiclassical Multidimensional Tunneling Calculations," D. G. Truhlar, in *Tunnelling in Molecules: Nuclear Quantum Effects from Bio to Physical Chemistry*, edited by J. Kaestner and S. Kozuch (RSC Publishing, Cambridge, 2021), pp. 261-282.
  doi.org/10.1039/9781839160370-00261
  https://truhlar.dl.umn.edu/sites/truhlar.dl.umn.edu/files/c88corrected_proofs.pdf

In *Polyrate*, analysis of reactants and products can be performed for up to two reactants and two products for RP-VTST and for two reactants and one product for VRC-VTST. *Polyrate* also calculates equilibrium constants and reverse reaction rates, so the VRC-VTST method can also calculate rate constants for one reactant and two products.

The RP-VTST method can be applied to gas-phase, liquid-phase, solid-phase, and gas-solid-interface reactions; the VRC-VTST method is used for gas-phase reactions.

Generalized transition state theory calculations may be carried out using the harmonic approximation for vibrations with either rectilinear or curvilinear coordinates in RP-VTST calculations. It is highly recommended to use curvilinear coordinates (via the "CURV2" or "CURV3" options) rather than Cartesian coordinates for generalized-normal mode analysis along the MEP in RP-VTST calculations; Cartesian coordinates will often yield unphysical imaginary frequencies along the reaction coordinate, and these will cause unphysical VTST results. One should always check in the long output that the generalized normal mode frequencies as functions of the reaction coordinate do not contain any imaginary frequencies transverse to the reaction path; imaginary frequencies may lead to unreliable or wrong results.

A limited amount of nontorsional anharmonicity may also be included, but a very important feature of *Polyrate* is its ability to handle multistructural and torsional-potential anharmonicity by utilizing the Polyrate in conjunction with the MSTOR program. See Section 10.O.

The RP-VTST rate constants can be computed using canonical variational theory (CVT), and microcanonical variational theory (μVT). Conventional transition state theory (TST) may also be employed. VRC-VTST calculations currently only can be applied to gas-phase reactions using CVT, μVT, and total energy and total angular momentum-resolved microcanonical variational theory, which for short may be called *E,J*-resolved microcanonical variational theory (E,J-μVT).

For thermal or ground-state reactions using RP-VTST, 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 (μOMT 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 or 4 (LCG3 or LCG4. 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 (μOMT) in which, at each energy, the tunneling probability is

approximated as the larger of the values calculated by the SCT and LCT methods. The option of carrying out a tunneling calculation is not available in conjunction with VRC-VTST option.

The *Polyrate* program computes rate constants by either single-level or dual-level approaches for RP-VTST calculations and by single-level approach for VRC-VTST calculations.

The single-level calculations and the lower level of dual-level calculations using RP-VTST 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. VRC-VTST calculations can only be used for single-level calculations based on a global PES or direct dynamics interfaced with an electronic structure package.

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.). Note that VRC-VTST calculations don't need gradient information if optimized geometries and vibrational frequencies of reactants and/or product are provided in the input file. 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 or frequencies for the reactants, products, and saddle point, and optionally these quantities and the gradient at a sequence of points on the minimum energy path. These data are then used for dynamics calculations.

For RP-VTST calculations using dual-level approach, the data along the reaction path may be "corrected" using higher-level data supplied by the user at selected points either for reaction paths based on potential energy surfaces or for those based on electronic structure input files. A generic name for such dual-level RP-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. This kind of calculation is called a double-slash (//) calculation. 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 for RP-VTST calculations. 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, since version 8.0 of *Polyrate* we have allowed 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.

Interface versions of *POLYRATE*, called CHARMMRATE, GAMESSPLUSRATE, GAUSSRATE, JAGUARATE, MC-TINKERATE, MORATE, MULTILEVELRATE, and NWCHEMRATE, which compute rate constants from potentials, gradients, and Hessians calculated by electronic structure methods, are also available

CHARMMRATE is a module of the CHARMM program; CHARMMRATE can compute enzyme-catalyzed rates by ensemble-averaged VTST. 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 GAMESSPLUS (or GAMESS) electronic structure packages based on *Polyrate* version 9.3, called GAMESSPLUSRATE, is available as a separate distribution.

An interface to the GAUSSIAN16 (or GAUSSIAN98 or GAUSSIAN03 or GAUSSIAN09) electronic structure packages based on *Polyrate* version 2017, called GAUSSRATE, is available as a separate distribution.

An interface to the JAGUAR package based on *Polyrate* version 9.6, called JAGUARATE, is available as a separate distribution.

An interface to the MC-TINKER package based on *Polyrate* version 9.1, called MC-TINKERATE, is available as a separate distribution.

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 University of Minnesota.

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

An interface to the NWCHEM package based on *Polyrate* version 9.7, called NWCHEMRATE, 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. Therefore we do not mention AMSOLRATE in most lists of available interface programs.

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 (RP-VTST).
- 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 (RP-VTST).
- 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 (RP-VTST).
- 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 (RP-VTST and VRC-VTST).
- 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 (RP-VTST).
- 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 (RP-VTST).

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.

Note that RP-VTST calculations can be performed using any of these options, but VRC-VTST calculations can only be performed using option 1.A. 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 for RP-VTST calculations 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 μVT μVT/ZCT μVT/SCT μVT/LCT		TST/ZCT CVT CVT/ZCT CVT/SCT μVT μVT/ZCT μVT/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	μVT μVT/ZCT μVT/SCT μVT/LCT		μVT μVT/ZCT μVT/SCT	
	hooks plus fu51		**fu30, fu31, or fu40 plus fu51**	
VTST-ISPE	μVT μVT/ZCT μVT/SCT		μVT μVT/ZCT μVT/SCT	

Note that whenever both SCT and LCT methods are available, then the μOMT 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 the hooks can be used to interface *Polyrate* with electronic structure codes for direct dynamics; examples are the CRATE, GAMESSPLUSRATE, GAUSSRATE, JAGUARATE, MC-TINKERATE, MORATE, MULTILEVELRATE, and NWCHMERATE programs, which are interfaces to CHARMM with the CHARMMRATE module, GAMESSPLUS, GAUSSIAN, JAGUAR, MC-TINKER, MOPAC, MULTILEVEL, and NWCHEM, respectively.

2. RECOMMENDED CITATION FOR *POLYRATE 17-C*

The recommended CITATION for the *Polyrate* code is

J. Zheng, J. L. Bao, R. Meana-Pañeda, S. Zhang, B. J. Lynch, 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, V. S. Melissas, J. Villà, I. Rossi, E. L. Coitiño, J. Pu, T. V. Albu, A. Ratkiewicz, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar, *Polyrate 17-C*, University of Minnesota, Minneapolis, 2018. https://comp.chem.umn.edu/polyrate/

References for methods used in the code are given in Sect. 21 of this manual.

3. RECOMMENDED CITATIONS 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). (used with permission)

2. The EF code for stationary point optimization is from J. Barker, 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). (used with permission)

4. LICENSING

Polyrate 17-C is licensed under the [Apache License, Version 2.0](). The manual of *Polyrate 17-C* is licensed under [CC-BY-4.0](). Publications of results obtained with the *Polyrate 17-C* software should cite the program and/or the article describing the program.

No guarantee is made that this software is bug-free or suitable for specific applications, and no liability is accepted for any limitations in the mathematical methods and algorithms used within. No consulting or maintenance services are guaranteed or implied.

The use of the *Polyrate 17-C* implies acceptance of the terms of the licenses.

5. PROGRAM AND MANUAL DESCRIPTION

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

The *Polyrate* program is written in FORTRAN 90 with the module and dynamically allocated memory features.

Users should be aware that, beginning with version 2008, *Polyrate* can perform two kinds of calculations. One is RP-VTST, and the other one is VRC-VTST, as explained in Section 1. These two kinds of calculations involve different subroutines to generate different executable programs. RP-VTST has both serial and MPI parallel versions and VRC-VTST has only an MPI parallel version and requires one to link to the SPRNG library. Starting with version 2010, the SPRNG package is included in the *Polyrate* distribution for the user's convenience, and the *Polyrate* installation script includes compiling the SPRGN library before linking *Polyrate* subroutines to this library. Therefore installation of SPRNG library and *Polyrate* program can be done with one script. To use RP-VTST, no pre-installed library or package is required.

The SPRNG package can be also obtained separately from the web page:
http://sprng.cs.fsu.edu/

5.B. Description of the manual

This manual is divided into five parts.

Part I has five sections and includes general information about *POLYRATE*.

Part II has five sections and includes theoretical background and computational details for RP-VTST. Section 6 provides an overview of 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 (Section 8) or direct dynamics calculations, which use an electronic structure input file (Section 9). Section 10 provides selected details of the calculations, especially for procedures that are not fully described in publications.

Part III has one section, Section 11, to describe the theoretical background of VRC-VTST.

Part IV has 3 sections for description of input files. Section 12 describes the complete *Polyrate* input and selected aspects of the output for RP-VTST calculations. Section 13 describes the input file for VRC-VTST. Section 14 describes files that give extra details of LCT calculations.

Part V gives more general information about the *Polyrate* program package. Section 15 contains instructions on how to install, test, and use the *Polyrate* program. Section 16 gives sample runs (the *Polyrate* test suite) that may be useful in understanding the input or testing the program. Section 17 lists the files in the program distribution, and it has a brief explanation of each subprogram. Section 18 contains instructions on using the Unix scripts that are provided. Section 19 describes file usage. Section 20 provides references indicated by brackets in the text of the manual, 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. 71 in the bibliography (Sect. 21), and should be consulted along with this manual. "Generalized Transition State Theory" (GTST), which is Ref. 5 in Sect. 21, is also a useful chapter, especially for an introduction to the older core ideas, although Ref. 71 supersedes it. If these books are unavailable in your library, you may obtain a copy of the chapters from the senior author (DGT) at

http://comp.chem.umn.edu/*Polyrate*

A PDF file with a complete compilation of all known errors in Ref. 5 is also given at this Web site.

5.B.1. Referencing conventions

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

5.B.2. Key references

Reference 5 in Sect. 21 is sometimes referred to simply as "the GTST book chapter":

"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.

Reference 71 in Sect. 21 is sometimes referred to as "the VTST/MT book chapter":

"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*, Vol. 23, edited by K. B. Lipkowitz and T. R. Cundari (Wiley-VCH, Hoboken, NJ, 2007), pp. 125-232.

These two book chapters explain the methods used in *Polyrate* up to 2005. PDF files of these chapters are available at the *Polyrate* Web site:

http://comp.chem.umn.edu/polyrate

All known errata to the GTST book chapter have been assembled into a PDF file. This file is also available at the *Polyrate* Web site.

A review that also covers the more recently developed methods is also available: Ref. 114 in Sect. 21:

"Variational Transition State Theory: Theoretical Framework and Recent Developments," J. L. Bao and D. G. Truhlar, Chemical Society Reviews **46**, 7548-7596 (2017). doi.org/10.1039/C7CS00602K

Taken together, these three articles explain almost all the methods used in *Polyrate*.

Part II Theory Background and Calculation Details on RP-VTST

6. THEORETICAL BACKGROUND ON RP-VTST

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 Sect. 21, especially the VTST/MT book chapter (Ref. 70 of Sect. 21) and the most recent review (Ref. 114 of Sect 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). Microcanonical variational theory (μVT) 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 Sect. 21.

The improved canonical variational theory (ICVT) option has been removed from *Polyrate*. If one sees a need for the more accurate threshold treatment of ICVT, one should instead use μVT because μVT is coded in *Polyrate* with the extended Beyer-Schwinehart (BS) algorithmfor calculating number of available state, and this iss more efficient and more complete than IVCT.

The conventional transition-state-theory canonical rate constant involves a symmetry number [Ref. 1, Sect. 22] 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. 22] except for the treatment of loose transition state using variable-reaction-coordinate (VRC) algorithm with single-faceted or multifaceted (MF) dividing surface (the VRC algorithm and MF dividing surfaces are discussed in detail in Section 11). The descriptions here are all referred to MEP. Following Fukui [Ref. 5, Sect. 22], 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. 22]. 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 $k_f^{GT}(T,s)$ 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:

$V_{MEP}(s)$ —   potential along the minimum energy path,

$|\mathbf{I}(s)|$ —   determinant of the moment of inertia tensor,

$X(s)$ — 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. 22] 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, ..., F-1, as the curvature vector $\boldsymbol{B}_F$.

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 the usual spectroscopists' mass-weighted coordinates (which have units of mass 1/2 length) if the mass weighting is amu$^{1/2}$.

Therefore setting μ equal to 1 amu procedure is especially recommended if the goal is to make the treatment transparent to those who are only used to the spectroscopists' mass weighting scheme.

The reverse rate constant is obtained from the forward one by detailed balance. The canonical variational theory [Ref. 6, Sect. 22] forward rate constant is obtained by minimizing the generalized transition state theory rate constant with respect to s:

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

(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 12.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 12.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	*atomic*	0	0	…	…	…	N.A.
linear reactant or product	*linrp*	$3N-5$	2	3	2	0	optional
nonlinear reactant or product	*nonlinrp*	$3N-6$	3	3	3	0	optional
solid-state reactant or product	*ssrp*	$3N$	0	0	0	0	prohibited
linear transition state	*lints*	$3N-6$	2	3	2	0	optional
nonlinear transition state	*nonlints*	$3N-7$	3	3	3	0	optional
solid-state transition state	*ssts*	$3N-1$	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 12.A.6) cannot 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 is calculated in the same manner as for a conventional transition state (which is a saddle point, also called a transition structure). 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. 22]. 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. 22].

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. The torsional potential anharmonicity along reaction path can be included using the formula of the multi-structural torsional method but only with one input structure. 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. 22 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. 22 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. 22].

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. 22], the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [Ref. 15, Sect. 22], or the Eigenvector Following (EF) method [Refs. 54-58, Sect. 22] to search over a restricted set of atomic Cartesians 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. 22] (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. 22].

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 $\mathbf{L}_m$ 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:

Stationary point: $G = R + T = $ 6 for nonlinear molecule,

5 for linear molecule,

0 for gas-solid interface.

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. (It is very important to use curvilinear coordinates rather than Cartesian coordinates for VTST calculations; Cartesian coordinate will very likely yield imaginary frequencies along the reaction coordinate, which will cause unphysical VTST results.)

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_{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. (It is very important to use curvilinear coordinates rather than Cartesian coordinates for VTST calculations; Cartesian coordinate will very likely yield imaginary frequencies along the reaction coordinate, which will cause unphysical VTST results.) 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:

$$V_a^{\mathrm{G}}\left(s=0\right)=V_a\left(n=0,s=0\right)=V_{\mathrm{MEP}}(s=0)+ZPE(s=0)$$

(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 isoinertial coordinate system. [Ref. 49, Sect. 22] 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

		nmode) or with the cubic starting algorithm based on this mode (if FIRSTSTEP is *cubic*).
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. 22] 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. 22; 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. 22] and Schmidt et al. [Ref. 20, Sect. 22]. It is called ES1 (Euler stabilization method, version 1), and it is described elsewhere [Ref. 18, Sect. 22]. 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. 22], 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. 22] 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 $k^{GT}(T, s)$, 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. (It is very important to use curvilinear coordinates rather than Cartesian coordinates for VTST calculations; Cartesian coordinate will very likely yield imaginary frequencies along the reaction coordinate, which will cause unphysical VTST results.)

The algorithm used for rectilinear coordinate calculations is discussed by Miller, Handy, and Adams [Ref. 7, Sect. 22]. 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

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

(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. 22] 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.

Starting with *Polyrate* version 2008, VRC option is recommended instead of NOSADDLE option for barrierless association reactions. The VRC option uses a variable reaction coordinate and single-faceted or multifaceted dividing surfaces. See Sections 11 and 13 for details.

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}, $m = 1, 2, ..., F-1$, 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. 22]. 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 is 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. 22], 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 μVT, 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 μVT, 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 μVT calculations), although it is often close to unity. The CAG correction was introduced in Refs. 10 and 44 of Sect. 22, 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. 22], and they are illustrated (in color) in an earlier review article [Ref. 46 in Sect. 22].

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 (μOMT) 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 (μOMT).

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 $\Delta G_{\mathrm{f}}^{\mathrm{GT},0}(T,s)$. 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 (2*N*+1)-point results.

The CD-SCSAG and LCG3 or LCG4 transmission probabilities are also used to calculate optimized multidimensional tunneling approximations called COMT and μOMT, 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 CVT rate constant with an MEPSAG tunneling correction yields an CVT/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 Sect. 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. μOMT method

The microcanonical optimized multidimensional tunneling (μOMT) temperature-dependent transmission coefficient is the ratio of the Boltzmann average of the μOMT quantum transmission probability to the Boltzmann average of the classical transmission probability, where the μOMT 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. 22.

6.O. Interpolated VTST and mapped interpolation

Calculating $V_{\mathrm{MEP}}(s)$, $|\mathbf{I}(s)|$, 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. 22 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.O.2 and 6.O.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, μVT, 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

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

where $V_{MEP}(s)$ is represented by an Eckart function and $ZPE(s)$ 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 $V_{MEP}(s)$ 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, $\Delta G_f^{GT,0}(T,s)$ is calculated using the interpolated $V_{MEP}(s)$, $|\mathbf{I}(s)|$, and $\omega(s)$. The SCT transmission coefficients can be obtained for first-order IVTST using the interpolated $\mu_{eff}(s)$.

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 12.F). In the current version, this method is available only for bimolecular reactions with two

products, and therefore the keywords MEPTYPER and MEPTYPEP for the unit fu50 input should both be set to 2. In addition, the keywords MEPTYPER 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 a 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 for 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 $V_{MEP}(s)$, $|\mathbf{I}|(s)$, $\omega(s)$, 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 if either forward or reverse reaction is bimolecular. The parameter z is determined as

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

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

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

The method employs different approaches for the interpolation of $V_{MEP}(s)$, 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 cubic polynomial can be function of either s or z. The user should base the choice of s or z on experience or should test the two choices (using function of s or z) to see which one gives reasonable potential energy curve. The user can also choose not to use the polynomial to generate the 10 extra energies; however, the default is to generate them by using a cubic polynomial function of z. 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 for a bimolecular reaction or a reaction with two products, 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 ENDSLP 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 $s =$ SLM on the reactant side to $s =$ 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 EXNSTEP * EXSTEP points on each side of the grids. Furthermore SLM is replaced by SLM – EXNSTEP * EXSTEP, and SLP is replaced by SLP + EXNSTEP * EXSTEP. As should be clear from Section 12.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 $V_{MEP}(s)$ 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 were 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 12.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 12.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 the rest of Section 8). In the present version of the code, user can also turn on the VRC option, which is off by default, to calculate rate for barrierless association reactions using a variable reaction coordinate and multifaceted dividing surfaces. When VRC is turned on, potential energies

can only be calculated through hooks since the Monte Carlo method is used for the variable-reaction-coordinate calculations with single-faceted or multifaceted dividing surfaces. Gradient and Hessian are not required for all Monte Carlo samples.

DL	IVTST0	IVTST1	POTENTIAL	Action
None	off	Off	*hooks*	Single-level calculation based on whole reaction path calculated from an analytic PES.
none	off	Off	*unit30*, *unit31*, or *unit40*	Single-level direct dynamics based on whole reaction path calculated from an electronic structure input file on unit fu30, fu31, or fu40
none	on	Off	*unit29* or *unit50*	Single-level direct dynamics IVTST-0 calculation based on electronic structure input data from unit fu29 or fu50
none	off	On	*unit29*	Single-level direct dynamics IVTST-1 calculation based on electronic structure input data from unit fu29.
ic	off	Off	*hooks*	Dual-level direct dynamics with lower-level data from analytic PES and higher-level data from unit fu50
ic	off	Off	*unit30*, *unit31*, or *unit40*	Dual-level direct dynamics with lower-level data from unit fu30, fu31, or fu40 and higher-level data from unit fu50
ispe	off	Off	*hooks*	Dual-level direct dynamics with lower-level data from analytic PES and higher-level data from unit fu51
ispe	off	Off	*unit30, unit31,* or *unit40*	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, if using he RP-VTST option without NOSADDLE, the program will determine the saddle point geometry and properties for the tight transition state. 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. For 0 saddle point, user can choose NOSADDLE option, which 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. However, rather than using NOSADDLE, the VRC option in GENERAL section is recommended for barrierless association reactions, and this option does not require to input any geometry for transition state.

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 angstroms instead of angstroms. Thus if REDM, specified by the SCALEMASS keyword which is described in Subsection 12.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 file aamod.f90 (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.

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: when using fu30, fu31, or fu40, if the positive direction of *s* in the output is not associated with the direction toward products, then change the choice for the SIGN variable in the PATH section of the input and repeat the run.

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 12.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. Note that SLM and SLP keywords are only for MEP, and invalid for variable reaction coordinate.

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   = $3 \times$ 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
> (It is the actual number of atoms in the system set in the input file.)

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 19. 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 aamod.f90 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 17.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 12.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 array POTINF(NPOTPT) in rate_const module 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*.

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:

$$
\begin{array}{ll}
X(1) - X(3): & X, Y, Z \text{ for atom A} \\
X(4) - X(6): & X, Y, Z \text{ for atom B} \\
X(7) - X(9): & X, Y, Z \text{ for atom C} \\
R(1): & R(A-B) \\
R(2): & R(B-C) \\
R(3): & R(C-A)
\end{array}
$$

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 module POTCM3:

```
module potcm3
   integer :: nsurf, nder, ndum(8)
```

```
    real(8) :: r(3),energy, dedr(3)
  end module potcm3
```

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):	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
DX(10):	DEDX for atom D
DX(11):	DEDY for atom D
DX(12):	DEDZ 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)
use potcm3
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 reaction path 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 MORATE, GAUSSRATE, GAMESSPLUSRATE, JAGUARATE, or NWCHEMRATE) 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., MORATE, GAUSSRATE,

GAMESSPLUSRATE, JAGUARATE, and NWCHEMRATE 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., MORATE, GAUSSRATE, GAMESSPLUSRATE, JAGUARATE, and NWCHEMRATE. 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 19 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 12.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 IGPOT variable. If the keyword option *bfgs* or *nr* is used for either of these keywords, IGPOT is 0, and the optimization is performed by NEWT, which is a *Polyrate* subroutine. The OHOOK option (IGPOT = *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 IGPOT 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 12.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 12.C, a description of the input when using unit fu31 is given in Section 12.D., and a description of the input when using unit fu40 is provided in Section 12.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

$V_{\mathrm{MEP}}(s)$ 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 $\nabla V_{\mathrm{MEP}}(s)$ 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 Sect. 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-orients 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 GAUSSIAN 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 GAUSSIAN 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 GAUSSIAN 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 GAUSSIAN calls the "standard orientation". However, notice that the forces and the force constants matrix printed out in the GAUSSIAN *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 GAUSSIAN 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 δs = SSTEP and (*ii*) the save grid, where frequencies and the Hessian are calculated and stored, with step size Δs = INH * SSTEP. See also sections 7, 10.D, and 12.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 Sect. 21. When using file fu30, this is the same as the gradient grid if LOPT(6) $\leq$ 2; it is a subset of the gradient grid if LOPT(6) is 3 or $\geq$ 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 Δs = INH * SSTEP. This interpolation can also be done by the IVTST-M method (For unit fu30, 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 24.A, item 15], $d\{[sign(s)]v_{i\gamma}(s)\}/ds$ becomes $[sign(s)]dv_{i\gamma}(s)/ds$. 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 *sign(s)* function

must be implemented before the fit is carried out because $\left[sign(s)\right]v_{i\gamma}(s)$ 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 IVTST0FREQ option. In version 2010, 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 12.A.7 and in Ref. 48 of Sect. 21.

For reactions without a saddle point and using RP-VTST *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 $\mu_{eff}(s)$ 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 $\mu_{eff}(s)$. [A similar extrapolation is also made for $V_{MEP}(s)$ 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 = SLP) or negative (s = 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

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

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 SSTOR, 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 except the SSTOR option. 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 rectillinear 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. 22] (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:

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

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

where $\varepsilon_{0,m}^{\mathrm{WKB}}$ is the WKB ground-state energy of mode m, and $Q_{\mathrm{vib},m}^{\mathrm{HO}}$ 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 treatments

Torsions may be treated multi-conformer methods as explained in Refs. 53 and 63 or by multi-structural methods that couple the torsions as explained in Refs. 89, 90, 92–96, 98–

105, and 113 of Sect. 21. These references explain the methods in detail, and they also give original references. Note that we now recommend using MS-VTST or MP-VTST rather than using the multi-conformer methods.

Beginning with version 8.2 of *Polyrate*, a torsional mode can be treated in the full generality of Ref. 53. Please note the errata that have been added to Ref. 53. We label the method developed in Ref. 53 as the Chuang-Truhlar (CT) method.

Beginning with version 2010 of *Polyrate*, several methods in Ref. 63 are also implemented. These additional methods are the reference Pitzer-Gwinn (RPG) method, the segmented- reference Pitzer-Gwinn (SRPG) method, the Ayala-Schlegel (AS) method, and the segmented Ayala-Schlegel (SAS) method.

Beginning with version 2016, *Polyrate* can treat the torsional anharmonicity along reaction path (saddle point and nonstationary points) by using coupled torsional potentials in the multistructural method including torsional anharmonicity (MS-T(C)). Note that only a single structure (SS) version is taken into account since only one saddle point is given in a single *Polyrate* input file. Therefore we call this the SS-T method. Multistructural (MS) effects should be taken into account separately by using the MSTOR program.

There are two kinds of hindered rotor treatments: multiconformer (MC) methods and multistructure (MS) methods. The MC methods involve identifying one or more of the normal modes (or generalized normal modes) as a torsion about of specified bond. When frequencies are required in MC methods they are normal-mode frequencies. The MS methods do not require associating specific torsion with specific normal modes. When torsional frequencies are required in MS methods, they are calculated in internal coordinates based on a coupled model potential. In *Polyrate* calculations, only one structure of the transition state or one reaction path is taken into account; therefore only single-structure torsional potential anharmonicity (SS-T) along the reaction path is taken into account.

10.B.3.a. Multiconformer methods

10.B.3.a.i. CT method

If mode m is treated as a hindered rotor (HR) by the CT method, 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 the interpolation is designed so that it remains reasonably accurate for the hindered-rotor regime between these limiting cases. The CT method is:

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

where the multiconformer harmonic oscillator (MC-HO) approximation for mode m is $Q_m^{MC\text{-}HO}$, the free rotor approximation for mode m is Q_m^{FR}, and Q_m^{I} is an intermediate result. They are given by

$$Q_{tor}^{MC-HO} = \sum_{j=1}^{P} \frac{e^{-\beta(U_j + \hbar\omega_{tor,j}/2)}}{1 - e^{-\beta\hbar\omega_{tor,j}}}$$

$$Q^{FR} = \frac{(2\pi IkT)^{1/2}}{\hbar\sigma}$$

$$Q^{I} = \sum_{j=1}^{P} \frac{e^{-\beta U_j}}{\hbar\beta\omega_{tor,j}}$$

where P is number of distinguishable minima along the torsion coordinate, β is $(k_B T)^{-1}$, k_B is Boltzmann constant, U_j is the energy of the jth well relative to the global minimum, $\omega_{tor,j}$ is the torsional frequency of the jth well along the normal-mode or generalized-normal-mode coordinate, I is the averaged torsional moment of inertia of the P minima, σ is the total symmetry number ($\sigma = M/P$, M is the total number of minima along the torsion coordinate in the range $[0, 2\pi)$). Note that usually $P = 1$ or $P = M$. Therefore, if M/P is not an integer, the program prints an error message and stops. The CT method approximates the one-dimensional hindered-rotator problem with a simple model potential of the form

$$V = \frac{W}{2}(1 - \cos M\phi)$$

where W is the height of the potential barrier, and ϕ is a coordinate that parameterizes the internal rotational motion. For a potential of the above form with a geometry-independent moment of inertia, the barrier height W_j, moment of inertia I_j, and one-dimensional torsional frequency $\omega_{tor,j}$ are interrelated by

$$\omega_{\text{tor},j} = M\left(\frac{W_j}{2I_j}\right)^{1/2}$$

If the frequencies $\omega_{\text{tor},j}$ are approximated as the same for all of the P minima and the energies U_j are approximated as 0 (same as the global minimum energy) for all of the P minima, we label this as single-frequency-and-energy (SFE) approximation. If only the frequency is approximated as the same for all the distinguishable minima, we label it as the single-frequency (SF) approximation. Otherwise, the treatment is labeled FULL.

Note that the single-conformer (SC) approximation in Ref. 52 is another name for the special case of $P=1$. Therefore, starting from version 2010 of *Polyrate* program, this SC approximation is not used as a stand-alone method.

10.B.3.a.ii RPG method

The torsional Pitzer–Gwinn method based on a reference potential gives the torsional partition function as

$$Q_{\text{tor}}^{\text{RPG}} = \frac{1}{P}\sum_{j=1}^{P}\frac{Q_{\text{tor}}^{\text{HO}}(\omega_{\text{tor},j})}{Q_{tor}^{\text{CHO}}(\omega_{\text{tor},j})}Q^{\text{FR}}\exp(-\beta W/2)I_0(\beta W/2)$$

where $\omega_{\text{tor},j}$ is the frequency at the well j, W is the averaged barrier height of all the wells to their left and right maxima, and I_0 is a modified Bessel function. If one uses the frequency associated with the minimum of the 1D torsional reference potential to approximate all the $\omega_{\text{tor},j}$, this equation reduces to the Eq. 24 in Ref. 62. The reader is referred to Ref. 62 for all further details.

10.B.3.a.iii. AS method

AS method uses a correction factor to adjust the RPG method
$$Q_{\text{tor}}^{\text{AS}} = \frac{1 + P_2\exp[-\beta W/2]}{1 + P_1\exp[-\beta W/2]}Q_{\text{tor}}^{\text{RPG}}$$
where P_1 and P_2 are fifth order polynomial functions of $x \equiv 1/Q^{\text{FR}}$ and $y \equiv \beta W/2$ which is given by Eq. 31 and 32 in Ref. 62.

10.B.3.a.iv. SRPG method

If the hindered-rotator potential does not closely resemble the form of $V = \dfrac{W}{2}(1 - \cos M\phi)$, it may be convenient to decompose the partition function into a sum of contributions from particular wells and to fit the potential between each peak and minimum using a simple cosine line shape. For each well, the contribution may be divided into the part to the left of the minimum and the part to the right of the minimum, and each part of the classical partition function may be evaluated analytically. This is referred as the segmented-reference classical (SRC) approximation, and it yields

$$Q^{\text{SRC}} = \sum_{j=1}^{P} Q_j^{\text{SRC}}$$

where

$$Q_j^{\text{SRC}} = \exp(-\beta U_j) Q_j^{\text{FR}}$$
$$\times \left[\frac{(\phi_j - \phi_j^L)}{2\pi} \exp(-\beta W_j^L / 2) I_0(\beta W_j^L / 2) + \frac{(\phi_j^R - \phi_j)}{2\pi} \exp(-\beta W_j^R / 2) I_0(\beta W_j^R / 2) \right]$$

where U_j is the energy of the local minimum centered at ϕ_j relative to the energy at the global minimum, W_j^L and W_j^R are the left and right barrier heights located at ϕ_j^L and ϕ_j^R, respectively, and

$$Q_j^{\text{FR}} = \frac{\sqrt{2\pi I_j kT}}{\hbar \sigma}$$

Quantum effects may be approximated by the Pitzer-Gwinn approximation to yield the segmented-reference Pitzer-Gwinn method

$$Q_{\text{tor}}^{\text{SRPG}} = \sum_{j=1}^{P} \frac{Q_j^{\text{HO}}(\omega_{\text{tor},j})}{Q_j^{\text{CHO}}(\omega_{\text{tor},j})} Q_j^{\text{SRC}}$$

10.B.3.a.v. SAS method

The AS method has a segmented approach that is analogous to the SRPG method, and we refer to this analog to the segmented Ayala-Schlegel (SAS) method, in which each occurrence of $\exp(-\beta W_j / 2)$ in the SRPG method is replaced by a term like

$$\exp(-\beta W_j / 2) \frac{1 + P_2(I_j, W_j)\exp(-\beta W_j / 2)}{1 + P_1(I_j, W_j)\exp(-\beta W_j / 2)}$$

where P_1 and P_2 are given by Eq. 31 and 32 in Ref. 62.

The test runs ch4ohtr1, ch4ohtr3, ch4ohtr4, cwmctr2, cwmctr3, and h3o2fu31tr1 are examples of using the hindered rotor options.

10.B.3.b. MS-T method

Starting from version 2016 of *POLYRATE,* the torsional-potential anharmonicity along the MEP can be calculated using the MS-T method with a coupled torsion potential (i.e., MS-T(C)), but only the input single transition structure is considered as generalized transition states in the rate constant calculation; this is the SS-T treatment, i.e. *Polyrate* used by itself does not compute the Multipath VTST (MP-VTST) rate constant, which requires multiple transition structures in order to compute the variational and transmission coefficients that correspond to their MEPs. Note that reactants and products are still treated by harmonic oscillator approximation. To obtain rate constants calculated by the full MS-T partition functions for reactants, products, and GTS, as well as the MP-VTST rate constant, the user has to use MSTOR program (which is a separate program) to calculate the rate constants at the conventional transition state and then use the variational effects (ratio of CVT rate to TST rate) and transmission coefficients calculated by *Polyrate* program (see the MSTOR manual for more details).

The conformational-rovibrational partition function using the MS-T method for a species with J conformational structures is

$$Q_{\text{con-rovib}}^{\text{MS-T}} = \sum_{j=1}^{J} Q_j^{\text{T}}$$

$$Q_j^{\text{T}} \equiv Q_{\text{rot},j} \exp\left(-\beta U_j\right) Q_j^{\text{QH}} T_j$$

where Q_j^{T} is the contribution of structure j to the rovibrational partition function with torsional anharmonicity, $Q_{\text{rot},j}$ is the classical rotational partition function for structure j, U_j is its relative potential energy, Q_j^{QH} is its QH vibrational partition function partition function, and T_j is a factor to account torsional-potential anharmonicity for all torsional modes in the structure j. In particular, T_j is

$$T_j = \left(2\pi\hbar\beta\right)^{t/2} \frac{\displaystyle\prod_{m=1}^{F} \omega_{j,m}}{\displaystyle\prod_{\bar{\bar{m}}=1}^{F-t} \bar{\bar{\omega}}_{j,\bar{\bar{m}}}} \frac{\sqrt{\det \mathbf{D}_j}}{\displaystyle\prod_{\tau=1}^{t} M_{j,\tau}} \prod_{\eta=1}^{t} \exp(-\beta W_{j,\eta}/2) I_0(\beta W_{j,\eta}/2)$$

where t is number of torsional modes, F is number of bound vibrational modes (stretches, bends, and torsions), $\omega_{j,m}$ and $\bar{\bar{\omega}}_{j,\bar{\bar{m}}}$ are respectively the normal mode harmonic frequencies and the torsion-projected normal mode frequencies (both sets scaled), $\mathbf{D}_j$ is the moment of inertia matrix of Kilpatrick and Pitzer, $M_{j,\tau}$ is the periodicity of torsion τ in structure j, $W_{j,\eta}$ is the torsional barrier height based on a coupled torsional potential, and I_0 is a modified Bessel function. Note that for nonstationary points along a reaction path j (starting from the saddle point j), the vibrational mode along reaction coordinate is projected out and the torsional-potential anharmonicity T_j is calculated for each nonstationary points where Hessian is calculated or available. Therefore, vibrational partition functions of points along reaction path includes the torsional-potential anharmonicity and then torsional-potential anharmonicity is taken into account for searching variational transition state.

To calculate T_j, one only needs to input the set of $M_{j,\tau}$ values for torsions and projected frequencies have to be calculated using non-redundant internal coordinates.

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. 22] and by Schmidt, Gordon, and Dupuis [Ref. 20, Sect. 22]. 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. 22], 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 × 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 × 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 16.C.3, 16.C.9, and 16.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. 22].) 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. 22]. This method does not require the

derivative of the gradient. A sample run with this option is discussed in Subsection 16.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 12.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 MEPTYPER or MEPTYPEP 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 12.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 $\omega_m^{LL}(s)$, $m = 1, 2, ..., F\text{-}1$ (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

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

where

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

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

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

where

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

and

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

where

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

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

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

where

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

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

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

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

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

we choose

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

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

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

where μ is the scaling mass (normally set to 1.0 amu), and $\omega^{\neq}$ 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. *icl* for vibrations and *deckart* for $V_{\text{MEP}}(s)$) as the default VTST-IOC method. Beginning with version 8.0 the defaults are *icl* 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 $\omega^{\neq}$ 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	*ica, seckart*
2	*icl, deckart*
3	*icl, seckart*
4	*icr, seckart*
5	*ica, deckart*
6	*icr, deckart*

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:

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

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

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

and

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

where μ is the scaling mass, $\omega^{\neq}$ 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

$$s_A = -\min\left(\left|s_A^0\right|, 2s_B^0\right) \tag{4}$$

and

$$s_B = -\min\left(2\left|s_A^0\right|, s_B^0\right) \tag{5}$$

Finally, the parameters s_0 and L are obtained as:

$$s_0 = \frac{s_A + s_B}{2} \tag{6}$$

and

$$L = \frac{-\left|s_A\right| + s_B}{2} \tag{7}$$

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 → 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 → C + D), where the s variable occupies interval [-∞, ∞], the mapped z values are located in the finite interval [-1, 1]. For A → B + C or A +B → 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:

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

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 μOMT.

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 12.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 $V_{MEP}(s)$ 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 $\mu^{SC}(s)$ 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 12.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. (It is very important to use curvilinear coordinates rather than Cartesian coordinates for VTST calculations; Cartesian coordinate will very likely yield imaginary frequencies along the reaction coordinate, which will cause unphysical VTST results.) 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 **G** and force constant matrix **F** at the save point are first transformed to the curvilinear internal gradient vector **g** and curvilinear internal force constant matrix **f** using equations 30 and 31 in Ref. 37 of Sect. 22 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 **B** matrix and **C** tensor are constructed using the difference Cartesian coordinate formulas from Pariseau et al. [Ref. 38, Sect. 22] for the bond stretch and nonlinear valence bend coordinates and from Hoy et al. [Ref. 39, Sect. 22] for the degenerate linear bend coordinate. The difference Cartesian **B** matrix and **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 **B** matrix and **C** tensor are constructed by using explicit FORTRAN statements [Ref. 40, Sect. 22] for the **B** matrix and **C** tensor derived by Challacombe and Cioslowski [Ref. 41, Sect. 22], 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 **f** is projected as

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

where

$$\mathbf{p} = \frac{\mathbf{gg}^{\mathrm{T}}}{\mathbf{g}^{\mathrm{T}}(\mathbf{BuB}^{\mathrm{T}})\mathbf{g}},$$

before applying the Wilson **GF** method [Ref. 36, Sect. 22] 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 12.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 12.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 Sect. 21. (It is very important to use curvilinear coordinates rather than Cartesian coordinates for VTST calculations; Cartesian coordinate will very likely yield imaginary frequencies along the reaction coordinate, which will cause unphysical VTST results.)

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 3*N*-6 for the *curv1* and *curv2* options. The option *curv3* allows the number of internal coordinates to be as large as 3*N* + 6.

10.M. IVTST0FREQ 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 $(\omega^{\neq} > \omega_R)$ and $(\omega^{\neq} > \omega_P)$ or if $(\omega^{\neq} < \omega_R)$ and $(\omega^{\neq} < \omega_P)$, where $\omega_m^{\neq} = \omega_m(s = 0)$, $\omega_R = \omega_m(s = -\infty)$, and $\omega_P = \omega_m(s = +\infty)$, an Eckart function is used to approximate the frequencies along the reaction path:

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

where

$$A_m = \omega_m(s = +\infty) - \omega_m(s = -\infty) \tag{2}$$

$$B_m = (2\omega_m^{\neq} - A_m - 2C_m) \pm 2\sqrt{\left[(\omega_m^{\neq} - C_m)(\omega_m^{\neq} - A_m - C_m)\right]} \tag{3}$$

$$C_m = \omega_m(s = -\infty) \tag{4}$$

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

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

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]

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

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

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

$$B = (2V^{\neq} - A - 2C) \pm 2\sqrt{\left[(V^{\neq} - C)(V^{\neq} - A - C)\right]} \tag{10}$$

$$V^{\neq} = V_{\text{MEP}}(s = 0) \tag{11}$$

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

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

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

where

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

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

$$x_m = 2\frac{\omega_m^{\neq} - E_m}{A_m} \tag{15}$$

c. For cases where two consecutive of the three frequencies are the same (i.e., $\omega^{\neq} = \omega_{\text{R}}$ or $\omega^{\neq} = \omega_{\text{P}}$, 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 $\omega_{\text{R}} = \omega^{\neq} \neq \omega_{\text{P}}$, then

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

where

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

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

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 $-\infty < s < s_{P-W}$.

    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:

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

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 $\omega_{m,R} = \omega_m^{\neq}$ or $\omega_m^{\neq} = \omega_{m,P-W}$, then the value of $\omega_{m,R}$ or $\omega_{m,P-W}$ is modified by ± 2 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 $(\omega^{\neq} > \omega_{R-W})$ and $(\omega^{\neq} > \omega_{P-W})$ or if $(\omega^{\neq} < \omega_{R-W})$ and $(\omega^{\neq} < \omega_{P-W})$, two cut-off Gaussian functions are used to approximate the frequencies along the reaction path:

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

$$A_{m,x} = (\omega_m^{\neq} - \omega_x)\exp(B_{m,x}); \qquad x = \text{R-W,P-W} \tag{21}$$

$$C_{m,x} = \omega_x; \qquad\qquad x = \text{R-W,P-W}. \tag{22}$$

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

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

b. If $\omega_{m,\text{R-W}} = \omega_m^{\neq}$ or $\omega_{m,\text{P-W}} = \omega_m^{\neq}$ 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 $\omega_{m,\text{R-W}} = \omega_m^{\neq} \neq \omega_{m,\text{P-W}}$, then the frequency is set to $\omega_{\text{R-W}}$ when $s \leq 0$ and evaluated using eqns. (20–23) when $s > 0$.

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

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

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

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 *nolctrst*. 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 modules 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 include tunneling into vibrationally excited final states in a second run if still worth it after looking at the output.

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

10.O. MS-VTST, MP-VTST, FMS-VTST, and FMR-VSTT calculations by using both the *Polyrate* and MSTOR programs

For molecules and transition states that have multiple conformations generated by torsions, we recommend MS-VTST [61], MP-VTST [62,63,72], FMS-VTST [68], or FMP-VTST [68] calculations. Such calculations can be carried out by making runs with *Polyrate* and MSTOR programs. The MSTOR program [64-66] is a program for applying the internal-coordinate torsional anharmonicity method [70,71,68] for treating torsional anharmonicity (MS-T); it is available as a separate package at

http://comp.chem.umn.edu/mstor/.

The internal-coordinate torsional anharmonicity method is a practical thermochemical method for including multiple structures and torsional anharmonicity to calculate conformational–vibrational–rotational partition function for complex systems. This scheme obviates the need of earlier approaches to identify torsional motions with specific normal modes, and it approximately accounts for coupling of torsional modes to one another and to other modes. It includes the conformational free energy associated with the existence of multiple structures (conformations) and the additional effects of torsional

potential anharmonicity. The original paper [70] included methods based on a subset of structures as well as the method using all structures. The method containing all structures was originally call MS-AS, but now we usually use all structures and, as explained in ref. [64], we simply refer to the method employing all structures as MS-T.

In the original papers introducing MS-VTST [61] and MP-VTST [62,63,72], we included multistructural torsional anharmonicity only at stationary points; tunneling transmission coefficients and recrossing transmission coefficients were calculated in the quasiharmonic (QH) approximation. (A quasiharmonic calculation is one that uses the harmonic oscillator formulas for vibrational partition functions but with effective (scaled) frequencies rather than the harmonic frequencies; the effective frequencies are usually chosen to improve the estimate of the zero point energy [69].) In Ref. [68] we introduced the capability to include multistructural anharmonicity all along the reaction path. To be precise, we will label calculations by the original methods as MS-VTST [61] and MP-VTST, and we will label calculations with multistructural anharmonicity all along the reaction path [68] as full MS-VTST (FMS-VTST) and full MP-VTST (FMP-VTST).

In carrying out MS-VTST and MP-VTST calculations, *Polyrate* is used to carry out single-structure quasiharmonic (SS-QH) calculations for reaction paths. In carrying out FMS-VTST and FMP-VTST calculations, *Polyrate* is used to carry out single-structure quasiharmonic what we shall call single-structure torsion (SS-T) calculations. Throughout this discussion, we use the word structure as a synonym for conformation. An SS calculation involves a single conformational structure of each reactant and a single conformational structure of the transition state. SS-QH calculations use the QH approximation along the reaction path, and SS-T calculations use the MS-T approximation along the reaction path.

A note about chirality: when one is carrying out a single-structure calculation to compare with experiment, if the structure has a distinguishable mirror image, then one needs to include an extra factor of two in its rate constant, as discussed elsewhere [67]. However, in this section, we literally mean a single structure and the factor of two is not included in such a case.

Some preliminaries about notation: (1) When we say "variational effects" we mean the difference between a variational transition state theory calculation and a conventional transition state theory calculation. In MS-VTST, MP-VTST, FMS-VTST, and FMP-VTST calculations, variational effects are included by multiplying a conventional transition state theory rate constant by a transmission coefficient that is called a recrossing transmission coefficient. (2) The methods explained here may be used with a variety of approximations for the tunneling, for example, ZCT, SCT, LCT, μOMT,

or LAT. In general we use the abbreviation MT ("multidimensional tunneling") to refer to any of these. (3) In our original paper

This subsection explains how to perform MS-VTST, MP-VTST, FMS-VTST, and FMP-VTST calculations by combining results from *Polyrate* and MSTOR. In brief, the MSTOR program is used to perform all the calculations for the stationary points (reactants, saddle points, and products), and *Polyrate* is used to perform SS-QH or SS-MP reaction path calculations to obtain variational effects and transmission coefficients.

In order to carry our either an MS-VTST or an MP-VTST calculation, the first step is to find all the conformational structures of the transition state (that is, all the saddle points leading from reactants to products along parallel reaction paths (as contrasted to reaction paths in series)) and all the structures of the reactant or reactants. Structures of the transition state are called transition structures, and structures of the reactant(s) are called reactant equilibrium structures or just reactant structures. Next one numbers the transition structures so that the lowest-energy one is called $k = 1$, and in another list one orders the reactant equilibrium structures so that the lowest-energy one is called $k = 1$. The energy difference between the lowest-energy transition structure and the lowest-energy reactant equilibrium structure is called the classical barrier height, $V^{\ddagger}$.

Then one uses MSTOR to calculate MS-T partitions functions for the transition state and the reactant(s). For the reactant of a unimolecular reaction, this partition function is called $Q^{\text{MS-T,R}}$. For the reactants of a bimolecular reaction, the product of the partition functions of the two reactants (Q^{R1} and Q^{R2}) is also called $Q^{\text{MS-T,R}}$, but the quantity we will need to calculate rate constants in the bimolecular case is Φ (the partition function per unit volume) rather than Q (the partition function) so that one has to convert Q to Φ when calculating bimolecular rate constants. Note also that it is possible that MS-T will be used for only one of the two reactants. If the reactant R2 does not have torsion, then one may use a simpler method, for example, the harmonic or quasiharmonic approximation, for Q^{R2}. Then $\Phi^{\text{MS-T, R}}$ is the bimolecular reactants' partition function per unit volume (with its zero of energy at the sum of the potential energies of the lowest-energy reactant structures), and $Q^{\text{MS-T, R}}$ is the total partition function of a unimolecular reactant (with its zero of energy at the potential energy of the lowest-energy reactant structure).

The partition function for the transition state is called $Q^{\text{MS-T,}\ddagger}$. Note that this partition function is sum over contributions from distinguishable structures:

$$Q^{\text{MS-T,}\ddagger} = \sum_{k=1}^{K} Q_k^{\text{T,}\ddagger} \tag{1}$$

where K is the number of distinguishable conformational structures of the transition state, and $Q_k^{\text{T,}\ddagger}$ is the rovibrational partition function of saddle point k with torsional potential anharmonicity (T). Note that the difference in energy between the zero of energy used for the transition state partition functions $Q_k^{\text{T,}\ddagger}$ and that used for the reactant partition function is taken here to be the classical barrier height, which is the potential energy of the lowest-energy conformational structure ($k = 1$) of the saddle point minus the potential energy of the lowest-energy equilibrium structure of the reactant.

The next step is to perform one or more SS calculations with *POLYRATE*; this is explained next.

10.O.1. MS-VTST

The MS-VTST rate constants including multidimensional tunneling are given by the following formulas for a bimolecular reaction and a unimolecular reaction respectively:

$$k_{\text{bimol}}^{\text{MS-CVT/MT}} = \kappa_1^{\text{QH}} \Gamma_1^{\text{QH-CVT}} \frac{k_B T}{h} \frac{Q_{\text{elec}}^{\ddagger} \sum_{k=1}^{K} Q_k^{\text{T,}\ddagger}}{\Phi^{\text{MS-T,R}}} \exp(-\beta V^{\ddagger}) \tag{2}$$

$$k_{\text{unimol}}^{\text{MS-CVT/MT}} = \kappa_1^{\text{QH}} \Gamma_1^{\text{QH-CVT}} \frac{k_B T}{h} \frac{Q_{\text{elec}}^{\ddagger} \sum_{k=1}^{K} Q_k^{\text{T,}\ddagger}}{Q^{\text{MS-T,R}}} \exp(-\beta V^{\ddagger}) \tag{3}$$

where κ_1^{QH} is the quasiharmonic tunneling transmission coefficient for the reaction path passing through transition structure 1, β is $1/k_{\text{B}}T$, k_B is Boltzmann's constant, T is temperature, h is Planck's constant, $Q_{\text{elec}}^{\ddagger}$ is the electronic partition function of the conventional transition state (which is denoted as $\ddagger$); and the CVT recrossing transmission coefficient for structure 1 is $\Gamma_1^{\text{QH-CVT}}$, which is given by the ratio of the quasiharmonic CVT rate constant for the reaction path through transition structure 1 to the quasiharmonic conventional TST one for transition structure 1.

In order to obtain κ_1^{QH} and $\Gamma_1^{\text{QH-CVT}}$, one performs an SS-QH calculation with *POLYRATE*. The κ_1^{QH} is printed out in the *Polyrate* output file and $\Gamma_1^{\text{QH-CVT}}$ is calculated

$$\Gamma_1^{\text{QH-CVT}} = \frac{k^{(1),\text{QH-CVT}}}{k^{(1),\text{QH-TST}}} \tag{4}$$

10.O.2. FMS-VTST

An FMS-VTST calculation is the same as an MS-VTST one except for the transmission coefficients. One replaces κ_1^{QH} and $\Gamma_1^{\text{QH-CVT}}$ by κ_1^{T} and $\Gamma_1^{\text{T-CVT}}$, which are obtained by running SS-T calculations in *Polyrate* with the keyword SSTOR. In particular, the recrossing transmission coefficient $\Gamma_1^{\text{T-CVT}}$ is calculated as

$$\Gamma_1^{\text{T-CVT}} = \frac{k^{(1),\text{T-CVT}}}{k^{(1),\text{T-TST}}} \tag{5}$$

10.O.3. MP-VTST

In MP-VTST, we calculate the transmission coefficient as a properly weighted average over tunneling transmission coefficients and variational transmission coefficients for several reaction paths instead of calculating transmission coefficients only from one path as is done in MS-VTST. This subsection gives the procedure to perform MP-VTST calculations.

The MP-VTST rate constants including multidimensional tunneling are given by the following formulas for a bimolecular reaction and a unimolecular reaction respectively:

$$
\begin{aligned}
k_{\text{bimol}}^{\text{MP–CVT/MT}} &= \frac{k_B T}{h} \frac{Q_{\text{elec}}^{\ddagger}}{\Phi^{\text{MS-T,R}}} \exp(-\beta V^{\ddagger}) \sum_{k=1}^{K} \kappa_k^{\text{QH}} \Gamma_k^{\text{QH–CVT}} Q_k^{\text{T},\ddagger} \\
&\approx \gamma \frac{k_B T}{h} \frac{Q_{\text{elec}}^{\ddagger} \sum_{k=1}^{K} Q_k^{\text{T},\ddagger}}{\Phi^{\text{MS-T,R}}} \exp(-\beta V^{\ddagger})
\end{aligned}
\tag{6}
$$

$$
k_{\text{unimol}}^{\text{MP–CVT/MT}} \approx \gamma \frac{k_B T}{h} \frac{Q_{\text{elec}}^{\ddagger} \sum_{k=1}^{K} Q_k^{\text{T},\ddagger}}{Q^{\text{MS-T,R}}} \exp(-\beta V^{\ddagger}) \tag{7}
$$

where the MP-VTST generalized transmission coefficient γ is approximated by

$$\gamma = \frac{\sum_{k=1}^{P} \kappa_k^{\mathrm{QH}} \Gamma_k^{\mathrm{QH\text{-}CVT}} Q_k^{\mathrm{T},\ddagger}}{\sum_{k=1}^{P} Q_k^{\mathrm{T},\ddagger}} \tag{8}$$

To evaluate the quantities in these formulas, we have to make P runs of *Polyrate* (where $P \le K$), one for each of the P user-selected distinguishable structures of the transition state (note that these structures are numbered by $k = 1, ..., P$). Each of the runs gives a quasiharmonic CVT rate constant $k^{(k),\mathrm{QH\text{-}CVT}}$, a quasiharmonic conventional TST rate constant $k^{(k),\mathrm{QH\text{-}TST}}$, a quasiharmonic tunneling transmission coefficient κ_k^{QH}, and a quasiharmonic CVT recrossing transmission coefficient $\Gamma_k^{\mathrm{QH\text{-}CVT}}$. The first three are printed in the *Polyrate* output file, and the fourth is given by

$$\Gamma_k^{\mathrm{QH\text{-}CVT}} = \frac{k^{(k),\mathrm{QH\text{-}CVT}}}{k^{(k),\mathrm{QH\text{-}TST}}} \tag{9}$$

The other quantities needed for MP-VTST rate constants can be calculated by using the MSTOR program.

Note also that Γ_k^{CVT} is independent of which reactant structure is used, but the choice of reactant structure affects the lower-energy limit of the integration used to calculate κ_k, and thus it could affect the values calculated for the rate constants at low temperature. Therefore all the additional *Polyrate* runs ($k = 2, ..., P$) needed to upgrade MS-VTST to MP-VTST should use structure k of the transition state and the lowest-energy structure of the reactant(s). Note that one has to calculate a reaction path long enough to make sure the κ_k is converged. In case the saddle point k is connected with a reactant(s) conformation much higher in energy than the lowest-energy structure of the reactant(s), there is possibilities that the κ_k is overestimated because of the program is trying to integrate κ_k to the lowest-energy structure of the reactant(s). This is one aspect if the calculation that needs to monitored carefully if calculations are made at low temperature.

Note that the wavy equal sign in equation (6) can be replaced by a straight equal sign if P equals K. When $P = 1$, the MP-VTST rate reduces to the MS-VTST rate.

Alternatively one also can calculate the MP-VTST rate constants by the following procedure. Let's define the multistructural torsional anharmonicity factors for reactants (R1 and R2 for bimolecular reactions, R1 = R for unimolecular reactions) and for saddle point 1 (lowest-energy saddle point) as

$$F^{R1} = \frac{Q^{MS\text{-}T,R1}}{Q_1^{QH,\,R1}} \tag{10}$$

$$F^{R2} = \frac{Q^{MS\text{-}T,R2}}{Q_1^{QH,\,R2}} \tag{11}$$

$$F^{\ddagger,\,1} = \frac{Q^{MS\text{-}T,\ddagger}}{Q_1^{QH,\,\ddagger}} \tag{12}$$

Then we obtain the MP-VTST rate constants as

$$k_{\text{bimol}}^{MP\text{--}CVT/MT} = \gamma\,\frac{F^{\ddagger,1}}{F^{R1}F^{R2}}\,k_{\text{bimol}}^{(1),\,QH\text{-}TST} \tag{13}$$

$$k_{\text{unimol}}^{MP\text{--}CVT/MT} = \gamma\,\frac{F^{\ddagger,1}}{F^{R1}}\,k_{\text{unimol}}^{(1),QH\text{-}TST} \tag{14}$$

10.O.4. FMP-VTST

An FMP-VTST calculation is the same as an MP-VTST one except for the transmission coefficients. One replaces κ_k^{QH} and $\Gamma_k^{QH\text{-}CVT}$ by κ_k^{T} and $\Gamma_k^{T\text{-}CVT}$ in equation (8); these quantities are obtained by running SS-T calculations in *Polyrate* with the keyword SSTOR. In particular, the recrossing transmission coefficient $\Gamma_k^{T\text{-}CVT}$ is calculated as

$$\Gamma_k^{T\text{-}CVT} = \frac{k^{(k),T\text{-}CVT}}{k^{(k),T\text{-}TST}} \tag{15}$$

Part III Theory Background for VRC-VTST

11. THEORETICAL BACKGROUND ON VRC-VTST

For an association reaction without a barrier or with a negligible barrier, we recommend using the multifaceted (MF) extension of variable-reaction-coordinate variational transition state theory (VRC-VTST) . The references for VRC-VTST is Y. Georgievskii and S. J. Klippenstein, *J. Chem. Phys.* **2003**, *118*, 5442, and the reference for the extension of VRC-VTST to use MF dividing surfaces is Y. Georgievskii and S. J. Klippenstein, *J. Phys. Chem. A* **2003**, *107*, 9776.

In VRC-VTST, the reaction coordinate and dividing surfaces are defined in a different way than in RP-VTST. In VRC-VTST, the reaction coordinate s is defined by the distance between a pivot point on one reactant and that on another, and the dividing surfaces are defined in terms of pivot points tied to each of the reactants. Choices of the number and location of these pivot points yield different definitions of the reaction coordinates and hence of the generalized transition state theory dividing surfaces, and reactive flux is variationally minimized with respect to both these choices and the distance between pivot points in the two reactants. The original version of VRC-VTST employed one pivot point on each reactant, and the MF extension involves two and more pivot points on one or both reactants. In this manual we consider the use of MF dividing surfaces to a special case of VRC-VTST. The evaluations of the reactive flux involves Monte Carlo integration over the coordinate space degrees of freedom of the classical phase space representation of the generalized transition state partition function for arbitrary orientations of the reagents over a wide range of interfragment separations.

Because there is no saddle point for a barrierless association reaction, the conventional TST rate is not calculated. The START section only needs the electronic degeneracy information for electronic partition function calculations. Association rate constants can be calculated by CVT, µVT, or *E,J* -µVT. Tunneling calculations cannot be performed for this type of calculation.

11.A. Variable reaction coordinate

When a single-faceted dividing surface is used, the reaction coordinate s for a barrierless association reaction is defined as the distance between two pivot points, one associated

with each of the two reactants. Since s is defined as a distance here, its value is always positive. Thus, it has a different meaning from the s used in RP-VTST. The following scheme shows the definition of the reaction coordinate s for the case of a single-faceted dividing surface.
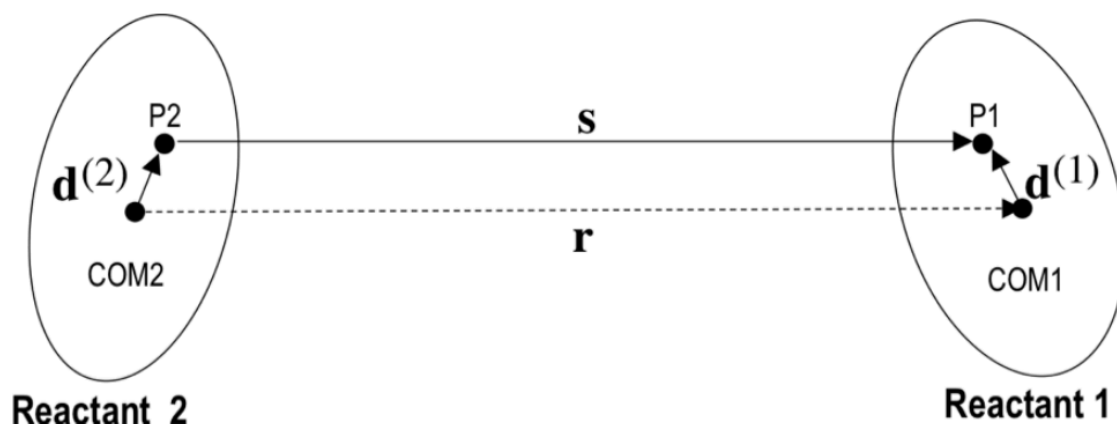
Here, the two ovals represent the two reactants. COM1 and COM2 are their centers of mass. P1 and P2 are pivot points associated with reactant 1 and reactant 2, respectively. $\mathbf{d}^{(1)}$ and $\mathbf{d}^{(2)}$ are vectors connecting a pivot point with its associated reactant's center of mass, respectively. When the distance s between two pivot points is fixed, the directions of $\mathbf{s}$, $\mathbf{d}^{(1)}$, and $\mathbf{d}^{(2)}$ are sampled over the whole phase space simultaneously. The lengths of these vectors are fixed during the sampling over their orientations.

In the VRC-VTST approach, the modes of motion are divided into two sets, conserved modes and transitional modes. The conserved modes correspond to the vibrational modes of the two reactants. In the complex of two reactants, these modes usually correspond to the modes with highest vibrational frequencies. VRC-VTST, at least in its present state of development, assumes that changes of conserved modes are negligible during the reaction process. The transitional modes include the other vibrational modes except conserved modes in the complex and the overall rotations. These modes change from free rotation and translation at infinite separation of the reactants to vibrational motion and overall rotation during the reaction process. The new vibrational modes are very floppy and are best thought of as liberations.

The high-pressure limit rate constant for the generalized transition state at location s along the reaction coordinate and at the temperature T can be expressed as

$$k(T,s) = \frac{\hbar^2}{2\pi} g_e \frac{\sigma_1\sigma_2}{\sigma^{\ddagger}Q_1Q_2} \left(\frac{2\pi}{\mu k_B T}\right)^{\frac{3}{2}} \int dE \, e^{-E/k_b T} \, dJ \, N(E,J,s) \qquad (1)$$

where s is the value of reaction coordinate, which is defined above as the distance between two pivot points; g_e is the ratio of the electronic partition function of the transition state to the product of the electronic partition functions of reactants; μ is the reduced mass; Q_1 and Q_2 are the rotational partition functions of the reactants calculated without symmetry numbers; J is the unitless total angular momentum quantum number; $N(E,J,s)$ is the number of accessible states of the generalized transition state s for total energy E and angular momentum $J\hbar$; and σ_1, σ_2, and $\sigma^{\ddagger}$ are the rotational symmetry numbers for the reactants and transition state, respectively.

The keyword SIGMAF in RATE section (see Section 13.A.6) is equal to $\sigma_1\sigma_2/\sigma^{\ddagger}$. For reactions in which the two reactants are the same, SIGMAF is 0.5; for reactions in which the two reactants are different, SIGMAF is 1.0. This is explained in the following paragraph. As background for this explanation, we first note that eq. (137) in the review by Bao and Truhlar[1] is identical to eq. (4) in the *PNAS* paper by Bao et al.,[2] although in the *PNAS* paper we include the rotational symmetry factors in the reactant partition functions, whereas in eq. (137) of the review we separated them. Note also that σ_1 and σ_2 should have their correct symmetry numbers as usual.

The next question is: What is $\sigma^{\ddagger}$? Georgievskii and Klippenstein[3] (in their eq. 2) define

$$\sigma = \sigma_1\sigma_2/\sigma^{\ddagger}$$

and they make the assumption that – since the transition state is loose – there is free rotation of both species, such that the transition state partition function contains the product of the two rotational partition functions. Then

$$\sigma^{\ddagger} = \sigma_1\sigma_2\sigma_{add}$$

[1] J. L. Bao and D. G. Truhlar, Variational Transition State Theory: Theoretical Framework and Recent Developments, *Chem. Soc. Rev.,* 2017, **46**, 7548-7596. doi.org/10.1039/C7CS00602K

[2] J. L. Bao, X. Zhang and D. G. Truhlar, Association of CF_2 and Dissociation of C_2F_4 by Variational Transition State Theory and System-Specific Quantum RRK Theory, *Proc. Natl. Acad. Sci. U. S. A.*, 2016, **113**, 13606. doi.org/10.1073/pnas.1616208113

[3] Y. Georgievskii and S. J. Klippenstein, Long-range transition state theory, *J. Chem. Phys.,* 2005, *122*, 194103. doi.org/ 10.1063/1.1899603

where σ_{add} is the additional symmetry gained in the transition state. Because we sample all possible orientations (Euler angles) of the transition state classically in VRP-TST, we must divide by two when the association reactants are identical because the phase space may be divided into two quantum mechanically indistinguishable spaces when the two reactants are identical. This means that σ_{add} equals 1 if the species are different and equals 2 if they are the same. Therefore $\sigma_1\sigma_2/\sigma^{\ddagger}$ equals 1 if the species are different and equals 0.5 if they are the same. One could call this the loose-transition-state assumption for the symmetry number of the transition state.

The number of accessible states for the transitional modes with total energy E and angular momentum quantum number J can be written as

$$N(E,J,s) = \left\langle N_q(E,J,Q,s) \right\rangle_\Omega \tag{2}$$

where $\left\langle \cdots \right\rangle_\Omega$ denotes the average over all possible orientations of the two reactants and of the orientation of the vector $\mathbf{s}$; Q means a specific configuration of the system. Since this multidimensional integral is carried out by Monte Carlo integration, the configurations of the system are randomly sampled over the whole phase space corresponding to a given s. In our implementation, one specific random configuration is determined by the following procedure:

(1) P1 is always taken as the origin of the system for every random configuration.

(2) The position of P2 is determined by a random solid angle $\Omega_{12} \equiv (\theta_{12}, \phi_{12})$

$$\mathbf{x}_{P2} = \mathbf{A}_{12}^{-1}(\Omega_{12}) \begin{pmatrix} 0 \\ 0 \\ s \end{pmatrix} \tag{3}$$

where $\mathbf{x}_{P2}$ is the Cartesian coordinate of pivot point P2; $\mathbf{A}_{12}^{-1}(\Omega_{12})$ is the inverse rotation matrix,

$$\mathbf{A}_{12}^{-1} = \begin{pmatrix} \cos\theta_{12}\cos\phi_{12} & -\sin\phi_{12} & \sin\theta_{12}\cos\phi_{12} \\ \cos\theta_{12}\sin\phi_{12} & \cos\phi_{12} & \sin\theta_{12}\sin\phi_{12} \\ -\sin\theta_{12} & 0 & \cos\theta_{12} \end{pmatrix} \tag{4}$$

(3) Since the Cartesian coordinates of P1 ($\mathbf{x}_{P1}$) and P2 ($\mathbf{x}_{P2}$) are known, the Cartesian coordinates of the center of mass of the kth reactant ($\mathbf{x}_{comk}$) are given by

$$\mathbf{x}_{comk} = \mathbf{A}_k^{-1}(\Omega_k) \begin{pmatrix} 0 \\ 0 \\ d^{(k)} \end{pmatrix} + \mathbf{x}_{Pk}, \quad (k = 1, 2) \tag{5}$$

$$\mathbf{A}_k^{-1}(\Omega_k) =$$

$$\begin{pmatrix} -\sin\psi_k\sin\phi_k + \cos\theta_k\cos\psi_k\cos\phi_k & -\cos\psi_k\sin\phi_k - \cos\theta_k\cos\phi_k\sin\psi_k & \sin\theta_k\cos\phi_k \\ \sin\psi_k\cos\phi_k + \cos\theta_k\cos\psi_k\sin\phi_k & \cos\psi_k\cos\phi_k - \cos\theta_k\sin\phi_k\sin\psi_k & \sin\theta_k\sin\phi_k \\ -\cos\psi_k\sin\theta_k & \sin\psi_k\sin\theta_k & \cos\theta_k \end{pmatrix}$$

$$(6)$$

where $\mathbf{A}_k^{-1}(\Omega_k)$ is the inverse rotation matrix, and $\Omega_k \equiv (\psi_k, \theta_k, \phi_k)$ is solid angle for the kth reactant; $d^{(k)}$ is the length of vector $\mathbf{d}^{(k)}$.

(4) Since the internal geometries of the reactants are fixed at the input ones, and their center of mass positions are determined in step 3, the coordinates of the generalized transition states are simply obtained by a translation.

The quantity $N_q(E,J,Q,s)$ can be loosely viewed as number of accessible states for a specific configuration Q with total energy E and angular momentum $J\hbar$, which can be expressed as

$$N_q(E,J,Q,s) = \Phi J^2 \mu s^2 \frac{8\pi^{k_a-1}}{\Gamma(v/2-1)} \frac{\displaystyle\prod_{k=1}^{2}\prod_{i=1}^{v_k}\sqrt{2\pi I_i^{(k)}}}{\displaystyle\prod_{i=1}^{3}\sqrt{2\pi I_i(Q)}}$$

$$\times \left\langle \max\left(E - V(Q) - \sum_{i=1}^{3}\frac{J_i^2}{2I_i(Q)}, 0 \right)^{\frac{v}{2}-2} \right\rangle_{\Omega_J} \quad , \qquad (7)$$

$$\Phi = \sqrt{1 + \mu\sum_{k=1}^{2}\sum_{i=1}^{v_k}\left(\mathbf{n}^{(12)}\times\mathbf{d}^{(k)}\cdot\mathbf{n}_i^{(k)}\right)^2 / I_i^{(k)}} \quad , \qquad (8)$$

where $\langle\cdots\rangle_{\Omega_j}$ denotes the averaging over all possible orientations of the vector of angular momentum $\mathbf{J}$ while its length is fixed. In equation (7), Φ is a kinematic factor given by Equation (8); v equals $v_1 + v_2 + 3$ where v_k ($k = 1,2$) is the dimensionality of the orientational space for the kth reactant; $I_i^{(k)}$ denotes the ith moment of inertia of the kth reactant, and I_i is the ith moment of inertia for the system as a whole; $\mathbf{n}^{(12)}$ is the unit vector directed from the second pivot point to the first one; $\mathbf{d}^{(k)}$ is the vector connecting the center of mass of the kth reactant to its pivot point; $\mathbf{n}_i^{(k)}$ is the unit vector directed along the ith principal axis of the kth reactant; k_a is the number of monatomic reactant, which should be equal to 0 or 1.

Although Equation (8) for the kinematic factor Φ is valid for both linear and nonlinear molecules, it is convenient to rewrite the term for linear molecule in a different form:

$$\mu I^{-1} \sum_{i=1}^{2} (\mathbf{n}^{(12)} \times \mathbf{d} \cdot \mathbf{n}_i)^2 = \mu I^{-1} \left[1 - (\mathbf{n}^{(12)} \cdot \mathbf{n}_0)^2 \right], \tag{9}$$

where $\mathbf{n}_0$ is the unit vector directed along the axis of the molecule; I is the moment of inertia.

There are three ways to perform variational calculations. They correspond to optimizing the transition state dividing surface using three different ensembles. The resulting rate constants are labeled canonical variational theory (CVT), microcanonical variational theory (μVT), and energy and total angular momentum resolved microcanonical variational theory (E,J -μVT), respectively. We now discuss these three choices:

(i) Energy and total angular momentum resolved microcanonical variational theory:

For E,J -μVT theory, $N(E,J)$ is minimized with respect to s for each E and J. Then the integral in Equation (1) is calculated using the optimized $N(E,J)$.

The minimum of the number of accessible states is obtained by interpolation from a quadratic fit to $N(E,J,s)$ at the three values of s nearest to the minimum. We call this method a 3-point fit; 3-point fits are also used in μVT and CVT. The integrals over E and J for the optimized $N(E,J)$ in Equation (1) are approximated by Gauss-Laguerre and Simpson's rule quadratures, respectively.

(ii) Microcanonical variational theory:

For microcanonical variational theory (μVT), the integral over J in Equation (1) is carried out first to get $N(E,s)$. By integrating Equation (7) over J and substituting the result into Equation (2), the number of accessible state $N(E,s)$ is obtained as the following expression:

$$N(E,s) = \mu s^2 \frac{2\pi^{k_a - 2}}{\Gamma((v+1)/2)} \prod_{k=1}^{2} \prod_{i=1}^{v_k} \sqrt{2\pi I_i^{(k)}} \times \left\langle \Phi \max(E - V(Q), 0)^{(v-1)/2} \right\rangle_{\Omega} \tag{10}$$

Then $N(E,s)$ is minimized with respect to s for each E, and the optimized $N(E)$ is used for the integral over E.

(ii) Canonical variational theory:

For canonical variational theory (CVT), $N(E,J,s)$ is calculated for all E and J at a single s value. Then the rate constants $k(T,s)$ is minimized with respect to s for each T. The analytic result of the integral in Equation (1) is expressed as:

$$Z(T,s) = \int dE e^{-E/kT} dJN(E,J,s)$$
$$= 2\pi^{k_a-2}\mu s^2 (kT)^{(\nu+1)/2} \prod_{k=1}^{2}\prod_{i=1}^{\nu_k} \sqrt{2\pi I_i^{(k)}} \left\langle \Phi e^{-V(Q)/kT} \right\rangle_\Omega \qquad (11)$$

11.B. Multifaceted dividing surface

When one uses single-faceted dividing surfaces, two pivot points (with one on each of the associating reactants) and a fixed distance between them form a spherical dividing surface. But in many cases the reactants of association reactions have multiple binding sites. Multiple binding sites on one or both reactants require more pivot points to increase the flexibility of the shape of the dividing surface. Let's assume that n_1 pivot points are assigned to reactant 1 and n_2 pivot points are assigned to reactant 2, and the reaction coordinate value is s. Therefore there are $n_1 \times n_2$ pairs of pivot points. Each pair of pivot points corresponds a sphere, and the points on each sphere are selected according to the following condition to form a composite dividing surface:

$$\min_{i=1, j=1}^{n_1, n_2} r_{i,j} = s \qquad (12)$$

where $r_{i,j}$ is the distance between the ith and jth pivot points associated with the first and the second reactant, respectively; s is the value of reaction coordinate. The meaning of Equation (12) is that at one particular configuration of the complex at any of the spheres formed by the $n_1 \times n_2$ pairs of pivot points, if the minimum of the distances between pivot points associated with different reactants equal to the value of reaction coordinate, this configuration will be accepted in Monte Carlo sampling. Therefore the composite dividing surface has $n_1 \times n_2$ elementary surfaces; each of the elementary surfaces is called a facet, and the composite surface is called a multifaceted (MF) dividing surface.

In variational calculations, various locations of pivot points should be considered to minimize the reaction rate. Users are responsible to try various sets of locations of pivot points; that is, *Polyrate* has not automated this. The optimal locations of pivot points are the locations that minimize the reaction rate; but the user must be careful that the MF

dividing surface defines by the chosen locations of the pivot points separates reactants from products.

11.C. Overview of program flow for VRC-VTST calculation

When the VRC method is used for loose transition states by turning on the VRC keyword in the GENERAL section, the program will not use the MEP as the reaction path, but will instead use variable reaction coordinates and either single-faceted or multifaceted dividing surfaces. After the MAIN program reads all the information provided at reactants, subroutine VRCTST is called. Note that user gives the information about two reactants in this case, and the START section only need the ELEC keyword for electronic degeneracy information. If only the forward reaction rate constants for association need to be calculated, the product information is not required. In VRC-VTST calculations, the MAIN program and subroutine VRCTST are parallelized using MPI, and they work independently on different processors to read the same fu5 input file and perform Monte Carlo sampling. During the Monte Carlo sampling, only energies are calculated (for various geometries along the reaction path), and gradients and Hessians are not required. When the Monte Carlo sampling is finished on all processors, one processor (rank 0 in MPI) is used to collect the data calculated on all the processors and to perform the rest of the calculations to get reaction rate constants.

11.D. Implementation details of the *E, J*-resolved microcanonical VRC algorithm

This section contains material from the Supporting Information Appendix of ref. [75] of Sect. 20.

We here describe the implementation details of the E, J-resolved microcanonical VRC algorithm for the case of two pivot points on fragment 1 and two on fragments 2; the generalization to a larger number of pivot points is straightforward. We denote the pivot points on fragment 1 as point 1 and point 2 and the pivot points on fragment 2 as point 3 and point 4. The range of reaction coordinate s searched for a variational transition state is from s_l to s_u ("l" stands for lower edge of the range, and "u" stands for upper edge of the range), and the search for a variational transition state is carried out by varying s with increments of size s_{step}.

We define the vector *s* as a vector connecting two pivot points, one in each fragment. This is illustrated in Scheme S1, where we are considering a particular pair of pivot points, namely pivot point 1 of fragment 1 and pivot point 3 of fragment 2, so we only show the vector *s* that connects these two pivot points, and only these two pivot are shown (other pivot points on fragments 1 and 2 are not shown).

In the computer code, we loop over the pairs of pivot points with one on each fragment (1-3, 1-4, 2-3, 2-4), and for each pair of pivot points, the vector *s* is defined as the vector that connects the pivot point on fragment 1 with the pivot point on fragment 2.

The magnitude of **s** is denoted as s; this is the same as the reaction coordinate s mentioned in the first paragraph.
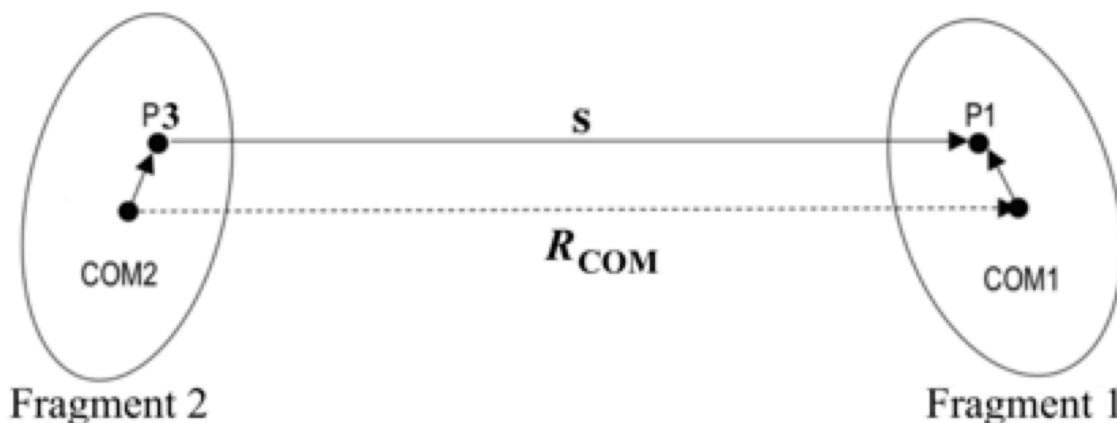
Scheme S1

Step 1: Set the length of the reaction coordinate s to s_1.

Step 2. Do for *m* from 1 to 4. For each m, we consider a pair of pivot points with one on each fragment. Index $m = 1$ corresponds to pair 1-3, $m = 2$ represents the pivot points pair 1-4, $m = 3$ represents the pivot points pair 2-3, $m = 4$ represents the pivot points pair 2-4. For our explanation we will use $m = 1$ as an example; in particular, $m = 1$ corresponds to choosing pivot point 1 and pivot point 3. For case $m = 1$ or 2, P1 is pivot point 1, and for $m = 3$ or 4, P1 is pivot point 2. For the case $m = 1$ or 3, P3 is pivot point 3; and for $m = 2$ or 4, P3 is pivot point 4.

Each performance of Step 2C-1 is a Monte Carlo sample. The number of Monte Carlo samples per value of m is denoted as N_{MC}, which is a number specified by the user, and which is the same notation as used in the input and the code. That is, for each m, there are N_{MC} Monte Carlo samples. Because each pair of pivot points determines one facet of the dividing surface, and we have four pairs (i.e., we have a four-faceted dividing surface), the total number of Monte Carlo samples is $4N_{MC}$. In eq 12 of *J. Phys. Chem. A* **2003**, *107*, 9776, which will be used in Step 4, the variable M is equal to $4N_{MC}$.

Step 2A for this pair *m*: Put fragment 1 in a specific *initial* orientation (this starting orientation is determined randomly) with respect to P1; set the vector **s** (as shown in Scheme S1) in a specific *initial* direction (this direction is also determined randomly as a start direction) so that P3 position has been decided; and then put the fragment 2 in a specific *initial* orientation (this starting orientation is determined randomly) with respect to P3. Therefore the coordinates of the whole system and other pivot points are determined. Calculate the length of the vector R_{COM} that connects the centers of mass of the fragments.

Step 2B for this pair *m*: **Do for *i* from $(m-1)N_{MC}+1$ to $(m-1)N_{MC}+N_{MC}$.** Starting from the *initial* configuration determined in Step 2A, we now start doing Monte Carlo sampling, i.e., randomly orientating the two fragments, while keeping the length of vector **s** as specified in Step 1. When orientating the fragments, P1 is *always* taken as the origin of the coordinates system. Then, the position of P3 is determined by a random solid angle. After reorienting s, also reorient fragment 2 (the solid angles for these

reorientations are determined randomly). This now yields configuration i.

Step 2C for this i:

Step 2C–1: Fragment 1 has pivot points 1 and 2, and fragment 2 has pivot points 3 and 4, and the distances between each of the various pairs of pivot points (with one on each fragment) can be computed. Compute the distance r_{ij} (where $i = 1, 2$, and $j = 3, 4$) for each pair of pivot points (1-3, 1-4, 2-3, 2-4) for current Monte Carlo sample configuration. If any of the four r_{ij} is greater than or equal to s (no matter which pair of the pivot points satisfies this condition), then go to Step 2C-2, otherwise abandon this configuration. If this configuration is abandoned, it does not count in M' in eq. 13 of *J. Phys. Chem. A* **2003**, *107*, 9776; if this configuration is accepted, then it increases the value of M' by 1.

Step 2C–2: For the specific configuration accepted in Step 2C-1, compute the moment of inertia of each fragment and the unit vector along its principal axis; and calculate the kinematic factor Φ (eq. 2.40 in *J. Chem. Phys.* **2003**, *118*, 5442). Next compute number of states $N_q(E,J,s,i)$ for this value of the reaction coordinate s and this specific specific configuration i, and for each combination of total energy E and angular momentum vector $\boldsymbol{J}$ (the length of which is denoted as J), averaged over all the possible orientations of vector $\boldsymbol{J}$ while its length is fixed; this is eq 9 of *J. Phys. Chem. A* **2003**, *107*, 9776, and it is computed at using eq 2.39 in *J. Chem. Phys.* **2003**, *118*, 5442. Note that we included the correction factor of 2π that is indicated in reference 12 of *J. Phys. Chem. A* **2003**, *107*, 9776.

Step 2D: If you have not yet done N_{MC} samples for this m, go back to Step 2B, except start with the present configuration rather than the initial one. If you have completed this m, go to step 2E.

End do loop over index i.

Step 2E: If m is less than 4, increment m by 1 and carry out Steps 2A to 2D for the next pair of pivot points.

End do loop over index m.

Step 3. Now compute the generalized transition state approximation (for a transition state at reaction coordinate value s) of the number of states for the transitional modes at total energy E and total angular momentum J. This is given by

$$N^{GT}(E,J,s) = \frac{1}{M}\sum_{i=1}^{M'} N_q(E,J,s,i) \tag{13}$$

Step 4: If $s < s_u$, increase the length s of vector $\boldsymbol{s}$ by s_{step}, and go back to Step 1. After all the values of s specified by the user have been considered, go to Step 5.

Step 5: In E,J-resolved microcanonical variational theory, for each E and J, minimize $N(E,J,s)$ with respect to s. This yields

$$N^{\mu VT}(E,J) = N^{GT}(E,J,s_*) \tag{14}$$

where s_* is the value of s that minimizes the number of states. Then plug $N^{\mu VT}(E,J)$ into the rate constant expression and evaluate the integral to obtain the final rate constant $k(T)$.

PART IV DESCRIPTION OF INPUT FILES

12. DESCRIPTION OF INPUT FOR RP-VTST

In the following subsections, a record-by-record description of the data for each input file that is used for RP-VTST calculations is given:

      Subsection 12.A: the input data needed from unit fu5;

      Subsection 12.B: the input data needed from unit fu29;

      Subsection 12.C: the input data needed from unit fu30;

      Subsection 12.D: the input data needed from unit fu31;

      Subsection 12.E: the input data needed from unit fu40;

      Subsection 12.F: the input data needed from unit fu50;

      Subsection 12.G: the input data needed from unit fu51.

12.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 12.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 12.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).

12.A.1. Section descriptions

The following input sections correspond to the major parts of a RP-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.

Section Name	Description
*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

12.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:

Keyword	Type	Default	Description
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	`none`	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	`seckart`	Eckart function fit to $V_{\text{MEP}}(s)$
FREQ	variable	`icl`	interpolated correction of frequency
IVTST0	switch	`off`	zero-order IVTST
IVTST1	switch	`off`	first-order IVTST
INPUNIT	variable	`angstrom`	specifies the units in fu5 input file
OUTUNIT	variable	`angstrom`	specifies the units in fu6 and fu14 output files
MDMOVIE	switch	`off`	write MDMOVIE input file

Basic:

Keyword	Type	Default	Description
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
SUPERMOL	switch	on	supermolecule mode for reactants and products
WRITE62	variable	none	write information in the fu62 file
WRITEFU30	switch	off	write an fu30 file
WRITEFU31	switch	off	write an fu31 file

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.

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 12.F. This keyword can also be used to perform IVTST-0 calculations with unit fu50; see Section 6.N. The valid options are

Option	Description
ioc	dual-level using fu50 (VTST-IOC)
ispe	dual-level using fu51 (VTST-ISPE)
none	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.

Option	Description	Default
both	separate calculations will be based on the corrected and uncorrected data	off
correct	calculation will be based only on the corrected data	on
energy	interpolated optimized corrections for energies:	*seckart*
	deckart - difference of two Eckart functions fit to the $V_{MEP}(s)$ curve at higher level and Eckart function fit to lower level	
	seckart - 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	
freq	interpolated optimized corrections for frequencies:	*icl*
	ica - interpolated correction based on arithmetic differences	
	icl - interpolated correction based on logarithm of ratios	
	icr - interpolated correction based on ratios	
	none - no corrections	

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

INPUNIT

INPUNIT is a variable keyword that specifies the units for the geometries and distances given in the fu5 input file. This keyword will have no effect on selecting the units for the geometries and variables for distances in other input files (units fu29, fu30, fu31, fu40, or fu50). The default is *ang*.

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

    Example:
        INPUNIT   AU

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.

OUTUNIT

OUTUNIT is a variable keyword that specifies the units for the geometries and distances given in the fu6 and fu14 output file. The default is *ang*.

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

    Example:
        OUTUNIT   AU

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.

Option	Definition
writefu1	write restart information to unit fu1 for future use
readfu1	restart calculation using information on unit fu1
readfu1&2	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.
writefu3	the merged MEP from *readfu1&2* is written to unit fu3
merge x	increment in s to be added to the s values of MEP #2 when it is merged with MEP #1. This is required if the READFU1&2 option is used. The default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU.
writefu30	write an fu30 file
writefu31	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_o 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

Option	Description
none	no information is printed in fu62
viball	frequencies (in cm^{-1}) are printed for all stationary points
vibsp	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

WRUTEFU31 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.

12.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.

Keyword	Type	Default	Description
EZERO	variable	*calculate*	method for determining zero of E
EZUNIT	variable	*a.u.*	units of zero of energy
POTENTIAL	variable	*hooks*	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`.

Option	Definition
`calculate`	If `calculate` is chosen the program will set EZERO to the value that it calculates for the energy of the reactants.
`read #`	The `read #` 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.

Option	Definition
`a.u.`	atomic units (hartrees)
`kcal`	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 12.C, 12.D, and 12.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.

Option	Definition
hooks	use the hooks
unit29	read in from electronic structure file fu29
unit30	read in from electronic structure file fu30
unit31	read in from electronic structure file fu31
unit40	read in from electronic structure file fu40

Example:
```
POTENTIAL unit30
```

12.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:

Keyword	Type	Default	Description
HESSCAL	variable	*ghook*	how to calculate Hessians

Second derivative *Polyrate* methods:

Keyword	Type	Default	Description
Other:			
FPRINT	switch	off	print second derivative information
NUMSTEP	variable	1×10^{-4} au	step size for 2nd derivatives of V
NUMTYPE	variable	*Central*	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 in fu6 output file. 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.

Option	Definition
ghook	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.
hhook	call HHOOK to get second derivatives.

Example:
```
   HESSCAL    ghook
```

NUMSTEP

NUMSTEP is a variable keyword giving the step size in unscaled angstrom (default) or in unscaled bohr (by specifying the keyword INPUNIT AU) 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} bohr.

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:

Option	Definition
central	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.
quadratic	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.
fourth	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*

12.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:

Keyword	Type	Default	Description
OPTMIN	variable	*bfgs*	how to optimize minima
OPTTS	variable	*nr*	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:**

Keyword	Type	Default	Description
Common:			
PRINT	switch	`off`	print geometrical parameters and derivatives after each opt step
Other:			
DLX2	variable	*1 x 10^{-5} au*	step size for first derivative of grad in geometry optimization
EFOPT	list	`off`	indicate the option for EF
GCOMP	variable	*1 x 10^{-5}*	maximum allowed component of the gradient for convergence at the minima
TSGCOMP	variable	*1 x 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 x 10^{-5} au*	"optimization failure" parameter

Glossary of OPTIMIZATION keywords

DLX2

DLX2 is a variable keyword used to specify the step size in unscaled angstrom (default) or in unscaled bohr (by specifying the keyword INPUNIT AU) for calculating the first derivatives of the gradients of the potential energy function in geometry optimization. The default value is 1×10^{-5} a_o.

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 1×10^{-5} E_h/a_o 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.*

Option	Definition
bfgs	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.)
ef	specifies to use the Eigenvector Following algorithm.
nr	specifies to use the Newton-Raphson algorithm with Brent line minimization.
ohook	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.*

Option	Definition
bfgs	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.)
ef	specifies to use the Eigenvector Following algorithm.
nr	specifies to use the Newton-Raphson algorithm with Brent line minimization..
ohook	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 in fu6 output file. 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. The default unit is angstrom and can also be bohr by specifying the keyword INPUNIT AU. 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 1×10^{-5} a_0.

> *Example:*
> STPTOL 0.0001

12.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:

Keyword	Type	Default	Description
Required:			
GEOM	list	`required`	Cartesian coordinates of atoms
Common:			
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
Other:			
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:

Keyword	Type	Default	Description
Common:			
ANTLR	variable	1×10^{-8}	criterion on Morse x_e
DEMIN	variable	`0.159`	lowest dissociation energy (D)
DQQP	variable	`0.0001 au,` `0.001 au`	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	`morsei`	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. points 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 $x_e <$ 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 Cartesians 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 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. β, and D_e are in atomic units and R_e is in angstrom by default and also can be in bohr by specifying the keyword INPUNIT AU. 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 used when fitting the quadratic-quartic potential to two points along the normal coordinate. It is in angstrom by default and also can be in bohr by specifying the keyword INPUNIT AU. 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 1×10^{-4} a_o and 1×10^{-3} a_o.

> *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
> $$L_{11} \quad L_{21} \quad L_{31} \quad L_{41} \quad L_{51} \quad L_{61}$$
> $$L_{12} \quad L_{22} \quad L_{32} \quad L_{42} \quad L_{52} \quad L_{62}$$
> $$L_{13} \quad L_{23} \quad L_{33} \quad L_{43} \quad L_{53} \quad L_{63}$$
> $$L_{14} \quad L_{24} \quad L_{34} \quad L_{44} \quad L_{54} \quad L_{64}$$
> $$L_{15} \quad L_{25} \quad L_{35} \quad L_{45} \quad L_{55} \quad L_{65}$$
> $$L_{16} \quad L_{26} \quad L_{36} \quad L_{46} \quad L_{56} \quad L_{66}$$
> END

> *Explanation of example:*
> L_{ij} is the jth component of the ith 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 ^{3}P 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 ≥ 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*.

Option	Definition
waven	Frequencies are given in cm^{-1}.
au	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 that is a guess to the species' stationary-point geometry. IF status is ≥ 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. It is in angstrom by default and also can be in bohr by specifying the keyword INPUNIT AU (not mass-scaled).

 Example:
```
    GEOM
        1    0.000000    0.000000 0.000000
        2   -0.241497   -0.945827 0.000000
        3    0.956805    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
> ```
> <u>*Explanation of the example:*</u>
> The element $F_{jj'}$ of the Hessian matrix **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:

Option	Definition
geom	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.
hooks	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:

Option	Definition
x-axis	the linear molecule is aligned with the x-axis
y-axis	the linear molecule is aligned with the y-axis
z-axis	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):

Option	Definition
morsei	Morse I: V_{MEP} + local D = DEMIN
morseii	Morse II: local D from V'''
morseiii	Morse III: Morse II if (V_{MEP} + local D) > DEMIN; otherwise Morse I
morseia	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.
morseiiia	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.

Option	Definition
atomic	species is a single atom
linrp	species is a linear reactant or product
nonlinrp	species is a nonlinear reactant or product
ssrp	species is a solid-state reactant or product
lints	species is a linear transition state or generalized transition state
nonlints	species is a nonlinear transition state or generalized transition state
ssts	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 FIRSTEP 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. Note that the MS-T method is not included in this keyword list, but MS-T can be specified in the PATH section using the SSTOR keyword. 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 SRPG is the recommended method when using the TOR keyword. Other methods may be appropriate for certain systems, but careful analysis should accompany any other choice. See Ref. 52 and 62 of Sect. 21 for a study and comparison of torsion methods.

The input required for each hindered rotator mode is:

          MODE   P   M   METHOD   N

where

          MODE   =   mode number m
          P   =   number P of distinct minima along the torsional coordinate
          M   =   number M of total minima along the torsional coordinate $(M \geq P)$
          METHOD = one of six methods defined in Section 10.3:
                    CT   Chuang-Truhlar interpolatory method
                    RPG   Torsional Pitzer-Gwinn method
                    SRPG   Segmented Torsional Pitzer-Gwinn method
                    AS   Ayala-Schlegel method
                    SAS   Segmented Ayala-Schlegel method
                    FR   Free Rotor
          N   =   number N of atoms in first subgroup

The user needs to be cautious with the definition of number of distinct minima along the torsional coordinate. For example, H_2O_2 has two minima that have the same energy and frequencies, but they are mirror images of one another, and one cannot be superimposed to the other. Therefore they are distinguishable. The total symmetry number is the ratio of M/P, which should not be a non-integer number. Here are a few examples for the values of P and M.

Molecule	P	M
CH_3-CH_3	1	3
$ClCH_2$-CH_3	1	3
HO-OH	2	2
$ClCH_2$-CH_2Cl	3	3
$ClCH_2$-CH_2	3	6

Among the six methods for torsional treatment, SRPG is recommended. Note each method requires a different set of keywords. For more input information see the TOROPT section below.

> *Example:*
> ```
>       TOR
>           10  1     3      SRPG 4
>       END
> ```

TOROPT

TOROPT is a list keyword that must be given for modes treated as hindered rotors . 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.

Which quantities may appear in the list of values depends on DATATYPE. The allowed values for DATATYPE and the associated lists are as follows:

DATATYPE	list
ISB	i_1, i_2, ..., 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)
CTSCHEME	OW/RO/CO/RW/CW/RWO/CWO (one of seven schemes defined in Ref. 52 of Sect. 21 for CT method)
CTLEVEL	FULL/SF/SC (one of three levels for CT method)
OMEGA	ω_2, ..., ω_P (the frequencies in cm^{-1})
U	U_2, U_3, ..., U_P (excitation energies in cm^{-1})
I	I_2, I_3, ..., I_P (moments of inertia in $m_e a_0^2$)
READI	I_1 (moment of inertia in $m_e a_0^2$)
W	W_1, W_2, ..., W_P (barrier heights in cm^{-1})
WL	WL_1, WL_2, ..., WL_P (WL_j is the barrier height from the minimum j to the maximum on its left in cm^{-1})
WR	WR_1, WR_2, ..., WR_P (WR_j is the barrier height from the minimum j to the maximum on its right in cm^{-1})
RATIOL	RL_1, RL_2, ..., RL_p (RL_j is the torsional angle difference between the minimum j and the maximum on its left in degrees)
RATIOR	RR_1, RR_2, ..., RR_p (RR_j is the torsional angle difference between the minimum j and the maximum on its right in degree)

The methods specified in the TOR section need different DATATYPE values and the associated lists. Note that ISB and NBOND are always required except when CTSCHEME = OW or READI is used. An explanation of DATATYPE requirement for each method is given below

1.   The RPG and AS methods require the following DATATYPE values:

ISB, NBOND, I, WL, and WR

In these methods the moment of inertia of the torsional motion I_1 is calculated by the curvilinear model using the input geometry in fu5 file, and the others are given by the user. The user can run jobs to obtain $I_2, \ldots, I_p$ using the corresponding geometries. The barrier height used in the RPG and AS methods is taken as the average of the input WL and WR values. The frequency of the lowest minimum is used.

2.   The SRPG and SAS methods require the following DATATYPE values:

ISB, NBOND, I, U, WL, WR, RATIOL, and RATIOR

The SPRG and SAS methods need the same information as their corresponding non-segmented counterparts except that U, RATIOL and RATIOR are also required.

3.   The FR method require the following DATATYPE values:

ISB, NBOND, and I

4.   The CT method requires the following DATATYEP values:

ISB, NBOND, CTSCHEME, CTLEVEL, OMEGA, U, and I.

CTSCHEME list is one of the seven schemes defined in Ref. 52 of Sect. 21:
OW   Scheme ωW
RO   Scheme Rω
CO   Scheme Cω
RW   Scheme RW
CW   Scheme CW
RWO   Calculate I with the R-scheme, read in W, obtain ω from normal mode or generalized normal-mode analysis
CWO   Calculate I with the C-scheme, read in W, obtain ω from normal mode or generalized normal-mode analysis (default)

CTLEVEL list is one of the three approximation levels used for CT method defined in Ref. 52 of Sect. 21:
FULL   full (default)
SF   single-frequency
SFE   single-frequency and single-energy

Note that the SFE method treats the P minima using the same I, ω, and W as the input geometry in fu5 file and do count the contributions of the P minima to the partition function. The single-conformer (SC) method defined in Ref. 52 of Sect. 21 is a special case of SFE method when P equals 1.

In the seven distinct schemes, R represents using the rectilinear model to calculate the reduced 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 reduced moments of inertia of the torsional motion (Ref. 59 in Sect. 22 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. The first five schemes use the relationship given by Eq. (12) of Ref. 52 in Sect. 21 to calculated either I, W, or ω from the other two. The RWO and CWO schemes require I, W, and ω, and use these variables in the equations of Ref. 52 in Sect. 21 without forcing a relationship between them.

If CTSCHEME = CO or Cω, then the sum of the masses of the atoms in ISB must be less than or equal to 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 RW schemes is used.

The set of data types expected in the TOROPT list for CT method depend on the choices of CTSCHEME and CTLEVEL and the number of distinguishable minima P. In particular, the expected records are:

Choices in   DATATYPES expected in TOROPT

P	CTLEVEL	CTSCHEME	Other expected DATATYPES
1	any	OW	W (W_1 only)
1	any	RO	ISB
1	any	CO	ISB, NBOND
1	any	RW	ISB, NBOND, W (W_1 only)
1	any	CW	ISB, NBOND, W (W_1 only)
>1	FULL	OW	U, OMEGA, W
>1	FULL	RO	ISB, OMEGA, U, I,
>1	FULL	CO	ISB, NBOND, OMEGA, U, I,
>1	FULL	RW	ISB, U, I, W
>1	FULL	CW	ISB, NBOND, U, I, W
>1	SF	OW	U, W
>1	SF	RO	ISB, U, I
>1	SF	CO	ISB, NBOND, U, I
>1	SF	RW	ISB, U, I, W (W_1 only)
>1	SF	CW	ISB, NBOND, U, I, W (W_1 only)
>1	SC	OW	W (W_1 only)
>1	SC	RO	ISB
>1	SC	CO	ISB, NBOND
>1	SC	RW	ISB, W (W_1 only)
>1	SC	CW	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 ω_j = frequency obtained from normal mode analysis at point on reaction path (j = 1, 2, …, P), W_j = 1000 cm^{-1} (j = 1, 2, …, P), U_j = 0, ISBN = {1 2 3 4}, and NBOND = {4 5}.

The READI option is used to read in the moment of inertia for the initial configuration.

Example for using CT method with OW and FULL options:

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

Example for using CT method with CW and SC options:

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

Example for using SRPG method:

```
    TOR
      6  2    2     SRPG 2
END
    TOROPT
        6    ISB       1  4
        6    NBOND     1  2
        6    WL          2545.505 376.905
      6  WR        376.905  2545.505
      6  RATIOL    112.504 67.496
      6  RATIOR    67.496  112.504
```

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 FREQUNIT) 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 default unit of *s* is angstrom and can also be bohr by specifying the keyword INPUNIT AU.

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 $0.01 \le s \le 0.5$ and $-0.15 \le s \le -0.01$ 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 12.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:

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

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

12.A.7. PATH section

The PATH section contains the keywords that control the calculation of the reaction 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:

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

Basic:

Keyword	Type	Default	Description
Others:			
FREQSCALE	Variable	*1*	Scale factor for frequencies
HESS	Switch	`on`	Compute the Hessian along MEP
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` *au*	step size for 10 extra points
TENSION	variable	`-1.0`	tension for spline under tension
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
SCALERP	switch	`off`	scales frequencies in VIB section if turned on
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
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:

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

Step sizes for the MEP:

Keyword	Type	Default	Description
Common:			
SSTEP	variable	5×10^{-4} *au*	step size for computing MEP
Others:			
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} *au*	step size for the special steps

Reaction coordinate for the VRC:

Keyword	Type	Default	Description
Common:			
SVRC	list	not used	Reaction coordinate setting for reaction path in variable reaction coordinate algorithm for loose transition state

Extrapolation:

Keyword	Type	Default	Description
Others:			
EXFIRST	list	`not used`	extrapolate first direction from saddle point.
EXNSTEP	variable	*0*	number of steps
EXREACT	switch	`off`	extrapolation is towards reactants
EXPROD	switch	`on`	extrapolation is towards products
EXSTEP	variable	*0.03 au*	step size
EXALPHA	variable	*1.0*	extrapolation range parameter
EXSECOND	list	`not used`	extrapolate second direction from saddle point
EXNSTEP	variable	*0*	number of steps
EXREACT	switch	`on`	extrapolation is towards reactants
EXPROD	switch	`off`	extrapolation is towards products
EXSTEP	variable	*0.03 au*	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:

Keyword	Type	Default	Description
Others:			
POTINF	variable	*0*	print *n* 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 geometry, V, and V' at each save point
PRSAVEMODE	variable	9999	print normal mode information at each save point
SDEBG1	variable	not used	*s* value for starting debug printing of generalized normal mode vibrational analysis along the MEP
SDEBG2	variable	not used	*s* value for ending debug printing of generalized normal mode vibrational analysis along the MEP
SPRNT	variable	*0*	extra reaction-path Hamiltonian information printed at large *s*
SSPECPR	switch	off	print information at special *s* values

Nonequilibrium Solvation:

Keyword	Type	Default	Description
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} cm^2s^{-1}
FRICTION	variable	*0*	solvation time in fs

Torsional Anharmonicity

Keyword	Type	Default	Description
SSTOR	list	not used	MS-T torsional anharmonicity along the reaction path by carrying out a single-structure (SS) calculation as described in Section 10.O.
MTOR	variable	*no default*	local periodicities of torsions

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:

Keyword	Description
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} cm^{2}s^{-1}
FRICTION	solvent friction time in fs (Note: the friction time is also called the solvation time)

Example:
```
BATH
   FRICTION   10.0    # in fs
   DIFFUSION           # in 10-5 cm^2s^-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 the 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. It is very important to use curvilinear coordinates rather than Cartesian coordinates for generalized normal mode analysis along MEP in VTST calculations; Cartesian coordinate will very likely yield imaginary frequencies along the reaction coordinate, which will cause unphysical VTST results. See Sections 6.G.1 and 10.M. The default is *cart* (Cartesian coordinates). The valid options are:

Option	Definition
curv1	use curvilinear internal coordinates for generalized normal mode analyses using method of Ref. 41 of Sect. 21
curv2	use curvilinear internal coordinates for generalized normal mode analyses using method of Ref. 43 of Sect. 21
curv3	use redundant internal coordinates for generalized normal mode analyses using method of Ref. 46 of Sect. 21
cart	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            curv2
```

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:

Option	Definition
dgrad	use a quadratic fit to obtain the derivative of the gradient with respect to the reaction coordinate
oneside	use one-sided differences for the derivative of the gradient with respect to the reaction coordinate
dhess	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 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 unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU. The default is 1×10^{-4} a_o.

    Example:
```
    DLX3                0.00005
```

ES1OPT

ES1OPT is a list keyword for using the Euler with stabilization integrator. The user must give the step size 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 default unit of step size DELTA2 is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU. The defaults are DELTA2 $= 5 \times 10^{-5}$ a_o and 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 $\mu_{eff}(s)$ 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.

Keyword	Description	Default
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 the extrapolation region for computing the adiabatic potential and other quantities needed for the tunneling calculations. The default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU.	0.03 a_o
EXALPHA	exponential range parameter α in reciprocal angstroms (only used with NOSADDLE). It can also be in reciprocal bohrs by the keyword INPUNIT AU.	1.0 a_o^{-1}

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.

Option	Description	Default
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 the extrapolation region	$0.03\ a_o$
	for computing the adiabatic potential and other quantities needed for the tunneling calculations The default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU.	
EXAPLHA	exponential range parameter α in reciprocal angstroms. It can also be in reciprocal bohrs by the keyword INPUNIT AU.	$1.0\ a_o^{-1}$

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:

Option	Definition
nmode	take the first step along the imaginary-frequency normal mode
cubic	use the cubic starting algorithm (see STEP 3)
gradi	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. For kinetics, the scale factor that one should choose is the one for reproducing the correct ZPE, i.e., the ZPE scale factor, which is available on the website: https://comp.chem.umn.edu/freqscale/version3b2.htm.

Please notice that, if the vibrational frequencies are directly read in from the .dat input file, the FREQSCALE will *not* be applied on these user-input vibrational frequencies; and therefore, usually, the user-input frequencies in .dat file should have already been scaled by the user with the same frequency scale factor for the reaction path. This feature thus also provides the user with an opportunity to use different scale factors for stationary points and for the saddle point and the non-stationary points on the reaction path.

Example:
   FREQSCALE   0.91

FREQINCR

FREQINCR is a variable keyword for specifying the value to which the frequencies that are lower than this value to increase (unit is cm^{-1})

Example:
   FREQINCR   200

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. (It is very important to use curvilinear coordinates rather than Cartesian coordinates for VTST calculations; Cartesian coordinate will very likely yield imaginary frequencies along the reaction coordinate, which will cause unphysical VTST results.) 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 (#) cannot 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:

Option	Definition
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.

Option	Definition
`lbexp`	value of the exponent for interpolating the curvature components. The default is 3.
`polys`	use a cubic polynomial for estimating 10 energy points before spline-fitting $V_{\mathrm{MEP}}(s)$.
`polyz`	use a cubic polynomial for estimating 10 energy points before spline-fitting $V_{\mathrm{MEP}}(z)$ (default).
`nopolys`	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.
`sincw`	variable keyword to indicate the step-size along the lowest-frequency mode to fit the cubic polynomial. The default unit is mass-scaled angstroms and it can also be in mass-scaled bohr by specifying the keyword INPUNIT AU. The default value is 0.01 a_0. this value is used only if `polys` or `polyz` is selected.

fixmuef	keyword 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) one 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.
tension	keyword used to specifically give a tension factor in interpolation using spline under tension. If this keyword not used, the program will automatically determine the tension factors. The value for this keyword must be between 0 an 85. We recommend to use this keyword only in case very large tension factor is needed.

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:

Option	Definition
index	index of the vibrational mode to be interpolated (this is determined from the frequencies of the saddle point arranged in decreasing order).
reactant x	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.
start x	used to indicate the frequency at $s = 0$ for the interpolation.
product x	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.
sr x	used to indicate the s value at the limit of reactant side ($s < 0$). The default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU. The default value is -1.0 bohr.
sp x	used to indicate the s value at the limit of product side ($s < 0$). The default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU. The default value 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

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

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

where $\mathbf{x}_{\text{current}}$ is the mass-scaled geometry corresponding to the last point input on fu30
or fu40, and $\hat{g}_{current}$ 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 in fu6 output file. 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 Supercomputing 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.

Option	Description	Default
coord 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
interval n	write the reaction-path information every n Hessian grid points.	1
freq 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
xmol	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.

Option	Description
esd	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
es1	use the Euler-with-stabilization integrator to follow the reaction path
pagem	use the Page-McIver integrator to follow the reaction path
vrpe	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. 22.) The variable μ is discussed in Section 6.A of this manual.

Example:
SCALEMASS 0.195

SCALERP

SCALERP is a switch keyword for scaling previously calculated frequencies that appear in the VIB section of input. This keyword is off by default. Frequencies that have already been calculated with the FREQSCALE option are scaled, and therefore this option would not be required. Frequencies that have been calculated without scaling, and then used

with the FREQSCALE option would require this keyword in order for the program to scale them correctly.

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 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. The default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU. (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 for each of these steps (FSIZE). The default unit of FSIZE is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU. The defaults are NFSTEP $= 0$ and FSIZE $= 1 \times 10^{-5}$ a_o.

    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 ($V_a{}^G$) 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 $V_a{}^G$ curve is chosen, the MEP will be determined until the $V_a{}^G(s)$ becomes smaller then the value calculated as: $[(1-x) V_a{}^G(s=0) + x V+\text{ZPE}(\text{R or P})]$ 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 $V_a{}^G(\text{R or P})$ for a bimolecular reaction but not for an unimolecular reaction for which $V_a{}^G(\text{R or P})$ 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 $V_{MEP}(s)$ becomes smaller then the value calculated as: $[(1-x) V_{MEP}(s=0) + x V_{MEP}(\text{R or P})]$ where $V_{MEP}(\text{R or P})$ 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.

Option	Description	Default*
CURVE	the curve on which comparison is made (the options are *vmep* and *vag*)	*vag*
POINT	the grid on which the end point is chosen (the options are *savegrid* and *finegrid*)	*savegrid*
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 $|s| >$ SPRNT for any reaction-path information to be printed. Default is 0. The default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU.

> *Example:*
> ```
> SPRNT   0.1
> ```

SRANGE

SRANGE is a list keyword needed to specify the limits on the reaction coordinate, s for which the MEP is to be computed. These limits, SLM and SLP, are the ranges over which the MEP is computed, i.e., SLM $\leq s \leq$ SLP. The default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU. The default values are SLM $= -1.0$ a_o and SLP $= +1.0$ a_o.

> *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 will be included in the Hessian grid. The default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU. 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.

SSTOR

SSTOR is a list keyword that specifies the parameters used for the torsional-potential anharmonicity calculations along reaction path by the SS-T method. Note that when only a single structure among all conformational structures is considered, the MS-T method is reduced to the SS-T method as described in Section 10.O. The SSTOR keyword also turns on the SS-T calculations along reaction path. The valid keyword options are defined below.

Option	Description	Default*
MTOR	local periodicities of torsions, i.e. $M_{j,\tau}$ values	*no default*

Example:
```
SSTOR
      MTOR          3 3 3 3
   END
```

SSTEP

SSTEP is a variable keyword that specifies the step size along the mass-scaled MEP for following the gradient of *V*, except that the first *n* steps can be different (see SFIRST). The default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU. The default is 5×10^{-4} a_o.

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
```

12.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.

<u>Basic:</u>

Keyword	Type	Default	Description
Common:			
SCT	switch	on	small-curvature tunneling
ZCT	switch	on	zero-curvature tunneling
Others:			
EMIN	variable	0	cutoff E for computing percent tunneling
QRST	list	not used	quantized-reactant-state tunneling
HARMONIC	switch	on	harmonic approximation
HARMWKB	switch	off	harmonic-WKB approximation
SYMWKB	switch	off	WKB and symmetric V_a^G
MODE	variable	required	number of the selected vibrational reaction-path mode
SRW	variable	required	s value of the reactant
STATES	variable	all	number of states included in calculation
QUAD	list	no changes	change quadrature parameters
NSEGBOLTZ	variable	1	segments for Boltzmann integral
NSEGTHETA	variable	1	segments for theta integral
NQE	variable	40	quadrature points for Boltzmann integral
NQTH	variable	40	quadrature points for theta integral
SCTOPT	list	spline	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:

Keyword	Type	Default	Description
Common:			
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
Others			
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`	quadrature points 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	`999`	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 aamod.f90 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 $n = 0,1,....$ 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 μOMT results (described in Section 6.I) are always calculated. Of the two options (LCT3 and LCT4), LCT4 is recommended; LCT4, being the final version, is usually just denoted LCT.

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 14.

Option	Description
state n	*n* is the quantum number of the vibrational state (0 is the ground state) for which information is to be printed.
interval m	*m* is the number of intervals of the energy grid the user wants to print out for this particular quantum state; LB and UB are
lb_1 ub_1	the indices of the lower and upper bounds of the energy
lb_2 ub_2	grid interval, respectively.
::	
lb_m ub_m	

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

Option	Description
state n	*n* is the quantum number of the vibrational state (0 is the ground state) for which information is to be printed.
Npoints	are the single point energy calculations carried out at each tunneling energy and fitted to a spline. *Npoints* must be specified with **ILCT1D**. The default is *Npoints* equal to 9.
NumberE NumberX	The 2D grid to perform the **ILCT2D** calculation can be specified here. *NumberE* are the number of tunneling energies and *NumberX* 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.

Option	Description	Default
ILCT1D	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
ILCT2D	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
interpolate	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
lcgmethod	Select the method for LCT calculations. Two values are possible: 3 and 4, for LCG3 and LCG4 methods respectively.	4
lctrst	restart a previous large curvature calculation from unit fu48. This is useful to calculate LCT4 transmission factors from a previous LCT3 run.	off
lctstr	store information in unit fu49 for a possible restart.	on
ngamp	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
ngtheta	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
prfreq	print information in fu22 file about the frequency of the mode $p(s)$, the classical energy $V_{MEP}(s)$, 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
prprob	print all uniformized probabilities at each n_p to fu22	off
rtpjac	print the MEP and representative tunneling path turning points in mass-scaled pseudo-Jacobian coordinates to fu24	off
vadavg	average the overlapping of two adiabatic potentials instead of choosing the minimum of them.	off
vefsrch numb	search for possible nonadiabatic LCG4 regions in totally adiabatic LCG3 regions. The search starts from that tunneling energy until *numb*	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 determines 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 determines 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 *samet* (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.

Option	Description	Default
harmonic	use the harmonic approximation to determine the energy levels of the reaction-path mode	on
harmwkb	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
symwkb	use the WKB approximation to find the energy levels and assume that the effective potential is symmetric with respect to the reactant	off
mode	the number of the selected vibrational reaction-path mode	required
srw	the value of the reaction coordinate at the position of the reactant. The default unit is angstrom and it can also be in bohr by specifying the keyword INPUNIT AU.	required
states	the number of quantized reactant states to be included in the tunneling calculation	*all*
stqvib	the number of quantized reactant states to be included in the vibrational partition function calculation	*all*

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.

Option	Description	Default
nqe	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
nqth	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
nsegboltz	number of equal segments for integrals that average the energy–dependent tunneling probabilities over a Boltzmann distribution of energies	1
nsegtheta	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.

Option	Description	Default
lagrange n	Lagrangian interpolation of order n is used to interpolate $\mu_{\text{eff}}(s)$	4
spline	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.

12.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:

Keyword	Type	Default	Description
Common:			
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
Others:			
ANALYSIS	list	`not used`	detailed analysis of TST/CVT ratio
EDGEOK	switch	`on`	continue calculation 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* a$_O$	max. s for special grid
SMIN	variable	*–1.0* a$_O$	min. s for special grid
GTEMP	list	`not used`	special grid for ΔG calc. for given T
GTLOG	switch	`off`	free energy of activation by logarithms
STATE	variable	*therm*	thermal or state-selected rate constant
STATEOPT	list	*thermal*	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 vibrational partition function

Methods:

Keyword	Type	Default	Description
Common:			
CVT	switch	`on`	canonical VTST
Others:			
TST	switch	`on`	conventional transition state theory
MUVT	switch	`off`	microcanonical VTST
MUVTOPT	list		options for μVT
FIT	variable	*gas53*	primary and secondary fits
NITER	variable	*30*	Gauss-Laguerre quadrature
PRENERGY	variable	*0*	# of energies on the grid to print
SLOWER	variable	*0.0*	lower limit of MEP for searching
SUPPER	variable	*0.0*	upper limit of MEP for searching
CUS	variable	`off`	canonical unified statistical model
US	switch	`off`	unified statistical calculation

Print:

Keyword	Type	Default	Description
Others:			
PRDELG	switch	`off`	print free energy curve
PRPART	variable	`not used`	print reactant, product, and saddle point partition functions
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. 22. 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 default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU. The defaults for the options are SMAX $= 1$ a_o and SMIN $= -1$ a_o.

> *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 default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU. The defaults for the options are SMAX = 1 a_o and SMIN = -1 a_o.

    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 that at the zero point energy). The default is NOGTLOG.

MUVT (NOMUVT)

MUVT is a switch keyword that specifies that μVT 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 μVT calculation. If MUVTOPT is given then it is assumed that the μVT calculation is desired and the default for MUVT becomes on. The valid options are defined below.

Option	Description	Default
niter	number of Gauss-Laguerre quad. pts. for computing μVT rate constants	30
prenergy	number of energies on the grid to be printed out; PRENERGY must be $\leq \min\lvert 6, niter \rvert$	0
slower	lower limit, 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.) The default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU.	0.0
supper	upper limit, 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.) The default unit is angstroms and it can also be in bohr by specifying the keyword INPUNIT AU.	0.0
fit	variable that specifies the degree of the primary and secondary *gas53* fits. The allowed values are given below. For a gas-phase reaction, gas53 is the recommended value, and for a reaction at a gas-solid interface, gsolid13 is recommended. Sometimes there is a numerical problem with the 5-point fit; in such a case it is recommended to use *gas31* rather than *gas53*.	

value	primary	secondary	phase
gas53	5	3	gas
gas15	1	5	gas
gas35	3	5	gas
gas31	3	1	gas
gas13	1	3	gas
gsolid13	1	3	gas-solid interface
gsolid15	1	5	gas-solid interface
gsolid31	3	1	gas-solid interface
gsolid53	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 to fu6. Default is
NOPRDELG.

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 to fu6. 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, that delineate a region for which extra vibrational mode and Q_{vib}
information be printed to fu6. The default is no extra printing. The default unit is
angstroms and it can also be in bohr by specifying the keyword INPUNIT AU.

    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} cm^3
molecule^{-1} 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:

Option	Description
therm	thermal run
adiab	adiabatic state-selected; STATEOPT is required
diab x	diabatic state-selected; x is the value of the reaction coordinate at which the adiabatic/diabatic switch is made; STATEOPT is required

Example:
```
STATE  adiab
```

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:

Value	Description
react1	reactant #1
react2	reactant #2
prod1	product #1
prod2	product #2
ts	transition state and along the reaction path
switts	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:

Value	Description
thermal	mode is thermal (default)
ground	mode is restricted to the ground state
first	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 μVT, then the conventional rate constants will automatically be calculated (NOTST will have no effect) except when VRC is turned on

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 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
```

12.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 16.

Section Name	Description
*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
*TSHESS	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.

12.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
```

12.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
```

12.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
```

12.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.

Keyword	Type	Default	Description
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`

12.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
```

12.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.

Record   **Format**

(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 **F** or the projected Hessian $\mathbf{F}^P$?

(2) If the frequencies are calculated from the Hessian, how is **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 **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) ≠ 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

≤ 0, no anharmonicity data read

$= 1$, read the third derivatives k_3 with respect to the normal coordinates at each grid point

≥ 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

≤ 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.

≥ 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 ($\mathbf{F}$(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*NRATOM, NRATOM = number of atoms in the species, and I indexes the N records in this set (I = 1,..., N).

> If LOPT(4) ≠ 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 (**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,...., $N3$), and
       where $N3$ = 3*NATOM, NATOM = number of atoms in the system.

If LOPT(4) ≠ 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) ≠ 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 reaction-path Hamiltonian 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*(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, *N*3)   (*)

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 *N*3 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) ≠ 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 ($\boldsymbol{B}_{mF}$ 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.

12.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.

12.D.1. Section descriptions

Section Name	Description
*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.

12.D.2 GENERAL section

Keyword	Type	Default	Description
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	`off`	interpolation of the exponent in the effective tunneling mass
KIES	switch	*off*	recalculation of the s 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.

Option Definition
ang   Geometries are given in Å.
au   Geometries are given in atomic units (default).
ms   Geometries are given in mass-scaled Cartesian coordinates.
uns   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.

Option   Definition
ang   Gradients are given in Å.
au   Gradients are given in atomic units (default).
ms   Gradients are given in mass-scaled Cartesian coordinates.
uns   Gradients are given in unscaled Cartesian coordinates
(default).
forces   Forces are given; the gradients are obtained from them.
grads   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.

Option Definition
ang   Gradients are given in Å.
au   Gradients are given in atomic units (default).
ms   Gradients are given in mass-scaled Cartesian coordinates.
uns   Gradients are given in unscaled Cartesian coordinates (default).
full   Full Hessian matrices are given.
packed   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 × 6 matrix is shown below:

Full matrix - example 1

$$
\begin{array}{cccccc}
F_{11} & F_{12} & F_{13} & F_{14} & F_{15} & F_{16} \\
F_{21} & F_{22} & F_{23} & F_{24} & F_{25} & F_{26} \\
F_{31} & F_{32} & F_{33} & F_{34} & F_{35} & F_{36} \\
F_{41} & F_{42} & F_{43} & F_{44} & F_{45} & F_{46} \\
F_{51} & F_{52} & F_{53} & F_{54} & F_{55} & F_{56} \\
F_{61} & F_{62} & F_{63} & F_{64} & F_{65} & F_{66}
\end{array}
$$

Full matrix - example 2

$$
\begin{array}{ccccccccc}
F_{11} & F_{12} & F_{13} & F_{14} & F_{15} & F_{16} & F_{21} & F_{22} & F_{23} \\
F_{24} & F_{25} & F_{26} & F_{31} & F_{32} & F_{33} & F_{34} & F_{35} & F_{36} \\
F_{41} & F_{42} & F_{43} & F_{44} & F_{45} & F_{46} & F_{51} & F_{52} & F_{53} \\
F_{54} & F_{55} & F_{56} & F_{61} & F_{62} & F_{63} & F_{64} & F_{65} & F_{66}
\end{array}
$$

Packed matrix - example 1

F_{11}
F_{21} F_{22}
F_{31} F_{32} F_{33}
F_{41} F_{42} F_{43} F_{44}
F_{51} F_{52} F_{53} F_{54} F_{55}
F_{61} F_{62} F_{63} F_{64} F_{65} F_{66}

Packed matrix - example 2

F_{11} F_{21} F_{22} F_{31} F_{32} F_{33}
F_{41} F_{42} F_{43} F_{44} F_{51} F_{52}
F_{53} F_{54} F_{55} F_{61} F_{62} F_{63}
F_{64} F_{65} F_{66}

The order of the coordinates is explained in Section 12.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.

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 an 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.

12.D.3. REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and SADDLE sections

Keyword	Type	Default	Description
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 exists 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 $V_{MEP}(s)$ 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.

12.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 "*".

12.E.1 Section Descriptions

Section Name	Description
*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 $s_1, s_2, s_3, ... s_N$ (in the notation of pages 96-99 of Ref. 14 of Sect. 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.

12.E.2 GEN40 section

Keyword	Type	Default	Description
TITLE	list	blank	job title or comment
PRINT	variable	*noextra*	print options
MAXLPTS	variable	*3*	maximum number of points used in Lagrange interpolation
GEOMUNIT	variable	*bohr*	unit for coordinate input
GRADUNIT	variable	*bohr*	unit for gradient input
FREQSOURCE	variable	*read*	option for obtaining frequencies
FREQUNIT	variable	*waven*	option for frequency input
HESSFORM	variable	*full*	form of Hessian matrix
HESSUNIT	variable	*bohr*	option for Hessian matrix input
ANHARM	variable	*none*	option for anharmonicity input
CURVATURE	variable	*setzero*	curvature calculation options
GRADDER	variable	*noextra*	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.

Option	Definition
none	No anharmonic data is read (default).
cubic	Read the third derivatives $k3$ with respect to the normal coordinates at each grid point.
quartic	Also read in fourth derivatives $k4$ 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).

Option	Definition
setzero	The B_{mF} are set to zero everywhere (default).
read	The B_{mF} are read directly.
compute	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.

Option	Definition
read	Frequencies are read directly (default).
hessian	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.

    Option   Definition

Option	Definition
waven	Frequencies are given in cm^{-1} (default)
au	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.

Option	Definition
	coordinates are in:
bohr	unscaled bohrs (default)
msbohr	mass-scaled bohrs
ang	unscaled angstroms
msang	mass-scaled angstroms
si	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*.

Option	Definition
noextra	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,
noextra	means that the gradient grid is the basic input grid.
oneextra	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. `oneextra` means that the gradient grid has twice as many points as the basic input grid.

twoextra 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).

Option	Definition
	gradients are in:
bohr	hartrees per unscaled bohr (default)
msbohr	hartrees per mass-scaled bohr
ang	hartrees per unscaled Å
msang	hartrees per mass-scaled Å
si	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.

Option	Definition
full	**F** is input as the full matrix (default).
packed	**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

$$\begin{array}{cccccc}
F_{11} & F_{12} & F_{13} & F_{14} & F_{15} & F_{16} \\
F_{21} & F_{22} & F_{23} & F_{24} & F_{25} & F_{26} \\
F_{31} & F_{32} & F_{33} & F_{34} & F_{35} & F_{36} \\
F_{41} & F_{42} & F_{43} & F_{44} & F_{45} & F_{46} \\
F_{51} & F_{52} & F_{53} & F_{54} & F_{55} & F_{56} \\
F_{61} & F_{62} & F_{63} & F_{64} & F_{65} & F_{66}
\end{array}$$

Full matrix - example 2

$$\begin{array}{ccccccccc}
F_{11} & F_{12} & F_{13} & F_{14} & F_{15} & F_{16} & F_{21} & F_{22} & F_{23} \\
F_{24} & F_{25} & F_{26} & F_{31} & F_{32} & F_{33} & F_{34} & F_{35} & F_{36} \\
F_{41} & F_{42} & F_{43} & F_{44} & F_{45} & F_{46} & F_{51} & F_{52} & F_{53} \\
F_{54} & F_{55} & F_{56} & F_{61} & F_{62} & F_{63} & F_{64} & F_{65} & F_{66}
\end{array}$$

Packed matrix - example 1

$$\begin{array}{cccccc}
F_{11} & & & & & \\
F_{21} & F_{22} & & & & \\
F_{31} & F_{32} & F_{33} & & & \\
F_{41} & F_{42} & F_{43} & F_{44} & & \\
F_{51} & F_{52} & F_{53} & F_{54} & F_{55} & \\
F_{61} & F_{62} & F_{63} & F_{64} & F_{65} & F_{66}
\end{array}$$

Packed matrix - example 2

$$\begin{array}{cccccc}
F_{11} & F_{21} & F_{22} & F_{31} & F_{32} & F_{33} \\
F_{41} & F_{42} & F_{43} & F_{44} & F_{51} & F_{52} \\
F_{53} & F_{54} & F_{55} & F_{61} & F_{62} & F_{63} \\
F_{64} & F_{65} & F_{66} & & &
\end{array}$$

The order of the coordinates is explained in Section 12.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.

Option	Definition
	Hessians are in:
bohr	hartrees per unscaled bohr^2 (default)
msbohr	hartrees per mass-scaled bohr^2
ang	hartrees per unscaled Å^2
msang	hartrees per mass-scaled Å^2
si	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 bohr^2.

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.

Option	Definition
noextra	Print out titles and options (default).
summary	Also print out summary.
input	Also echo input as it is read.
intermediate	Also print out intermediate information about fitting exponential functions and print out interpolated results each time RPHINT is called.
debugging	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.

12.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.

Keyword	Type	Default	Description
SVALUE	variable	*dgr* (or *dgp*)	value of s (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	*1 2 3 ...*	indices for reordering frequencies, anharmonicities, and curvature components
FREQ	list	(see Glossary)	vibrational frequencies for the reactant or product species
3RDDERIV	list	(see Glossary)	third derivatives of V with respect to motion along the normal coordinates for the reactant or product
4THDERIV	list	(see Glossary)	fourth derivatives of V 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)

dgr $$s = \max \begin{cases} -2.0a_0 \\ s_1 - (s_2 - s_1) \end{cases}$$

dgp $$s = \max \begin{cases} 2.0a_0 \\ s_N + (s_N - s_{N-1}) \end{cases}$$

VVALUE

VVALUE is a required variable keyword that specifies the value of the potential (in a.u.) at the reactant or product.

12.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.

Keyword	Type	Default	Description
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
3RDDERIV	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 FREQUNIT 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 FREQUNIT to *au*.

Keyword	Type	Default	Description

12.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.

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

Keyword	Type	Default	Description
HESSIAN	list	(see Glossary)	Hessian matrix at the current grid point
FREQ	list	(see Glossary)	vibrational frequencies at the current grid point
3RDDERIV	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 FREQUNIT 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.

12.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.

Keyword	Type	Default	Description
0IVTST	switch	off	zero-order IVTST calculation
(NO)DETMI	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.
MEPTYPER	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 *V* in the nonadiabatic region in LCT calculations

Keyword	Type	Default	Description
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	s value at this point
ENEPCS	variable	not used	lower-level V at this point
ENEPCA	variable	not used	corrected V 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	s value at this point
ENERCS	variable	not used	lower-level V at this point
ENERCA	variable	not used	corrected V 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	au	units used for det **I** input

Glossary for file fu50 keywords

0IVTST

0IVTST is a switch keyword that specifies that the zero-order IVTST calculation will be performed. The default is off.

DETMI (NODETMI)

DETMI 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 UNITMI keyword, which is discussed latter in this section. For an atomic species, the user should input a value of zero for det **I**. The default is NODETMI, i.e., no correction for det **I**.

Example:
```
DETMI
      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 latter in this section, are not both equal to *two*, 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:

Option	Description
all	correct the moments of inertia of all the hindered-rotor modes of the reactants, products, and saddle point.
−n(n=1,2...)	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...)	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:

Option	Description
quadlin	linear or quadratic correction depending on the magnitude of the corrections in different regions
linear	linear correction only
nocor	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:

Option	Description
one	There is only one reactant species and interpolation is carried out between this species and the saddle point.
two	There are two reactant species and interpolation is carried out between these species (using the merged set of frequencies) and the saddle point.
well	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:

Option	Description
one	There is only one product species and interpolation is carried out between this species and the saddle point.
two	There are two product species and interpolation is carried out between these species (using the merged set of frequencies) and the saddle point.
well	The correction on the product side is based on a well.

Example:
```
MEPTYPER   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 inertial 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 inertial 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 CH_4 + OH reaction, the symmetry number for the internal rotation of the CH_3 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.

Option	Description
spc	s value in bohrs at this point
enepcs	lower-level potential energy in kcal/mol at this point (required only if MEPTYPEP is *well*)
enepca	corrected potential energy in kcal/mol at this point (required only if MEPTYPEP is *well*)
nofreqcorr	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 MEPTYPER 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.

Option	Description
src	s value in bohrs at this point
enercs	lower-level potential energy in kcal/mol at this point (required only if MEPTYPER is *well*)
enerca	corrected potential energy in kcal/mol at this point (required only if MEPTYPER is *well*)
nofreqcorr	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.

Option	Description	Default
rpl	range parameter L in bohrs for an Eckart function	1.0
br	range parameter for a cut-off Gaussian function on the reactant side	5.0
bp	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 MEPTYPER 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:

Option	Description
au	atomic units
si	SI units, i.e., kg m^2 or (kg m^2)^3
cgs	c.g.s. units, i.e., g cm^2 or (g cm^2)^3
scale	scaled units, i.e., 10^{-40} g cm^2 or (10^{-40} g cm^2)^3
amu	atomic mass unit, i.e., a.m.u. bohr^2 or (a.m.u. bohr^2)^3 .

Example:
```
   UNITMI     AMU
```

12.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.

12.G.1. Section Descriptions

Section Name	Description
*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.

12.G.2 ISPEGEN section

The *ISPEGEN section is used to provide information at the stationary points.

Keyword	Type	Default	Description
EXERXN	variable	required	corrected exoergicity
ENESAD	variable	required	corrected classical barrier height
MEPTYPER	variable	*two*	type of reactant-side MEP
MEPTYPEP	variable	*two*	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
ENEPCA	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:

Option	Description
one	There is one product and wells, if any, are not recognized.
two	There are two products and wells, if any, are not recognized.
well	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:

Option	Description
one	There is one reactant and wells, if any, are not recognized.

| two | There are two reactants and wells, if any, are not recognized. |
| well | 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:

Option	Description
spc	value of s, in bohrs, at this point
enepcs	lower-level potential energy, in kcal/mol, at this point with respect to the reactant (required if and only if MEPTYPEP is set to *well*)
enepca	corrected potential energy, in kcal/mol, at this point with respect to the reactant (required if and only if MEPTYPEP is set to *well*)

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 MEPTYPER 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:

Option	Description
src	value of s, in bohrs, at this point
enercs	lower-level potential energy, in kcal/mol, at this point with respect to the reactant (required if and only if MEPTYPER is set to *well*)
enerca	corrected potential energy, in kcal/mol, at this point with respect to the reactant (required if and only if MEPTYPER is set to *well*)

Example:
      PCINFO
            src   *-3.0*
            enercs   *-5.0*
            enerca   *-4.0*

12.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.

Keyword	Type	Default	Description
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
```

13. DESCRIPTION OF INPUT FOR VRC-VTST

In VRC-VTST rate calculations, input file fu5 is the only one required; the other input files used in RP-VTST calculations are not needed.

13.A Section descriptions of input file fu5

The following input sections correspond to the major parts of a VRC-VTST rate calculation. Each section begins with a line giving the section name, preceded by an "*". Keywords for each section are discussed further in the rest of this subsection. In each section, we list the keywords used in VRC-VTST rate calculations. The keywords that are only used for VRC-VTST are discussed in this section; the keywords that can be used for both RP-VTST and VRC-VTST are not discussed again and users can find the details in Section 12.

Section Name	Description
*GENERAL	general data such as the title and defining the atoms involved
*ENERGETICS	options for computing potential energy
*REACT1	definition and properties of the first reactant
*REACT2	definition and properties of the second reactant
*START	Only used to define the electronic degeneracy for generalized transition state
*PROD1	definition and properties of the first product
*PATH	definition of reaction coordinate range and step size
*RATE	options for computing the reaction rate

13.A.1. GENERAL section

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 the box that are only used for VRC-VTST rate calculations. In any run for VRC-VTST this section would have the TITLE, ATOMS, and VRC keywords.

Basic:

Keyword	Type	Default	Description
TITLE	list	`blank`	title with 5 lines maximum
ATOMS	list	`required`	atoms in system
CHECK	switch	`off`	check the input data
INPUNIT	variable	`angstrom`	specifies the units in fu5 input file
OUTUNIT	variable	`angstrom`	specifies the units in fu6 and fu14 output files
SUPERMOL	switch	`on`	supermolecule mode for reactants and products
VRC	switch	`Off`	use variable reaction coordinate to treat association reaction with loose transition state
VRCOPT	list	`not used`	variable reaction coordinate algorithm options
JMAX	variable	300	maximum angular momentum quantum number J in E,J -μVT
JSTEP	variable	5	step size in the integral over J in E,J -μVT
GTYPE	variable	4	type of random number generator in SPRNG
NITER	variable	4	Gauss-Laguerre quadrature
NMC	variable	1000	number of Monte Carlo samples per facet in multifaceted dividing surface

Glossary of GENERAL keywords only used for VRC-VTST

VRC

VRC is a switch keyword that allows the user to employ the variable reaction coordinate algorithm to treat loose transition states. When this keyword is turned on, the PROD1, PROD2, WELLR, WELLP, and START sections are not used in fu5 input file. Only forward reaction rates will be calculated in this case. If each reactant has only one pivot point, a spherical (i.e. single-faceted) dividing surface will be used. If more than one pivot point is assigned to one or two of the reactants, a multifaceted dividing surface will be used. Any keywords related the MEP algorithm are not used when VRC is turned on.

VRCOPT

VRCOPT is a list keyword that allows the user control over the parameters used in variable reaction coordinate algorithm. If VRCOPT is given then it is assumed that the VRC calculation is desired and the default for VRC becomes on. The valid *opt*ions are defined below.

Option	Description	Default
jmax	maximum angular momentum quantum number J calculated for the number of accessible state $N(E,J)$ in E,J -μVT	300
jstep	step size of angular momentum quantum number J in the integral over J to calculate the contribution to reactive flux from transitional modes	5
gtype	the type of random number generator in SPRNG package. Type 4 is Multiplicative Lagged Fibonacci Generator in SPRNG version 2.0. It is not recommended to change the default value. User needs to refer SPRNG manual to change the value to the desired type.	4
niter	number of Gauss-Laguerre quadrature points for total energy E.	4
nmc	number of Monte Carlo sampling for each facet in a multifaceted dividing surface. In parallel calculation, the actual number used in the calculation may be changed according to the number of processors used. If the remainder when NMC is divided by number of processors is not zero or NMC is smaller than number of processors, the number of sampling for each processor will be int(NMC/number of processors) + 1. User need to be careful to balance the nmc and number of processors used in parallel calculation.	1000

Example:
```
VRCOPT
   NMC          600
   JMAX         500
   JSTEP        10
END
```

13.A.2. ENERGETICS section

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

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

Glossary of ENERGETICS keywords

EZERO

Refer to Section 12.A.3

EZUNIT

Refer to Section 12.A.3

POTENTIAL

POTENTIAL is a variable keyword that allows the user to specify the source of the potential energy information for a single-level calculation. There are only one valid option for a VRC-VTST calculation, namely *hooks*. 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. When *Polyrate* is used in conjunction with a interface program, e.g. GAUSSRATE, the hooks call the corresponding electronic structure program to perform electronic structure calculations to obtained potential energy.

Option	Definition
hooks	use the hooks

Example:
```
POTENTIAL hooks
```

13.A.3. OPTIMIZATION section

The OPTIMIZATION section is used to specify the method for carrying out geometry optimizations when the POTENTIAL keyword selection is *hooks*. In principle, the keywords used for RP-VTST can also be used for VRC-VTST to optimize reactants. But these keywords have not been tested for VRC-VTST calculations. Therefore it is strongly recommended to input the optimized geometries instead of using *Polyrate* to optimize the reactant geometries because a semi-global analytical potential energy surface when using *Polyrate* as a stand-alone package is required in VRC-VTST calculation and it usually doesn't have analytical derivatives used for geometry optimization.

13.A.4. REACT1, REACT2, PROD1, AND START sections

The summary table below gives the keywords that can be used for a VRC-VTST calculations in REACT1, REACT2, PROD1, and START sections. Note that in VRC-VTST calculations an additional keyword is introduced for setting the Cartesian coordinates of pivot points used for the variable reaction coordinate. These pivot points are only required in the REACT1 and REACT2 sections.

In START section, only the ELEC keyword is used. This gives electronic degeneracy of the generalized transition state. No other keyword is required in this section when *Polyrate* runs as a stand-alone program. When running in direct dynamics with other rate program, e.g. GAUSSRATE, the STATUS keyword should be set to 6 to avoid any electronic structure calculation for START section.

Basic:

Keyword	Type	Default	Description
Required:			
GEOM	list	`required`	Cartesian coordinates of atoms
PIVOT	list	`Required only for reactants`	Cartesian coordinates of pivot points for reactants
Common:			
ELEC	list	`all degen. = 1`	electronic degeneracies
ENERGY	variable	`none`	energy of the stationary point
FREQ	switch	`on`	perform normal-mode analysis
FREQUNIT	variable	`waven`	units of the frequencies
INITGEO	variable	`geom`	location of input data that specify initial geometries
STATUS	variable	`0`	status of the stationary point work
SPECIES	variable	`nonlinrp`	species type (linearity)
VIB	list	`none`	frequencies of the stationary points

Glossary of REACT1 and REACT2 keywords

PIVOT

PIVOT is a list keyword that is used to define the locations of pivot points. The Cartesian coordinates of pivot points must be given along with their identification number. The identification number can be the same for pivot points on the different reactants. The origin of the Cartesian coordinates must be the same as the origin of the corresponding reactant Cartesian coordinates, so that they can give the positions relative to the corresponding reactant.

Example:
```
PIVOT
     1    0.000000   0.000000  0.200000
     2    0.000000   0.000000 -0.200000
END
```

13.A.5. PATH section

The PATH section contains the keywords that control the calculation of the reaction path. In VRC-VTST rate calculations, only one keyword, SVRC, is allowed in this section.

SVRC

SVRC is a list keyword that specifies the reaction coordinate s range and step size used in variable reaction coordinate and single-faceted or multifaceted dividing surface algorithm for loose transition state. If VRC keyword in GENERAL section is turned on, user needs to specify this keyword only in the PATH section, and the other keywords in the PATH section are not required.

> *Example:*
> ```
> SVRC
>     2.0  4.0  0.1
> END
> ```

This example tells the program that the s range is 2.0 – 4.0 Å (note that default unit is Å) and that the step size is 0.1 Å.

13.A.6. RATE section

The RATE section contains the keywords that control the rate constant calculations. Note that a conventional TST calculation is not a valid option for VRC-VTST rate since there is no barrier on potential energy surface. Only forward rate constants (bimolecular association rate) are calculated for VRC-VTS and the reverse unimolecular dissociation rate calculation is not implemented yet. The VRC-VTST rate constants are calculated by CVT, μVT, and E,J -μVT. An alphabetical glossary of all the keywords follows the summary table.

Keyword	Type	Default	Description
SIGMAF	variable	*1.0*	sym. factor for forward reaction
SIGMAR	Variable	*1.0*	sym. Factor for reversed reaction
TEMP	list	see below	temperatures for rate calculations

SIGMAF

SIGMAF is a variable keyword whose argument specifies the number of equivalent ways for forming the GTS for the forward reaction. Note that only two values are used in VRC-VTST calculations. The value should be 1 for the case where the two reactants are different, and the value should be 0.5 for the case two reactants are identical.

SIGMAR

SIGMAF is a variable keyword whose argument specifies the number of equivalent ways for forming the GTS for the reversed reaction.

14. 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

Part V More General Program Package Information

15. 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 Perl and C Shell scripts should make *Polyrate* reasonably easy to install, test, and use.

Subsection 15.A discusses running *Polyrate* in a Unix environment, including Linux. Subsection 15.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.

15.A. Instructions for a Unix environment

If the *Polyrate* program has been received in tar.gz format, the recommended directory structure will be created automatically when this file is uncompressed and untarred, and the files will be put in their canonical places as listed in Subsection 15.A.1.

15.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 *Polyrate*17 directory, i.e., it is the parent directory where you are going to install *POLYRATE*.

Polyrate may be downloaded as a single tar.gz file with a structure that is illustrated in STEP 1. To uncompress and untar the *Polyrate* tar.gz file in the canonical place the user has to execute:
```
user@machine [~/Polyrate17] % tar —xzvf Polyrate17.tar.gz
```

STEP 1   Verify that the following directory structure was created properly when the file
              was uncompressed and untarred. User may use the 'ls –R' command.

```
user@machine [~/Polyrate17] % ls –R
```

PC/   doc/   obj/   script/   src/   sprng2.0/   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:
hnitr1.dat   hnitr1.fu15   hnitr1.fu6   poly_hni.mak   pothni.dat
hnitr1.fu14   hnitr1.fu5   hnitr1.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:
c2h7pot5.f ch4o_module.f90 chain4.F coord3.F dumpot.F h3.F   ho2.F ohcl.F
potch5j2.dat potcwmc3b.dat poth2nilat.dat setup4.F   ch3.F   ch5.F   clhbr.F coord4.F h2ni.F
h3_module.F nh3.F potch4o.dat potcmc1.dat potcwmcvib.dat pothni.dat   surf3.F
ch4o.F   chain3.F   cmc.F   cwmc.F h2o2.F hni.F   oh3.F potch5j1.dat potcmc2.dat
poth2niads.dat setup3.F   surf4.F

./script:
comp_pot.jc* comp_src.jc* exe_chk.jc* link_exe.jc* potcomp.jc*

./src:
aamod.f90 dattim.F   ef.F   fromlapack.F hooks.F intbsv2.F.bak ivtstm.F   Makefile
polyag.F   polymq.F rtpjac.F   util.f90 acespoly.F dummy_mpi.F energetics.F givtst.F
intbsv1.F intbsv3.F   main.F   poly31.F polyhl.F polyrr.F sprng_f.h vrctst_mpi.F
alloc.f90 dumpot.F fromblas.F headr.F   intbsv2.F interface.F   main_vrcmpi.F poly40.F
poly_mpi.F polysz.F torsion.f90

./src/acesrate:
acespoly.F

./src/morate:
mopoly.F

./testo:
ch3tr1.fu15   ch5fu40tr2.fu15   ch5j1tr2.fu15   cmctr1.fu15   nh3tr1.fu6   oh3tr2.fu15
ch4cltr1.fu15   ch5fu40tr3.fu15   ch5j1tr3.fu15   cmctr2.fu15   nh3tr2.fu15   oh3tr3.fu15
ch4cltr2.fu15   ch5fu40tr4.fu15   ch5j1tr4.fu15   cmctr3.fu15   oh3fu30tr1.fu15 oh3tr4.fu15
ch4ohtr1.fu15   ch5fu40tr5.fu15   ch5j1tr5.fu15   cwmctr1.fu15 oh3fu30tr2.fu15 oh3tr5.fu15
ch4ohtr2.fu15   ch5fu40tr6.fu15   ch5j1tr6.fu15   cwmctr2.fu15 oh3fu30tr3.fu15 oh3tr6.fu15
ch4ohtr3.fu15   ch5fu50tr1.fu15   ch5j1tr7.fu15   cwmctr3.fu15 oh3fu30tr4.fu6 oh3tr7.fu15
ch4ohtr4.fu15   ch5fu50tr2.fu15   ch5j1tr8.fu15   h2nitr1.fu15 oh3fu30tr5.fu6 oh3tr8.fu15
ch4otr1.fu15   ch5fu50tr3.fu15   ch5j1tr9.fu15   h3o2fu31tr1.fu15 oh3fu30tr6.fu15 oh3tr9.fu15
ch5fu29tr1.fu15 ch5fu51tr1.fu15   ch5j2itr1.fu15 h3tr1.fu15   oh3fu31tr1.fu15 ohcltr1.fu15
ch5fu29tr2.fu15 ch5icfu30tr1.fu15 ch5j2itr2.fu15 h3tr2.fu15   oh3fu31tr2.fu15 ohcltr2.fu15
ch5fu30tr1.fu15 ch5icfu30tr2.fu15 ch5j2tr1.fu15 h3tr3.fu15   oh3fu40tr1.fu15 ohcltr3.fu15
ch5fu30tr2.fu15 ch5icfu31tr1.fu15 ch5j2tr2.fu6   h3tr4.fu15   oh3fu40tr2.fu15 ohcltr4.fu15
ch5fu30tr3.fu15 ch5icfu40tr1.fu15 ch5j2tr3.fu15 hnitr1.fu15   oh3fu40tr3.fu15 ohcltr5.fu15
ch5fu30tr4.fu15 ch5icfu40tr2.fu15 clhbrtr1.fu15 ho2tr1.fu15   oh3fu40tr4.fu6
ch5fu30tr5.fu15 ch5j1tr10.fu15   clhbrtr2.fu15 ho2tr2.fu15   oh3fu40tr5.fu6
ch5fu30tr6.fu15 ch5j1tr11.fu15   clhbrtr3.fu15 ho2tr3.fu15   oh3fu40tr6.fu15
ch5fu31tr1.fu15 ch5j1tr12.fu15   clhbrtr4.fu15 ho2tr4.fu15   oh3tr10.fu15
ch5fu40tr1.fu15 ch5j1tr1.fu15   clhbrtr5.fu15 ho2tr5.fu15   oh3tr1.fu15

./testrun:
ch3   ch5   clhbr nh3   runch3.jc runch5.jc runh2ni.jc runho2.jc
ch4cl check_all.jc cmc   h3 h3o2 oh3   runch4cl.jc runclhbr.jc runh3o2.jc runnh3.jc
ch4o check_test.jc cwmc hni ohcl   runch4oh.jc runcmc.jc runh3.jc runoh3.jc
ch4oh clean_up.jc h2ni ho2 run_all.jc runch4o.jc runcwmc.jc runhni.jc runohcl.jc

./testrun/ch3:

ch3tr1.dat ch3tr1.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*   ch5j2tr1.dat
ch5fu30tr3.fu30   ch5fu40tr2.jc*   ch5fu50tr3.dat   ch5icfu40tr1.fu40 ch5j1tr4.dat   ch5j2tr1.jc*
ch5fu30tr3.jc*   ch5fu40tr3.dat   ch5fu50tr3.fu50   ch5icfu40tr1.fu50 ch5j1tr4.jc*   ch5j2tr2.dat
ch5fu30tr4.dat   ch5fu40tr3.fu40   ch5fu50tr3.jc*   ch5icfu40tr1.jc*   ch5j1tr5.dat   ch5j2tr2.jc*
ch5fu30tr4.fu30   ch5fu40tr3.jc*   ch5fu51tr1.dat   ch5icfu40tr2.dat   ch5j1tr5.jc*   ch5j2tr3.dat
ch5fu30tr4.jc*   ch5fu40tr4.dat   ch5fu51tr1.fu51   ch5icfu40tr2.fu40 ch5j1tr6.dat   ch5j2tr3.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/h3o2:
h3o2fu31tr1.dat h3o2fu31tr1.jc h3o2fu31tr1.fu31

./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:
oh3fu30tr1.dat oh3fu30tr4.dat oh3fu30tr7.dat oh3fu40tr1.dat oh3fu40tr4.dat oh3tr1.dat
oh3tr3.jc*   oh3tr8.dat
oh3fu30tr1.fu30 oh3fu30tr4.fu30 oh3fu30tr7.fu30 oh3fu40tr1.fu40 oh3fu40tr4.fu40 oh3tr1.jc*
oh3tr4.dat   oh3tr8.jc*
oh3fu30tr1.jc* oh3fu30tr4.jc* oh3fu30tr7.jc* oh3fu40tr1.jc* oh3fu40tr4.jc* oh3tr10.dat
oh3tr4.jc*   oh3tr9.dat
oh3fu30tr2.dat oh3fu30tr5.dat oh3fu31tr1.dat oh3fu40tr2.dat oh3fu40tr5.dat oh3tr10.jc*
oh3tr5.dat   oh3tr9.jc*
oh3fu30tr2.fu30 oh3fu30tr5.fu30 oh3fu31tr1.fu31 oh3fu40tr2.fu40 oh3fu40tr5.fu40 oh3tr11.dat
oh3tr5.jc*   tr7old.fu30
oh3fu30tr2.jc* oh3fu30tr5.jc* oh3fu31tr1.jc* oh3fu40tr2.jc* oh3fu40tr5.jc* oh3tr11.jc*
oh3tr6.dat
oh3fu30tr3.dat oh3fu30tr6.dat oh3fu31tr2.dat oh3fu40tr3.dat oh3fu40tr6.dat oh3tr2.dat
oh3tr6.jc*
oh3fu30tr3.fu30 oh3fu30tr6.fu30 oh3fu31tr2.fu31 oh3fu40tr3.fu40 oh3fu40tr6.fu40 oh3tr2.jc*
oh3tr7.dat
oh3fu30tr3.jc* oh3fu30tr6.jc* oh3fu31tr2.jc* oh3fu40tr3.jc* oh3fu40tr6.jc* oh3tr3.dat
oh3tr7.jc*

./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 *Polyrate17* 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 files .poly_path and .poly_ver will point to the location and the version

of *POLYRATE*, respectively that you are installing. Then the script will ask which kind of executable (RP-VTST or VRC-VTST) the Makefile to generate. You should input your choice, i.e., "RP" or "VRC". Next, the script will automatically detect available C and FORTRAN compilers and ask if you want to use the detected options. Usually, you should answer "yes" so that you can use the automatically detected compilers. If you want to use a different compiler, you should do the followings: (step 1) answer "yes" and proceed with the current configuring; (step 2) go to src/ directory, and open the Makefile; (step 3) At the beginning of the Makefile, you will see:

```
F77C = /usr/bin/gfortran −O −fno−automatic
 CC = /usr/bin/gcc
```

Please change the definition of these two variables (F77C and CC) to the location of your fortran and C compilers.

Note that no executable file is generated in this configuration step and the generated Makefile is for either RP-VTST or VRC-VTST.

STEP 3   Check the Makefile to ensure that the correct compiler options appear. *POLYRATE17* supports dynamic memory but previous versions do not, so compiler options that force static memory are typically required for previous versions. 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 test runs. If you want to use your own potential energy surface, you have to add it to the Makefile. The name of the new executable, the file name of the potential energy surface and its object have to be added. Comments in the Makefile will help you make these additions.

STEP 5   The following two compiling problems are common enough to merit special mention. If you get 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. If during the compilation of the RP-VTST version, the installation stops because the file mpif.h cannot be found, the user should generate a dummy file called mpif.h in the src directory (for instance, by typing *touch mpif.h* <Return> in the *Polyrate* src directory).
.

15.A.2. Testing

The directory structure and files should be set up as instructed in Subsection 15.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 the user run all the test runs in the distribution package before using *POLYRATE*. 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 are grouped into 15 different sets, in particular, ch3, ch4cl, ch4oh, ch5, clhbr, cmc, cwmc, h2ni, h3, h3o2, hni, ho2, nh3, oh3, and ohcl test runs. (See Subsection 16.C.) Each set has its own subdirectory in the *Polyrate17*/testrun directory. (See Subsection 15.A.1.) The subdirectories contain the input data files and the job control files for the test runs. The only thing you need to run all the test runs for either RP-VTST or VRC-VTST is to change the working directory to the *Polyrate17*/testrun directory and run the script run_allRPVTST.jc or run_allVRCVTST.jc.

```
./run_allRPVTST.jc & <Return>
```

(or submit it as a batch job)

Note `run_allRPVTST.jc` and `run_allVRCVTST.jc` need different Makefiles. The user has to take the steps described in Section 15.A.1 to generate a Makefile either for VRC-VTST calculations or for RP-VTST calculations.

(2)   To run a subset of the test runs

   1. To run a set of test runs, use the scripts available in the testrun/ directory. For example, to run all the h3 test runs, change the working directory to the *Polyrate17*/testrun directory and use the script runh3.jc.

      (change to the 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 the directory to 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 15.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 testo/ directory. There is a useful script, check_test.jc, in the 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 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 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 *Polyrate17*/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 *Polyrate17*/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 *Polyrate17*/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 *Polyrate17*/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.

15.A.3. Using POLYRATE

Many users will user *Polyrate* mainly as part of a direct dynamics calculations carried out with GAMESSPLUSRATE, GAUSSRATE, JAGUARRATE, MC-TINKERATE, MORATE, MULTILEVELRATE, or NWCHEMRATE. However the rest of this section is devoted to using *Polyrate* as a stand-alone program.

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 subroutine 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 into the Makefile located 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 *Polyrate17*/poten directory and named oh2.f. You have to add the following line where the executables are defined

```
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 the following line to compile the surface, beginning with a tab.

```
$(F77C) -c ../poten/oh2.f
```

Finally, add the lines that 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 12 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 *Polyrate17* directory or somewhere else, 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 *Polyrate17*/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. *Polyrate* only recognizes a certain set of input file names and will generate a set of output files with certain names (see Sections 12, 15.B and 16.C), therefore, 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. Using a script also makes 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 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 *Polyrate17*/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 that it is used, and also anytime there is a change in one of the files that it is dependent 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 .jc job control script once the

executable is compiled for the first time. Executables are put in the exe directory by default.

15.B. Comments for expert users

Of course, users need not use the provided Unix scripts, files, or 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 practical issues to be aware of, as follows.

Compilation of POLYRATE
For the RP version of the code, compile

>   dattim.F   ef.F   fromlapack.F hooks.F   intbsv2.F.bak ivtstm.F,
>   polyag.F   polymq.F rtpjac.F   util.f90   acespoly.F dummy_mpi.F
>   energetics.F givtst.F, intbsv1.F intbsv3.F   main.F   poly31.F
>   polyhl.F   polyrr.F sprng_f.h   vrctst_mpi.F, alloc.f90 dumpot.F
>   fromblas.F   headr.F   intbsv2.F interface.F   main_vrcmpi.F
>   poly40.F , poly_mpi.F polysz.F torsion.f90

For the RP version of the 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,

The *POLYRATE*17 program should be compiled with 64-bit precision and dynamic memory allocation.

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.

Creation of an executable file

Link the *Polyrate* object file with potential object file(s) to create specific *POLYRATE*-potential executable file(s).

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 (they correspond to input and output files respectively). 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 16.C. Also, these data files need to be in the working directory of the calculation.

15.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. There are two MPI versions. One is for calculations with MEP and the other one is for calculations with variable reaction coordinate (VRC). These two versions need to be compiled by including different subroutines since they use different parallel schemes.

A.   MPI version for calculations with MEP

For samples on how to make MPI work using MEP, 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
```

B   MPI version for calculations with VRC

When installing *Polyrate* program, user will be asked whether the MPI executable for variable reaction coordinate need to be compiled by a Perl installation script under *Polyrate17* directory, named *configure*. This script will automatically search MPI compiler for user. But user has to be responsible to make sure the compiler and its options are correct in Makefile since MPI can be very machine dependent.

Since this MPI version for variable reaction coordinate use SPRNG (SCALABLE PARALLEL RANDOM NUMBER GENERATOR LIBRARY) package in Monte Carlo integrations, user needs to install SPRNG before *Polyrate* installation. Starting with version 2010, SPRGN 2.0 package is included in *Polyrate* program. The *Polyrate* installation script will compile the SPRNG 2.0 library before generate Makefile for VRC-VTST run. The SPRNG package can also be downloaded from its website at the Florida State University (http://sprng.cs.fsu.edu/) and installed separately by users. When linking object files, user should give the correct path for SPRNG library by changing the $SPRNGLIB variable in Makefile if user desires use separately installed SPRNG library.

16. SAMPLE TEST RUNS

16.A. 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. They are files located in the *Polyrate17*/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)
ch3tr1	ch3.F	(none)
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)

16.B. 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*.

$CH_3 + H_2 \rightarrow CH_3$

$CH_3 + H_2 \rightarrow CH_4 + H$ see Refs. 17 and 20 in Sect. 21

$CH_3 + HPh \rightarrow CH_4 + Ph$ see Ref. 21 in Sect. 21

$Cl + HBr \rightarrow ClH + Br$ see Ref. 10 in Sect. 21

$Cl^- + CH_3Cl' \rightarrow ClCH_3 + Cl'^-$ see Ref. 23 in Sect. 22

$Cl^-(H_2O) + CH_3Cl' \rightarrow ClCH_3 + Cl'^-(H_2O)$ see Ref. 24 in Sect. 22

$H_2 + H' \rightarrow HH' + H$ see Ref. 12 in Sect. 21

$OH + O \rightarrow HO_2$ see Ref. 4 in Sect. 21

$O_2 + H \rightarrow HO_2$ see Ref. 4 in Sect. 21

$NH_3 \rightarrow NH_3$ (inversion)   see Ref. 7 in Sect. 21

$OH + H_2 \rightarrow H_2O + H$ see Refs. 1, 3, and 6 in Sect. 21

$H(a)/Ni(100) \rightarrow H(a)/Ni(100)$ see Ref. 33 in Sect. 22

$H_2(a)/Ni(100) \rightarrow 2H(a)/Ni(100)$ see Ref. 19 in Sect. 21

$OH + CH_4 \rightarrow H_2O + CH_3$ see Refs. 26 and 32 in Sect. 21

$Cl + CH_4 \rightarrow HCl + CH_3$ see Refs. 22 and 26 in Sect. 21

$CH_4 + O \rightarrow CH_3 + OH$ see Ref. 52 in Sect. 22

$O + HCl \rightarrow OH + Cl$

O + HCl(v=1) $\rightarrow$ OH + Cl   see Refs. 22 and 26 in Sect. 21

H_2O_2 + H $\rightarrow$ H_2O + OH

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 CH_3 + H_2 $\rightarrow$ CH_4 + 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 CH_3 + HPh $\rightarrow$ CH_4 + 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 Cl + HBr $\rightarrow$ ClH + Br use the extended LEPS surface of Babamov, Lopez, and Marcus [Ref. 26, Sect. 22]. The calculations for the $Cl^-(H_2O_m)$ + CH_3Cl' $\rightarrow$ $ClCH_3$ + $Cl'^-(H_2O_m)$ reactions, $m = 0,1$, use the potential energy surface of Tucker and Truhlar [Ref. 23, Sect. 22] and Zhao et al. [Refs. 24 and 27, Sect. 22], while the DMBE surface of Varandas et al. [Ref. 28, Sect. 22] is used for the reaction H_2 + H' $\rightarrow$ HH' + H. The calculations for the reaction OH + O $\rightarrow$ HO_2 and O + H_2 $\rightarrow$ HO_2 use the ab initio surface of Melius and Blint [Ref. 29, Sect. 22], and all the runs for the reaction OH + H_2 $\rightarrow$ H_2O + H use the Schatz-Elgersma fit to the Walch-Dunning ab initio potential energy surface [Refs. 30 and 31, Sect. 22]. The embedded diatomics-in-molecules (EDIM) potential with parameter set 4 of Truong et al. [Refs. 25 and 32, Sect. 22] is used for the H/Ni(100) surface diffusion test run, and the Lee and DePristo [Ref. 33, Sect. 22] potential energy surface, as modified by Truong et al. [Ref. 19, Sect. 21], is used for the H_2/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 Cl^- + CH_3Cl' $\rightarrow$ $ClCH_3$ + Cl'^- use the semiempirical surface of Tucker and Truhlar [Ref. 23, Sect. 22], and the calculations for the NH_3 inversion use potential energy surface II of Brown, Tucker, and Truhlar [Ref. 7, Sect. 21]. For the OH + CH_4 $\rightarrow$ H_2O + CH_3 [Ref. 32, Sect. 21] and Cl + CH_4 $\rightarrow$ HCl + 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.

16.C.1. Test run ch5j1tr1

CH$_3$ + H$_2$ → CH$_4$ + H   with harmonic approximation

This sample run is for the reaction CH$_3$ + H$_2$ → CH$_4$ + 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 = *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*]. A restart file containing reaction-path information is written to unit fu1 [RESTART = *writeful*] 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

16.C.2. Test run ch5j1tr2

$CH_3 + H_2 \rightarrow CH_4 + H$ with harmonic approximation

This sample run is for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ 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

16.C.3. Test run ch5j1tr3

$CH_3 + H_2 \rightarrow CH_4 + H$ with mass-weighted reaction coordinate

This sample run for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ 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

16.C.4. Test run ch5j1tr4

$CH_3 + H_2 \rightarrow CH_4 + H$ with harmonic approximation in redundant internal coordinates and the Page-McIver method to calculate reaction-path curvature

This sample run for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ demonstrates the calculation of reaction-path curvature components from the Hessian, generalized normal mode eigenvectors, and gradient [CURV = *dhess*] using eq. (48) of Page and McIver [Ref. 17, Sect. 22]. The MEP is followed using the Euler steepest-descents (ESD) method [RPM = *esd*] with a step size of 0.0001 bohr [SSTEP]. A generalized normal mode analysis is performed every 0.01 bohr along the MEP [INH = *100*], and all the vibrations are treated harmonically in redundant internal coordinates [COORD = *curv3*]. Both CVT [CVT] and ICVT [ICVT] rate constants are calculated, and tunneling corrections are computed using both MEPSAG and CD-SCSAG methods [ZCT, 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

16.C.5. Test run ch5j1tr5

$CH_3 + HPh \rightarrow CH_4 + Ph$ with LCG3 calculation for the ground state only

This sample run is for the $CH_3 + HPh \rightarrow CH_4 + Ph$ reaction, where Ph denotes a single particle with the mass of a phenyl group. The CH_5 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 = *writeful*]. 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

16.C.6. Test run ch5j1tr6

$CH_3 + HPh \rightarrow CH_4 + Ph$ 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 $CH_3 + HPh \rightarrow CH_4 + Ph$ 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

16.C.7. Test run ch5j1tr7

$CH_3 + H_2 \rightarrow CH_4 + H$ with Page-McIver mass-weighted reaction path and IVTST0FREQ

This sample run for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ 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. 22]. 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 1×10^{10} 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

16.C.8. Test run ch5j1tr8

CH$_3$ + H$_2$ → CH$_4$ + 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 CH$_3$ + H$_2$ → CH$_4$ + 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*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

16.C.9. Test run ch5j1tr9

CH$_3$ + H$_2$ → CH$_4$ + H   with harmonic approximation

This sample run is for the reaction CH$_3$ + H$_2$ → CH$_4$ + 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

16.C.10. Test run ch5j1tr10

$CH_3 + HPh \rightarrow CH_4 + Ph$ 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 $CH_3 + HPh \rightarrow CH_4 + Ph$ 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

16.C.11. Test run ch5j1tr11

$CH_3 + HPh \rightarrow CH_4 + Ph$ with interpolated LCG4 calculation with all energetically accessible excited states.

This sample run for the $CH_3 + HPh \rightarrow CH_4 + Ph$ 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

16.C.12. Test run ch5j2tr1

$CH_3 + H_2 \rightarrow CH_4 + H$ with harmonic approximation and ES1 method

This sample run for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ with the J2 potential energy surface is similar to the one described in Subsection 16.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

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

16.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

16.C.14. Test run ch5j2tr2

$CH_3 + H_2 \rightarrow CH_4 + H$ with MEP only

This sample run is for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$. 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

16.C.15. Test run ch5j2tr3

$CH_3 + H_2 \rightarrow CH_4 + H$ with WKB approximation for two modes

This sample run is for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ 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

16.C.16. Test run ch5j2itr1

$CH_3 + H_2 \rightarrow CH_4 + H$ with zero-order IVTST approximation

This zero-order IVTST test run [IVTST0] is for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ 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.

16.C.17. Test run ch5j2itr2

$CH_3 + H_2 \rightarrow CH_4 + H$ with first-order IVTST approximation

This first-order IVTST test run [IVTST1] is for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ 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.

16.C.18. Test run ch5fu29tr1

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 $CH_3 + H_2 \rightarrow CH_4 + H$ [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	*Polyrate* FILE NAME	
ch5fu29tr1.dat	poly.fu5	input data file
ch5fu29tr1.fu29	poly.fu29	input data file
ch5fu29tr1.fu6	poly.fu6	long output file

16.C.19. Test run ch5fu29tr2

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 $CH_3 + H_2 \rightarrow CH_4 + H$ [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	*Polyrate* FILE NAME	
ch5fu29tr2.dat	poly.fu5	input data file
ch5fu29tr2.fu29	poly.fu29	input data file
ch5fu29tr2.fu6	poly.fu6	long output file

16.C.20. Test run ch5fu30tr1

$CH_3 + H_2 \rightarrow CH_4 + H$ TST reaction rate with frequencies read from fu30 directly

This is a conventional TST test run for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ [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	*Polyrate* FILE NAME	
ch5fu30rt1.dat	poly.fu5	input data file
ch5fu30tr1.fu30	poly.fu30	input data file
ch5fu30tr1.fu6	poly.fu6	long output file

16.C.21. Test run ch5fu30tr2

$CH_3 + H_2 \rightarrow CH_4 + H$ TST reaction rate with information read from fu30 and full
Hessian matrices input

This is a conventional TST test run for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ [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	*Polyrate* FILE NAME	
ch5fu30tr2.dat	poly.fu5	input data file
ch5fu30tr2.fu30	poly.fu30	input data file
ch5fi30tr2.fu6	poly.fu6	long output file

16.C.22. Test run ch5fu30tr3

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 $CH_3 + H_2 \rightarrow CH_4 + H$ [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 (some anharmonicity options are not supported for
conventional TST calculations in *POLYRATE*–2017). 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	*Polyrate* FILE NAME	
ch5fu30tr3.dat	poly.fu5	input data file
ch5fu30tr3.fu30	poly.fu30	input data file
ch5fu30tr3.fu6	poly.fu6	long output file

16.C.23. Test run ch5fu30tr4

$CH_3 + H_2 \rightarrow CH_4 + H$ TST reaction rate with information read from fu30 and
frequencies input directly

This is a conventional TST test run for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ [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 $V_{MEP}(s)$ has been scaled by a factor of 0.86 to be consistent with Ref. 48 in Section 22. 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	*Polyrate* FILE NAME	
ch5fu30tr4.dat	poly.fu5	input data file
ch5fu30tr4.fu30	poly.fu30	input data file
ch5fu30tr4.fu6	poly.fu6	long output file

16.C.24. Test run ch5fu30tr5

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 $CH_3 + H_2 \rightarrow CH_4 + H$. 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 $amu^{1/2}$ a_0) 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, 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 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 Sect. 21. Several errors were detected in Ref. 20 (see Section 24.B in this manual for the corresponding errata).

Polyrate I/O FILES

FILE NAME	*Polyrate* FILE NAME	
ch5fu30tr5.dat	poly.fu5	input data file
ch5fu30tr5.fu30	poly.fu30	input data file
ch5fu30tr5.fu6	poly.fu6	long output file

16.C.25. Test run ch5fu30tr6

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 CH_3 + $H_2 \rightarrow CH_4$ + 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	*Polyrate* FILE NAME	
ch5fu30tr6.dat	poly.fu5	input data file
ch5fu30tr6.fu30	poly.fu30	input data file
ch5fu30tr6.fu6	poly.fu6	long output file

16.C.26. Test run ch5fu31tr1
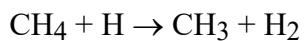
$CH_4 + H \rightarrow CH_3 + H_2$ IVTST-M-56/6 calculation using fu31

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

16.C.27. Test run ch5fu40tr1

$CH_3 + H_2 \rightarrow CH_4 + H$ TST reaction rate with frequencies read from fu40 directly

This is a conventional TST test run for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ [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 FREQUNIT = *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	*Polyrate* FILE NAME	
ch5fu40rt1.dat	poly.fu5	input data file
ch5fu40tr1.fu40	poly.fu40	input data file
ch5fu40tr1.fu6	poly.fu6	long output file

16.C.28. Test run ch5fu40tr2

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 $CH_3 + H_2 \rightarrow CH_4 + H$ [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	*Polyrate* FILE NAME	
ch5fu40tr2.dat	poly.fu5	input data file
ch5fu40tr2.fu40	poly.fu40	input data file
ch5fi40tr2.fu6	poly.fu6	long output file

16.C.29. Test run ch5fu40tr3

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 $CH_3 + H_2 \rightarrow CH_4 + H$ [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. 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	*Polyrate* FILE NAME	
ch5fu40tr3.dat	poly.fu5	input data file
ch5fu40tr3.fu40	poly.fu40	input data file
ch5fu40tr3.fu6	poly.fu6	long output file

16.C.30. Test run ch5fu40tr4

$CH_3 + H_2 \rightarrow CH_4 + H$ TST reaction rate with information read from fu40 and
frequencies input directly.

This is a conventional TST test run for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ [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 $V_{MEP}(s)$ has been scaled by a factor of 0.86 to be consistent with Ref. 48 in Section 22. 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*, FREQUNIT = *waven*]. The rate constants are calculated at twelve temperatures [TEMP].

Polyrate I/O FILES

FILE NAME	*Polyrate* FILE NAME	
ch5fu40tr4.dat	poly.fu5	input data file
ch5fu40tr4.fu40	poly.fu40	input data file
ch5fu40tr4.fu6	poly.fu6	long output file

16.C.31. Test run ch5fu40tr5

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 CH_3 + $H_2 \rightarrow CH_4$ + 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 $amu^{1/2}$ a_0) 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, 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 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 Sect. 21. Several errors were detected in Ref. 20 (see Section 24.B in this manual for the corresponding errata).

FILE NAME	*Polyrate* FILE NAME	
ch5fu40tr5.dat	poly.fu5	input data file
ch5fu40tr5.fu40	poly.fu40	input data file
ch5fu40tr5.fu6	poly.fu6	long output file

16.C.32. Test run ch5fu40tr6

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 CH_3 +
$H_2 \rightarrow CH_4 + H$. 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   *Polyrate* FILE NAME

        ch5fu40tr6.dat   poly.fu5   input data file
        ch5fu40tr6.fu40   poly.fu40   input data file
        ch5fu40tr6.fu6   poly.fu6   long output file

16.C.33. Test run ch5fu50tr1

$CH_3 + H_2 \rightarrow CH_4 + H$ zero-order IVTST reaction rate with information read from fu50 and frequencies input directly.

This is a conventional TST test run for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ [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 in put 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	*Polyrate* FILE NAME	
ch5fu50tr1.dat	poly.fu5	input data file
ch5fu50tr1.fu50	poly.fu50	input data file
ch5fu50tr1.fu6	poly.fu6	long output file

16.C.34. Test run ch5fu50tr2

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 $CH_3 + H_2 \rightarrow CH_4 + H$ 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

16.C.35. Test run ch5fu50tr3

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 $CH_3 + H_2 \rightarrow CH_4 + H$ 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

16.C.36. Test run ch5fu51tr1
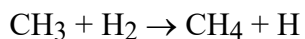
CH$_3$ + H$_2$ → CH$_4$ + H   VTST-ISPE calculation with MP2/6-31G** and J1

 This is a VTST-ISPE test run for the reaction CH$_3$ + H$_2$ → CH$_4$ + H using an analytic surface as the lower-level [dl and potential = hooks]. Electronic structure information calculated at the MPS/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	

16.C.37. Test run ch5icfu30tr1

CH$_3$ + H$_2$ → CH$_4$ + 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 CH$_3$ + H$_2$ → CH$_4$ + 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 $V_{MEP}(s)$ has been scaled by a factor of 0.86 to be consistent with Ref. 48 in Section 22. 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

16.C.38. Test run ch5icfu30tr2

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 $CH_3 + H_2 \rightarrow CH_4 + H$. 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 $V_{MEP}(s)$ has been scaled by a factor of 0.86 to be consistent with Ref. 48 in Section 22. 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

16.C.39. Test run ch5icfu31tr1
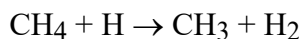
$CH_4 + H \rightarrow CH_3 + H_2$ VTST-IOC calculation using IVTST-M-56/6 data for the lower level of calculation and using the fu31 input file

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 $V_{MEP}(s)$ scaled by a factor of 0.86, as taken from Ref. 48 in Section 22. 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   *Polyrate* FILE NAME

    ch5fu31tr1.dat   poly.fu5   input data file
    ch5fu31tr1.fu31   poly.fu31   input data file
    ch5fu31tr1.fu50   poly.fu50   input data file
    ch5fu31tr1.fu6   poly.fu6   long output file
    ch5fu31tr1.fu15   poly.fu15   summary output file

16.C.40. Test run ch5icfu40tr1

$CH_3 + H_2 \rightarrow CH_4 + H$ 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 $CH_3 + H_2 \rightarrow CH_4 + H$. 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 $V_{MEP}(s)$ scaled by a factor of 0.86 taken from Ref. 48 in Section 22. 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

16.C.41. Test run ch5icfu40tr2

$CH_3 + H_2 \rightarrow CH_4 + H$ VTST-IOC calculation using file fu40 for the lower level information

This sample runuses 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 $CH_3 + H_2 \rightarrow CH_4 + H$. 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 $V_{MEP}(s)$ scaled by a factor of 0.86 taken from Ref. 48 in Section 22. 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

16.C.42. Test run clhbrtr1

Cl + HBr → HCl + 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 Cl + HBr → HCl + 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 $^2P_{3/2}$ and $^2P_{1/2}$ 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	*Polyrate* FILE NAME	
clhbrtr1.dat	poly.fu5	input data file
clhbrtr1.fu6	poly.fu6	long output file
clhbrtr1.fu14	poly.fu14	summary output file
clhbrtr1.fu15	poly.fu15	summary output file

16.C.43. Test run clhbrtr2

Cl + HBr → 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 respective 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

16.C.44. Test run clhbrtr3

Cl + HBr → HCl + 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 NGTHETA = 140. The
LCG4 tunneling probabilities are evaluated at NGAMP = 40 tunneling paths.

Polyrate I/O FILES

FILE NAME	*Polyrate* FILE NAME	
clhbrtr3.dat	poly.fu5	input data file
clhbrtr3.fu6	poly.fu6	long output file
clhbrtr3.fu14	poly.fu14	summary output file
clhbrtr3.fu15	poly.fu15	summary output file

16.C.45. Test run clhbrtr4

Cl + HBr → 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 in 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	*Polyrate* FILE NAME	
clhbrtr4.dat	poly.fu5	input data file
clhbrtr4.fu6	poly.fu6	long output file
clhbrtr4.fu14	poly.fu14	summary output file
clhbrtr4.fu15	poly.fu15	summary output file

16.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	*Polyrate* FILE NAME	
chbrtr5.dat	poly.fu5	input data file
chbrtr5.fu6	poly.fu6	long output file
chbrtr5.fu14	poly.fu14	summary output file
chbrtr5.fu15	poly.fu15	summary output file
chbrtr5.fu48	poly.fu48	LCT restart file

16.C.47. Test run cmctr1
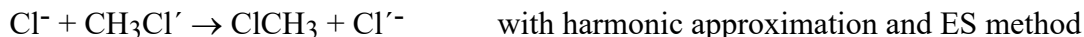
$Cl^- + CH_3Cl' \rightarrow ClCH_3 + Cl'^-$ with harmonic approximation and ES method

This sample run for the reaction $Cl^- + CH_3Cl' \rightarrow ClCH_3 + Cl'^-$ 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

16.C.48. Test run cmctr2

$Cl^- + CH_3Cl' \rightarrow ClCH_3 + Cl'^-$ 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 MEPTYPER to *well* and MEPTYPEP 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. 22]. 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 17.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

16.C.49. Test run cmctr3
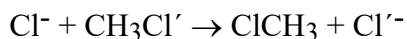
Cl⁻ + CH₃Cl′ → ClCH₃ + Cl′⁻   VTST-ISPE correction with MP2/6-31G**

This sample test run for the reaction $Cl^- + CH_3Cl' \rightarrow ClCH_3 + Cl'^-$ uses the default harmonic approximation for all vibrational modes. The MEP is followed by the Euler steepest-descents 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

16.C.50. Test run cwmctr1

Cl$^-$(H$_2$O) + CH$_3$Cl$'$ → ClCH$_3$ · Cl$'^-$(H$_2$O)   TST rate constants only
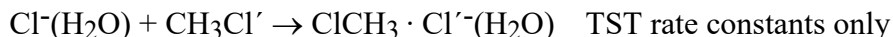
In this sample run the reactant, product, and saddle point properties for the reaction Cl$^-$(H$_2$O) + CH$_3$Cl$'$ → ClCH$_3$ · Cl$'^-$(H$_2$O), where Cl$^-$(H$_2$O) represents a chloride ion in the presence of a water molecule, are calculated, and only forward TST rate constants [TST, FORWARDK, NOCVT] 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		

16.C.51. Test run cwmctr2

$Cl^-(H_2O) + CH_3Cl' \rightarrow ClCH_3 \cdot Cl'^-(H_2O)$ with the hindered-internal-rotator approximation

In this sample run for the same reaction as in the cwmctrl 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

16.C.52. Test run cwmctr3

$Cl^-(H_2O) + CH_3Cl' \rightarrow ClCH_3 \cdot Cl'^-(H_2O)$ with the hindered-internal-rotator approximation

In this sample run for the same reaction as in the cwmctrl 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

16.C.53. Test run h3tr1

H$_2$ + H′ → HH′ + 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 H$_2$ + H′ → HH′ + 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 = *morseiiia*], 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	*Polyrate* FILE NAME	
	h3tr1.dat	poly.fu5	input data file
	h3tr1.fu6	poly.fu6	long output file
	h3tr1.fu14	poly.fu14	summary output file
	h3tr1.fu15	poly.fu15	summary output file

16.C.54. Test run h3tr2

$H_2 + H' \rightarrow HH' + H$ 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 $H_2 + H' \rightarrow HH' + H$ 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

16.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

16.C.56. Test run h3tr4

$H_2 + H' \rightarrow HH' + H$ 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

16.C.57. Test run ho2tr1

OH + O $\rightarrow$ HO$_2$ with Morse I approximation

In this sample run, CVT, μVT, and US rate constants [CVT, MUVT, MUVTOPT, US] are calculated at two temperatures [TEMP] for the reaction OH + O $\rightarrow$ HO$_2$. (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 μVT 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

16.C.58. Test run ho2tr2

OH + O → HO$_2$ with Morse I approximation using curvilinear coordinates

In this sample run, CVT, µVT, and US rate constants [CVT, MUVT, MUVTOPT, US] are calculated at two temperatures [TEMP] for the reaction OH + O → HO$_2$ 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

16.C.59. Test run ho2tr3

OH + O → HO$_2$ with WKB approximation for one mode

This sample run for the reaction OH + O → HO$_2$ 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

16.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

16.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 MEPTYPER 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 17.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

16.C.62. Test run nh3tr1

NH$_3$ → NH$_3$ (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 17.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	*Polyrate* FILE NAME	
nh3tr1.dat	poly.fu5	input data file
nh3ic.dat	poly.fu50	input data file
nh3tr1.fu6	poly.fu6	long output file

16.C.63. Test run nh3tr2

$NH_3 \rightarrow NH_3$ (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	*Polyrate* FILE NAME	
nh3tr2.dat	poly.fu5	input data file
nh3tr2.fu6	poly.fu6	input data file
nh3tr2.fu14	poly.fu14	summary output file
nh3tr2.fu15	poly.fu15	summary output file

16.C.64. Test run oh3tr1

OH + H$_2$ → H$_2$O + H   with harmonic approximation and using the VRP(ESD) algorithm

This sample run is for the OH + H$_2$ → H$_2$O + 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

16.C.65. Test run oh3tr2

OH + H_2 → H_2O + H   with Morse I-quadratic-quartic approximation

This sample run for the OH + H_2 → H_2O + 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

16.C.66. Test run oh3tr3

OH ($\upsilon = 0$) + H$_2$ $\rightarrow$ H$_2$O + H   with WKB and harmonic approximations

This sample run for the OH ($\upsilon = 0$) + H$_2$ $\rightarrow$ H$_2$O + 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

16.C.67. Test run oh3tr4

Adiabatic OH $(\upsilon = 0) + H_2(\upsilon = 1) \rightarrow H_2O + H$ 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 $(\upsilon = 0) + H_2(\upsilon = 1) \rightarrow H_2O + H$. 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 H_2 is restricted to the first excited state. Finally, for the product H_2O, 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

16.C.68. Test run oh3tr5

Diabatic OH + H$_2$(υ = 1) → H$_2$O + 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 + H$_2$(υ = 1) → H$_2$O + 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 H$_2$ vibration is restricted to the first excited state. The highest and second-highest frequency normal modes of the product H$_2$O 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

16.C.69. Test run oh3tr6

OH + H$_2$ → H$_2$O + H   with harmonic approximation and the ES1 integrator

This sample run is for the OH + H$_2$ → H$_2$O + 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

16.C.70. Test run oh3tr7

OH + H$_2$ → H$_2$O + 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 OH + H$_2$ → H$_2$O + 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

16.C.71. Test run oh3tr8

OH + H$_2$ $\rightarrow$ H$_2$O + 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 + H$_2$ $\rightarrow$ H$_2$O + 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" $\rightarrow$ 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	*Polyrate* FILE NAME	
oh3tr8.dat	poly.fu5	input data file
oh3tr8.fu6	poly.fu6	long output file
oh3tr8.fu14	poly.fu14	summary output file
oh3tr8.fu15	poly.fu15	summary output file

16.C.72. Test run oh3tr9

OH + H$_2$ → OH$_2$ + 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   *Polyrate* FILE NAME

       ch5fu31tr1.dat   poly.fu5   input data file
       ch5fu31tr1.fu31   poly.fu31   input data file
       ch5fu31tr1.fu50   poly.fu50   input data file
       ch5fu31tr1.fu6   poly.fu6   long output file
       ch5fu31tr1.fu15   poly.fu15   summary output file

16.C.73. Test run oh3tr10

OH + H$_2$ → H$_2$O + 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

16.C.74. Test run oh3fu30tr1

OH + H$_2$ → H$_2$O + 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 + H$_2$ → H$_2$O + 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

16.C.75. Test run oh3fu30tr2

OH + H$_2$ → H$_2$O + 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 + H$_2$ → H$_2$O + 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

16.C.76. Test run oh3fu30tr3

OH + H$_2$ → H$_2$O + 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 + H$_2$ → H$_2$O + 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

16.C.77. Test run oh3fu30tr4

OH + H$_2$ → H$_2$O + 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 OH + H$_2$ → H$_2$O + 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 amu$^{1/2}$ bohr [SSTEP].

Polyrate I/O FILES

FILE NAME	*Polyrate* FILE NAME	
oh3fu30tr4.dat	poly.fu5	input data file
oh3fu30tr4.fu30	poly.fu30	input data file
oh3fu30tr4.fu6	poly.fu6	long output file

16.C.78. Test run oh3fu30tr5

$OH + H_2 \rightarrow H_2O + H$ 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 $OH + H_2 \rightarrow H_2O + H$. 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   *Polyrate* FILE NAME

oh3fu30tr5.dat   poly.fu5   input data file
oh3fu30tr5.fu30   poly.fu30   input data file
oh3fu30tr5.fu6   poly.fu6   long output file

16.C.79. Test run oh3fu30tr6

OH + H$_2$ → OH$_2$ + 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   *Polyrate* FILE NAME

        oh3fu30tr6.dat   poly.fu5   input data file
        oh3fu30tr6.fu30   poly.fu30   input data file
        oh3fu30tr6.fu6   poly.fu6   long output file
        oh3fu30tr6.fu14   poly.fu14   summary output file
   oh3fu30tr6.fu15   poly.fu15   summary output file

16.C.80. Test run oh3fu31tr1

OH + H$_2$ → OH$_2$ + 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 goes beyond the values of s estimated for the stationary point (or points).

Polyrate I/O FILES
FILE NAME   *Polyrate* FILE NAME

     oh3fu31tr1.dat   poly.fu5   input data file
     oh3fu31tr1.fu31   poly.fu31   input data file
     oh3fu31tr1.fu6   poly.fu6   long output file
     oh3fu31tr1.fu15   poly.fu15   summary output file

16.C.81. Test run oh3fu31tr2

OH + DH → 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 + H_2 → H_2O + 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 → 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 + H_2 (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

16.C.82. Test run oh3fu40tr1

$OH + H_2 \rightarrow H_2O + H$ 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 [POTENTIAL = *unit40*] for the reaction $OH + H_2 \rightarrow H_2O + H$. Hessians are read in mass-scaled coordinates [HESSFORM = *msbohr*, HESSUNIT = *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 [COORD = *curv2*] used for the O-H + H'-H" $\rightarrow$ 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	
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

16.C.83. Test run oh3fu40tr2

OH + H$_2$ → H$_2$O + 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 + H$_2$ → H$_2$O +
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	*Polyrate* FILE NAME	
oh3fu40tr2.dat	poly.fu5	input data file
oh3fu40tr2.fu40	poly.fu40	input data file
oh3fu40tr2.fu6	poly.fu6	long output file
oh3fu40tr2.fu14	poly.fu14	summary output file
oh3fu40tr2.fu15	poly.fu15	summary output file

16.C.84. Test run oh3fu40tr3

OH + H_2 → H_2O + 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 + H_2 → H_2O + 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

16.C.85. Test run oh3fu40tr4

OH + H$_2$ → H$_2$O + 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 = *unit40*] for the reaction OH + H$_2$ → H$_2$O + 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	*Polyrate* FILE NAME	
oh3fu40tr4.dat	poly.fu5	input data file
oh3fu40tr4.fu40	poly.fu40	input data file
oh3fu40tr4.fu6	poly.fu6	long output file

16.C.86. Test run oh3fu40tr5

OH + H$_2$ → H$_2$O + 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 + H$_2$ → H$_2$O + 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	*Polyrate* FILE NAME	
oh3fu40tr5.dat	poly.fu5	input data file
oh3fu40tr5.fu40	poly.fu40	input data file
oh3fu40tr5.fu6	poly.fu6	long output file

16.C.87. Test run oh3fu40tr6

$OH + H_2 \rightarrow OH_2 + H$ 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   *Polyrate* FILE NAME

oh3fu40tr6.dat	poly.fu5	input data file
oh3fu40tr6.fu40	poly.fu40	input data file
oh3fu40tr6.fu6	poly.fu6	long output file
oh3fu40tr6.fu14	poly.fu14	summary output file
oh3fu40tr6.fu15	poly.fu15	summary output file

16.C.88. Test run hnitr1

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

hnitr1.dat   poly.fu5   input data file
hnitr1.fu6   poly.fu6   long output file
hnitr1.fu14   poly.fu14   summary output file
hnitr1.fu15   poly.fu15   summary output file

POTENTIAL ENERGY SURFACE DATA FILE

pothni.dat   UNIT = 4
                                    FILE NAME = pothni.dat

16.C.89. Test run h2nitr1

H$_2$(a)/Ni(100) $\rightarrow$ 2H(a)/Ni(100)   with harmonic approximation and ESD integrator

This sample run is for the unimolecular gas-solid-interface reaction H$_2$(a)/Ni(100) $\rightarrow$ 2H(a)/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

16.C.90. Test run ch4ohtr1

OH + CH$_4$ → H$_2$O + CH$_3$ 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 + CH$_4$ → H$_2$O + CH$_3$ 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

16.C.91. Test run ch4ohtr2

OH + CH$_4$ $\rightarrow$ H$_2$O + 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 OH + CH$_4$ $\rightarrow$ H$_2$O + CH$_3$ 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	*Polyrate* FILE NAME	
ch4ohtr2.dat	poly.fu5	input data file
ch4ohtr2.fu29	poly.fu29	input data file
ch4ohtr1.fu6	poly.fu6	output file

16.C.92. Test run ch4ohtr3

OH + CH$_4$ → H$_2$O + CH$_3$ 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 + CH$_4$ → H$_2$O + CH$_3$ 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. This sample run is the same as the ch4ohtr1 test run, except that the lowest-frequency vibrational mode of the generalized transition state is treated as a hindered rotor using the CO torsion option.

Polyrate I/O FILES

FILE NAME	*Polyrate* FILE NAME	
ch4ohtr3.dat	poly.fu5	input data file
ch4ohic.dat	poly.fu50	input data file
ch4ohtr3.fu6	poly.fu6	long output file
ch4ohtr3.fu14	poly.fu14	summary output file
ch4ohtr3.fu15	poly.fu15	summary output file

16.C.93. Test run ch4ohtr4

OH + CH$_4$ → H$_2$O + CH$_3$ 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 + CH$_4$ → H$_2$O + CH$_3$ 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. This sample run is the same as the ch4ohtr1 test run, except that the lowest-frequency vibrational mode of the generalized transition state is treated as a hindered rotor using the CW torsion option.

Polyrate I/O FILES

FILE NAME	*Polyrate* FILE NAME	
ch4ohtr4.dat	poly.fu5	input data file
ch4ohic.dat	poly.fu50	input data file
ch4ohtr4.fu6	poly.fu6	long output file
ch4ohtr4.fu14	poly.fu14	summary output file
ch4ohtr4.fu15	poly.fu15	summary output file

16.C.94. Test run ch4cltr1

$CH_4 + Cl \rightarrow HCl + CH_3$ zero-order IVTST calculation

This sample run uses the IVTST0 keyword for a zero-order IVTST (*i.e.*, IVTST-0) calculation of the reaction $CH_4 + Cl \rightarrow HCl + CH_3$ 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 $^2P_{3/2}$ and $^3P_{1/2}$ 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

16.C.95. Test run ch4cltr2

$CH_4 + Cl \rightarrow HCl + CH_3$ first-order IVTST calculation

This sample run uses the IVTST1 keyword for a first-order IVTST (*i.e.*, IVTST-1) calculation of the reaction $CH_4 + Cl \rightarrow HCl + CH_3$ 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 $^2P_{3/2}$ and $^3P_{1/2}$ 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

16.C.96. Test run ch4otr1

CH$_4$ + O → CH$_3$ + 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   *Polyrate* FILE NAME

  ch4otr1.dat   poly.fu5 input data file
  ch4otr1.fu6   poly.fu6 long output file
  ch4otr1.fu14   poly.fu14 summary output file
  ch4otr1.fu15   poly.fu15 summary output file

POTENTIAL ENERGY SURFACE DATA FILE
  potch4o.dat   UNIT = 4
                FILE NAME = potch4o.dat

16.C.97. 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

16.C.98. Test run ohcltr2

O + HCl → OH + Cl   ICVT calculation using WKB-on-the-true-potential for ground-state energies and Morse 1 anharmonicity for excited stateenergies.

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	*Polyrate* FILE NAME	
ohcltr2.dat	poly.fu5	input data file
ohcltr2.fu6	poly.fu6	long output file
ohcltr2.fu14	poly.fu14	summary output file
ohcltr2.fu15	poly.fu15	summary output file

16.C.99. Test run ohcltr3

O + HCl(v=1) → 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

16.C.100. Test run ohcltr4

O + HCl(v=1) → 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 statebend energies.

This sample run is the same as ohcltr3, except that stretch vibraton 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

16.C.101. Test run ohcltr5

O + HCl → OH + Cl   ICVT and MUCVT calculation using Morse 1
                           anharmonicity and internal coordinates

This sample run is the same as ohcltr1, except that microcanonical variational theory (MUCVT) rate constants are also calculated. The primary and secondary fits use 1-point and 3-point fitting [*MUVTOPT, FIT=GAS13*].

Polyrate I/O FILES

FILE NAME	*Polyrate* FILE NAME	
ohcltr5.dat	poly.fu5	input data file
ohcltr5.fu6	poly.fu6	long output file
ohcltr5.fu14	poly.fu14	summary output file
ohcltr5.fu15	poly.fu15	summary output file

16.C.102. Test run ch3tr1
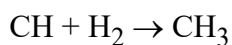
CH + H$_2$ → CH$_3$ VRC-VTST calculation

This sample run calculates the variable reaction coordinate variational transition state theory rate constants by canonical variational theory, microcanonical variational theory, and total energy and angular momentum resolved microcanonical variational theory. One pivot point is assigned to each reactant. Pivot point for CH is assigned as the C atom, and pivot point for H$_2$ is the center of mass of H$_2$. The range of reaction coordinate s is [1.3, 6.5] Å with a step size 0.2 Å. The number of Monte Carlo samples (keyword *NMC*) is set to 100 for each dividing surface. Note that this number should be set to around 1000 in a production run in order to have a much lower uncertainty of the Monte Carlo integration.

Polyrate I/O FILES

FILE NAME	*Polyrate* FILE NAME	
ch3tr1.dat	poly.fu5	input data file
ch3tr1.fu6	poly.fu6	long output file
ch3tr1.fu15	poly.fu15	summary output file

16.C.103. Test run h3o2fu31tr1
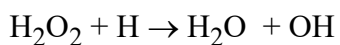
$H_2O_2 + H \rightarrow H_2O + OH$ TST with torsional treatment for H_2O_2 using the segmented-reference Pitzer-Gwinn (SRPG) method

This sample run calculates rate constants by conventional TST. This run is used to show the usage of the SRPG method for treating a torsion. The potential energy surface is provided by the h3o2fu31tr1.fu31 file.

Polyrate I/O FILES

FILE NAME	*Polyrate* FILE NAME	
h3o2fu31tr1.dat	poly.fu5	input data file
h3o2fu31tr1.fu31	poly.fu31	input data file
h3o2fu31tr1.fu6	poly.fu6	long output file
h3o2fu31tr1.fu15	poly.fu15	summary output file

16.C.104. Test run h3o2fu31tr2
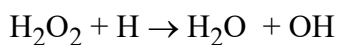
$H_2O_2 + H \rightarrow H_2O + OH$ TST with torsional treatment for H_2O_2 using the segmented-reference Alyala-Schlegel (SAS) method

This sample run calculates rate constants by conventional TST. This run is used to show the usage of the SAS method for treating a torsion. The potential energy surface is provided by the h3o2fu31tr2.fu31 file.

Polyrate I/O FILES

FILE NAME	*Polyrate* FILE NAME	
h3o2fu31tr2.dat	poly.fu5	input data file
h3o2fu31tr2.fu31	poly.fu31	input data file
h3o2fu31tr2.fu6	poly.fu6	long output file
h3o2fu31tr2.fu15	poly.fu15	summary output file

16.C.105. Test run h3o2fu31tr3
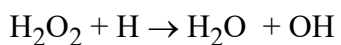
$H_2O_2 + H \rightarrow H_2O + OH$ TST with torsional treatment for H_2O_2 using the reference Pitzer-Gwinn (RPG) method

This sample run calculates rate constants by conventional TST. This run is used to show the usage of the RPG method for treating a torsion. The potential energy surface is provided by the h3o2fu31tr3.fu31 file.

Polyrate I/O FILES

FILE NAME	*Polyrate* FILE NAME	
h3o2fu31tr3.dat	poly.fu5	input data file
h3o2fu31tr3.fu31	poly.fu31	input data file
h3o2fu31tr3.fu6	poly.fu6	long output file
h3o2fu31tr3.fu15	poly.fu15	summary output file

16.C.106. Test run h3o2fu31tr4
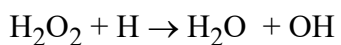
$H_2O_2 + H \rightarrow H_2O + OH$ TST with torsional treatment for H_2O_2 using the Alyala-Schlegel (AS) method

This sample run calculates rate constants by conventional TST. This run is used to show the usage of the AS method for treating a torsion. The potential energy surface is provided by the h3o2fu31tr4.fu31 file.

Polyrate I/O FILES

FILE NAME	*Polyrate* FILE NAME	
h3o2fu31tr4.dat	poly.fu5	input data file
h3o2fu31tr4.fu31	poly.fu31	input data file
h3o2fu31tr4.fu6	poly.fu6	long output file
h3o2fu31tr4.fu15	poly.fu15	summary output file

16.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	ch5j1tr12
ch5j2tr1	ch5j2tr2	ch5j2tr3	clhbrtr1	clhbrtr2	clhbrtr3
clhbrtr4	clhbrtr5	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	ohcltr5	ch3tr1

fu30

ch5fu30tr1	ch5fu30tr2	ch5fu30tr3	ch5fu30tr4	ch5fu30tr5	ch5fu30tr6
oh3fu30tr1	oh3fu30tr2	oh3fu30tr3	oh3fu30tr4	oh3fu30tr5	oh3fu30tr6

fu31

ch5fu31tr1	oh3fu31tr2	ohfu31tr1	h3o2fu31tr1	h3o2fu31tr2	h3o2fu31tr3
h3o2fu31tr4					

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

17. DESCRIPTION OF FILES IN RELEASE OF VERSION 17

The source code is partitioned into 33 FORTRAN files. Note that a full *Polyrate* source consists of everything in the 33 FORTRAN files that including modules and parameters. A description of each *Polyrate* subprogram is given in Subsection 17.A.1. Subsection 17.A.2 briefly describes 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*, or *unit40* are in dumpot.f.) The input and output files for the sample runs are explained in Subsection 16.C. The Unix C shell scripts are explained in Subsection 15.A and Section 18. A list of all the files in the distribution package can be found in Subsection 15.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 15, except that this manual may be in a separate file (not tarred with everything else).

17.A. FORTRAN files and subprograms, MODULE files, and PES subroutines

Subsection 17.A.1 contains an alphabetical list and brief descriptions of all the subprograms that make up the *Polyrate* program, Subsection 17.A.2 describes the MODULE files and the user-supplied potential energy subprograms, and Subsection 17.A.3 describes the FORTRAN files.

17.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. 48 in Sect. 21 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 $V_{MEP}(s)$ 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.

AVERJ

Calculate an integral over all possible orientation of the angular momentum vector $\mathbf{J}$ with its length fixed in the evaluation of the number of available states.

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 determining 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.

BTENTR

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. 22.

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 $V_{MEP}(s)$ 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.

CENTR1

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.

CROSSPROD

Calculate cross product of two vectors.

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.

DEFALT

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 calculations.

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 computs 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 ($\text{IOP} = 1$) or the ground-state vibrationally adiabatic energy curve ($\text{IOP} = 2$).

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 following 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 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 **A** to square matrix **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 x 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

Calculates 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

Forms the product $\mathbf{A}^T\mathbf{FA}$, where $\mathbf{A}$ is the A-matrix and $\mathbf{F}$ is the Cartesian force constant matrix.

FORMF2

Forms the second term of curvilinear internal force constant matrix.

FORMG

Forms 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 $\mathbf{Av} = \mathbf{c}$, 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 gyrator.

HQSC

Semiclassical solution of a harmonic-quartic potential.

HRPART

Calculates the partition functions using the hindered-internal-rotor 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 connectivies 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.

INITVRC

Initializes the array IATSV(I,J) for the generalized transition state in variable reaction coordinate calculations because START section is not give in the fu5 file.

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 $P^{LCG34}(E)$ 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 $T_{tun}(\tilde{s}_0, n_p)$ 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 $\tilde{s}_0$ 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.

PMOI

Calculate priccipal moment of inertia and their unit vector along the ith principal axis for each reactant or complex of reactants

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. 22 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 x 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.

RAN1

Random number generator from Numerical Recipes.

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.

REACTCOM

Calculate center of mass for reactants

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 μVT 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. 48 in Sect. 21.

RPIVOT

      Reads the Cartesian coordinates of pivot points

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.

RSVRC

Reads the range and step size of variable reaction coordinate

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.

RVRCOPT

 Reads the options for variable reaction coorindate calculations.

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.

SAVEXRP
  Calculate Cartesian coordinate of each reactant and the other pivot points relative to the first pivot point of reactant 1.

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 n×2 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, $V_{MEP}(s)$ 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, $V_{MEP}(s)$, 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 an 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.

VRCNEJ

Calculate number of available states for transitional modes using variable reaction coorindate algorithm at canonical, microcanonical, and E,J -resolved microcanonical levels.

VRCTST

Main subroutine to calculate reaction rate for loose transition state using variable reaction coordinate and single-faceted or multifaceted dividing surface.

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.

17.A.2 User-supplied subprograms and module files

MODULE files:

aamod.f90

This FORTRAN 90 file contains all the modules used for passing information between the various subprograms. It also contains all the permanent constants for the *Polyrate* code. These parameters should not be changed.

alloc.f90

This FORTRAN 90 file dynamically allocates memory for arrays according to the user input file.

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).

17.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.

headr.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.

main_vrcmpi.F

        This file contains the driver for polyatomic VRC-TST calculations. It is a substitution of main.F file.

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.

vrctst_mpi.F

This file contains subprograms that are used to calculate reaction rate for loose transition states using variable reaction coordinates and single-faceted or multifaceted dividing surfaces. These subprograms are parallelized using MPI.

17.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 *POLYRATE17* 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 12.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 15 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
```

17.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 using rectilinear model. 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 *Polyrate17*/util directory and in Refs. 24, 29, and 30 in Sect. 21. A sample input file, cwmc.rmi, is also provided in the *Polyrate17*/util directory.

You can use the compiling and linking commands listed in the Section 15 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* cwmctrl 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
    ```

17.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 *Polyrate17*/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 15 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].

17.E. Utility program rotate30.f, rotate31.f, and rotate40.f

ROTATEx (x = 30, 31, 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 22.

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.

17.F. Utility program *SS-QRRK*

The utility program directory SS-QRRK contains one .exe file, one source code file (.f90), one directory for test runs, and one directory for outputs of test runs. *SS-QRRK* code is a FORTRAN90 code for carrying out system-specific quantum RRK calculations. Please refer to a separate manual for the use of *SS-QRRK* code.

18. UNIX C SHELL SCRIPTS AND PERL SCRIPTS

Polyrate includes many useful scripts for users. The following subsections discuss the functions of these scripts and explain how to use them.

18.A. Installation script

This script, *configure*, resides in the *Polyrate17* directory. It is used to install the *Polyrate* program on your computer. Note that this installation is only for generating the Makefile, and no executable is generated in this process. Usually, the first thing you do after uncompressing and untarring the distribution tar file is to run this script. (See Subsection 15.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 *Polyrate17* directory in the file system you are using. This lets other scripts in *Polyrate* 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. Asks the user to choose which kind of executable the user wants the Makefile to generate: RP-VTST or VRC-VTST. (If you want both, you must run the script twice.)

3. 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.

4. 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 *Polyrate17*/util directory to the *Polyrate17*/src directory and names it dattim.f.

    If VRC-VTST installation is selected by the user, the script will install the SPRNG 2.0 library. If RP-VTST installation is chosen, the script will skip the SPRNG installation step.

5. 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 15.A.1 for a description of the directory structure following a successful installation.

18.B. Potential energy surface compiling scripts

This script is distributed in the current version, but it is no longer needed when using Makefile.

There are two potential energy surface subroutine compiling scripts in the *Polyrate17*/script directory. The script comp_pot.jc is used to compile all the potential energy surface subroutines for the test runs. (See Subsection 15.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 *Polyrate17*/poten directory. You then change to the *Polyrate17*/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 *Polyrate17*/obj directory. You can, if you wish, copy the script into the *Polyrate17*/poten directory so you can use it more conveniently. Remember that you must put your potential source code in the *Polyrate17*/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.

18.C. Source code compiling script

This script is distributed in current, but it is no longer needed when using Makefile.

The script comp_src.jc in the 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 15.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 src/ directory, change to the 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 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 src/ directory and always name a new parameter file param#.inc where # can be any integer greater than 4.

18.D. Linking script

This script is distributed in the current version, but it is no longer needed when using Makefile.

The script link_exe.jc in the *Polyrate17*/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 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 exe/ directory.

18.E. Test run scripts

The test run scripts have been discussed in Subsection 15.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 Sect. 21) 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.

19. 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 file aamod.f90 (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.

Unit	Usage
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 12.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 16.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 = *unit 30* in the ENERGETICS Section of file fu5
fu31	Keyword-based electronic structure input file used if POTENTIAL = *unit31* in the ENERGETICS section of file fu5.
fu40	Keyword-based electronic structure input file used if POTENTIAL = *unit40* 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 *ioc* or ICOPT]
fu51	Input file for higher-level data for VTST-ISPE calculations [DL keyword set to *ispe*]

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 *none*

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

20. CITATIONS INDICATED BY BRACKETS IN THE 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 Molecule*s (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.

[61] T. Yu, J. Zheng, and D. G. Truhlar, Chem. Sci. 2 (2011) 2199.

[62] T. Yu, J. Zheng, and D. G. Truhlar, J. Phys. Chem. A 116 (2012) 297.

[63] J. Zheng and D. G. Truhlar, Faraday Discuss. 157 (2012) 59.

[64] J. Zheng, S. L Mielke, K. L Clarkson, and D. G. Truhlar, Computer Phys. Commun. 183 (2012) 1803.

[65] J. Zheng, R. Meana-Pañeda, and D. G. Truhlar, Computer Phys. Commun.184 (2013) 2032.

[66] J. Zheng, S. L Mielke, K. L Clarkson, R. Meana-Pañeda, and D. G. Truhlar, MSTOR, (University of Minnesota, Minneapolis, 2014). http://comp.chem.umn.edu/mstor

[67] A. Fernández-Ramos, B. A. Ellingson, R. Meana-Pañeda, J. M. C. Marques, and D. G. Truhlar, Theor. Chem. Acc. 118 (2007) 813.

[68] J. Zheng and D. G. Truhlar, J. Chem. Theory Comput. 9 (2013) 2875.

[69] I. M. Alecu, J. Zheng, Y. Zhao, and D. G. Truhlar, J. Chem. Theory Comput. 6 (2010) 2872.

[70] J. Zheng, T. Yu, E. Papajak, I, M. Alecu, S. L. Mielke, and D. G. Truhlar, Phys. Chem. Chem. Phys. 13 (2011) 10885.

[71] J. Zheng and D. G. Truhlar, J. Chem. Theory Comput. 9 (2013) 1356.

[72] J. L. Bao, R. Meana-Pañeda, and D. G. Truhlar, Chem. Sci. 6 (2015), 5866.

[73] J. L. Bao, J. Zheng, and D. G. Truhlar, J. Am. Chem. Soc. 138 (2016), 2690.

[74] J. L. Bao, X. Zhang, and D. G. Truhlar, Phys. Chem. Chem. Phys. 18 (2016), 16659.

[75] "Barrierless association of CF_2 and dissociation of C_2F_4 by variational transition state theory and system-specific quantum Rice–Ramsperger–Kassel theory," J. L. Bao, X. Zhang, and D. G. Truhlar, Proceedings of the National Academy of Sciences, U.S.A., 113, (2016), 13606.

21. BIBLIOGRAPHY

Note : See Sect. 20 for references indicated by brackets in the text of the manual.

The following is a list of some publications reporting calculations carried out with *Polyrate* or with its very much older 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 OH + $H_2 \rightarrow H_2O$ + H", A. D. Isaacson and D. G. Truhlar, J. Chem. Phys. **76,** 1380 (1982).

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,** 2252 (1982); erratum: **87,** 4554 (1983).

3. "Statistical-Diabatic Model for State-Selected Reaction Rates. Theory and Application of Vibrational-Mode Correlation Analysis to OH(n_{OH}) + H_2(n_{HH}) $\rightarrow$ H_2O + H", D. G. Truhlar and A. D. Isaacson, J. Chem. Phys. **77,** 3516 (1982).

4. "Variational Transition State Theory Calculations for an Atom-Radical Reaction with no Saddle Point: O + OH", S. N. Rai and D. G. Truhlar, J. Chem. Phys. **79,** 6049 (1983).

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: OH + $H_2 \rightarrow H_2O$ + H", A. D. Isaacson, M. T. Sund, S. N. Rai, and D. G. Truhlar, J. Chem. Phys. **82,** 1338 (1985).

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,** 4451 (1985).

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,** 558 (1985).

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,** 1843 (1986).

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**, 2252 (1985).

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 H_2", B. C. Garrett, D. G. Truhlar, A. J. C. Varandas, and N. C. Blais, International J. Chem. Kinetics **18,** 1065 (1986).

12A. "Evaluation of Dynamical Approximations for Calculating the Effect of Vibrational Excitation on Reaction Rates", B. C. Garrett, D. G. Truhlar, J. M. Bowman, and AF Wagner, J. Phys. Chem. **90,** 4305 **(**1986).

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,** 4997 (1986).

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,** 91 (1987).

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,** 6229 (1987).

16. "A Comparative Study of Potential Energy Surfaces for $CH_3 + H_2 \rightarrow CH_4 + H$", R. Steckler, K. Dykema, F. B. Brown, D. G. Truhlar, and T. Valencich, J. Chem. Phys. **87,** 7024 (1987).

17. "A New Potential Energy Surface for the $CH_3 + H_2 \rightarrow CH_4 + H$ 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,** 7036 (1987).

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. Baldridge, and D. Bartol, J. Phys. Chem. **92,** 1476 (1988).

19. "Calculation of Reaction Rates and Kinetic Isotope Effects for Dissociative Chemisorption of H_2 and D_2 on Ni(100), Ni(110) and Ni(111) Surfaces", T. N. Truong, G. Hancock, and D. G. Truhlar, Surface Science **214,** 523 (1989).

20. "*Ab Initio* Reaction Paths and Direct Dynamics Calculations", K. K. Baldridge, M. S. Gordon, R. Steckler, and D. G. Truhlar, J. Phys. Chem. **93,** 5107 (1989).

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,** 271 (1989); erratum: **140,** 207 (1990).

22. "Transition State Structure, Barrier Height, and Vibrational Frequencies for the Reaction $Cl + CH_4 \rightarrow CH_3 + HCl$", T. N. Truong, D. G. Truhlar, K. K. Baldridge, M. S. Gordon, and R. Steckler, J. Chem. Phys. **90,** 7137 (1989).

23. "Projection Operator Method for Geometry Optimization with Constraints", D.-h. Lu, M. Zhao, and D. G. Truhlar, J. Comp. Chem. **12,** 376 (1991).

24. "A Simple Approximation for the Vibrational Partition Function of a Hindered Internal Rotation", D. G. Truhlar, J. Comp. Chem. **12**, 266 (1991).

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,** 7875 (1991).

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,** 8875 (1991).

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,** 5758 (1992).

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. Gonzàlez-Lafont, T. N. Truong, D. Maurice, Y-.P. Liu, and G. C. Lynch, ACS Symp. Series **502,** 16 (1992).

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,** 235 (1992).

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,** 143 (1993).

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, D.-h. Lu, and D. G. Truhlar, J. Amer. Chem. Soc. **115,** 2408 (1993).

32. "Interpolated Variational Transition State Theory and Tunneling Calculations of the Rate Constant of the Reaction $OH + CH_4$ at 223-2400 K", V. S. Melissas and D. G. Truhlar, J. Chem. Phys. **99,** 1013 (1993).

33. "Bond-Distance and Bond-Angle Constraints in Reaction-Path Dynamics Calculations", D.-h. Lu and D. G. Truhlar, J. Chem. Phys. **99,** 2723 (1993).

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,** 7806 (1993).

35. "Effect of Phonon Coupling on Hydrogen Tunneling Rates at Gas-Surface Interfaces", S. E. Wonchoba and D. G. Truhlar, J. Chem. Phys. **99,** 9637 (1993).

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,** 1715 (1994).

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,** 9985 (1995).

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,** 341 (1995).

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,** 3188 (1995).

42. "Dual-Level Reaction Path Dynamics (The /// Approach to VTST with Semiclassical Tunneling). Application to OH + $NH_3 \rightarrow H_2O + NH_2$ " J.C. Corchado, J. Espinosa-Garcia, W.-P. Wu, I. Rossi, and D. G. Truhlar, J. Phys. Chem. **99,** 687 (1995).

43. "Reaction-Path Dynamics in Curvilinear Internal Coordinates Including Torsions," K. A. Nguyen, C. F. Jackels, and D. G. Truhlar, J. Chem. Phys. **104,** 6491 (1996).

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,** 3808 (1997); erratum: **101**, 8741 (1997).

45. "Reaction-Path Dynamics with Harmonic Vibrational Frequencies in Curvilinear Internal Coordinates: H + *trans*-$N_2H_2 \rightarrow H_2 + N_2H$", Y.-Y. Chuang and D. G. Truhlar, J. Chem. Phys. **107,** 83 (1997).

46. "Reaction Path Dynamics in Redundant Internal Coordinates," Y.-Y. Chuang and D. G. Truhlar, J. Phys. Chem. A **102,** 242 (1998).

47. "Variation Transition State Theory Without the Minimum-Energy Path," J. Villà and D. G. Truhlar, Theor. Chem. Acc. **97,** 317 (1997).

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,** 2424 (1998).

49. "Variational Reaction Path Algorithm," P. L. Fast and D. G. Truhlar, J. Chem. Phys. **109,** 3721 (1998).

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,** 6237 (1998).

51. "Interface of Electronic Structure and Dynamics for Reactions in Solution," Y.-Y. Chuang, C. J. Cramer, and D. G. Truhlar, International Journal of Quantum Chemistry **70**, 887-896 (1998). (Sanibel issue) doi.org/10.1002/(SICI)1097-461X(1998)70:4/5<887::AID-QUA34>3.0.CO;2-W

52. "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,** 1140 (1999).

53. "Statistical Thermodynamics of Bond Torsional Modes", Y.-Y. Chuang and D. G. Truhlar, J. Chem. Phys. **112,** 1221 (2000); **121**, 7036(E) (2004); **124**, 0000(E) (2006).

54. "Nonequilibrium Solvation Effects for a Polyatomic Reaction in Solution", Y.-Y. Chuang and D. G. Truhlar, J. Amer. Chem. Soc., **121,** 10157 (1999).

55. "Improved Algorithm for Corner Cutting Calculations", A. Fernandez-Ramos and D. G. Truhlar, J. Chem. Phys. **114,** 1491 (2001).

56. "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,** 4957 (2002).

57. "Tests of Potential Energy Surfaces for H + CH4 <-> CH3 + H2: Deuterium and Muonium Kinetic Isotope Effects for the Forward and Reverse Reaction," J. Pu and D. G. Truhlar, J. Chem. Phys. **117**, 10675-10687 (2002).

58. "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**, 4112-4124 (2004).

59. "Temperature Dependence of Carbon-13 Kinetic Isotope Effects of Importance to Global Climate Change," H. Lin, Y. Zhao, B. A. Ellingson, J. Pu, and D. G. Truhlar, Journal of the American Chemical Society **127**, 2830-2831 (2005).

60. "A New Algorithm for Efficient Direct Dynamics Calculations of Large- Curvature Tunneling and Its Application to Radical Reactions with 9-15 Atoms," A. Fernández-Ramos and D. G. Truhlar, Journal of Chemical Theory and Computation **1**, 1063-1078 (2005).

61. "Variational Transition State Theory," B. C. Garrett and D. G. Truhlar, in *Theory and Applications of Computational Chemistry: The First Forty Years*, edited by C. E. Dykstra, G. Frenking, K. S. Kim, and G. E. Scuseria (Elsevier, Amsterdam, 2005), pp. 67-87.

62. "Variational Transition State Theory and Multidimensional Tunneling for Simple and Complex Reactions in the Gas Phase, Solids, Liquids, and Enzymes," D. G. Truhlar, in *Isotope Effects in Chemistry and Biology*, edited by A. Kohen and H.-H. Limbach (Marcel Dekker, Inc., New York, 2006), pp. 579-620.http://t1.chem.umn.edu/Truhlar/docs/C71.pdf

63. "Statistical Thermodynamics of Bond Torsional Modes. Tests of Separable, Almost-Separable and Improved Pitzer-Gwinn Approximations", B. A. Ellingson, V. A. Lynch, S. L. Mielke, and D. G. Truhlar, Journal of Chemical Physics **125**, 084305/1-17 (2006).

64. "Multidimensional Tunneling, Recrossing, and the Transmission Coefficient for Enzymatic Reactions," J. Pu, J. Gao, and D. G. Truhlar, Chemical Reviews **106**, 3140-3169 (2006).

65. "Multi-Configuration Molecular Mechanics Based on Combined Quantum Mechanical and Molecular Mechanical Calculations," H. Lin, Y. Zhao, O. Tishchenko, and D. G. Truhlar, Journal of Chemical Theory and Computation **2**, 1237-1254 (2006).

66. "Optimizing the Performance of the Multiconfiguration Molecular Mechanics Method," O. Tishchenko and D. G. Truhlar, Journal of Physical Chemistry A **110**, 13530-13536 (2006).

67. "Transition State Theory and Chemical Reaction Dynamics in Solution," D. G. Truhlar and J. R. Pliego Jr., in *Continuum Solvation Models in Chemical Physics: Theory and Application*, edited by B. Mennucci and R. Cammi (Wiley), submitted Oct. 6, 2006.

68. "Modeling of Bimolecular Reactions," A. Fernández-Ramos, J. A. Miller, S. J. Klippenstein, and D. G. Truhlar, Chemical Reviews **106**, 4518-4584 (2006).

69. "Variational Transition State Theory in the Treatment of Hydrogen Transfer Reactions, " D. G. Truhlar and B. C. Garrett, in *Hydrogen Transfer Reactions*, edited by J. T. Hynes, J. P. Klinman, H. H. Limbach, and R. L. Schowen (Wiley-VCH, Weinheim, Germany, 2007), Vol. 2, pp. 833-874.

70. "Hydrogen Atom Transfers in B12 Enzymes," R. Banerjee, D. G. Truhlar, A. Dybala-Defratyka, P. Paneth, in Hydrogen Transfer Reactions, edited by J. T. Hynes, J. P. Klinman, H.-H. Limbach, and R. L. Schowen (Wiley-VCH, Weinheim, Germany, 2007), Vol. 4, pp. 1473-1495.

71. "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*, Vol. 23, edited by K. B. Lipkowitz and T. R. Cundari (Wiley-VCH, Hoboken, NJ, 2007), pp. 125-232.

72. "Global Potential Energy Surfaces with Correct Permutation Symmetry by Multi-Configuration Molecular Mechanics," O. Tishchenko and D. G. Truhlar, Journal of Chemical Theory and Computation **3**, 938-948 (2007). doi.org/10.1021/ct600315h

73. "Symmetry Numbers and Chemical Reaction Rates," A. Fernández-Ramos, B. A. Ellingson, R. Meana-Pañeda, J. M. C. Marques and D. G. Truhlar, Theoretical Chemistry Accounts **118**, 813-826 (2007). doi.org/10.1007/s00214-007-0328-0

74. "Multi-Coefficient Gaussian-3 Calculation of the Rate Constant for the OH + CH_4 Reaction and its $^{12}C/^{13}C$ Kinetic Isotope Effect with Emphasis on the Effects of Coordinate System and Torsional Treatment," B. A. Ellingson, J. Pu, H. Lin, Y. Zhao, and D. G. Truhlar, Journal of Physical Chemistry A **111**, 11706-11717 (2007). doi.org/10.1021/jp072843j

75. "Explanation of the Unusual Temperature Dependence of the Atmospherically Important OH + H_2S → H_2O + SH Reaction and Prediction of the Rate Constant at Combustion Temperatures," B. A. Ellingson, and D. G. Truhlar, Journal of the American Chemical Society **129**, 12765-12771 (2007). doi.org/10.1021/ja072538b

76. "Reactions of Hydrogen Atom with Hydrogen Peroxide," B. A. Ellingson, D. P. Theis, O. Tishchenko, J. Zheng, and D. G. Truhlar, J. Phys. Chem. A **111**, 13554-13566 (2007). doi.org/10.1021/jp077379x

77. "Density Functional Study of Methyl Radical Association Reaction Kinetics," J. Zheng, S. Zhang, and D. G. Truhlar, Journal of Physical Chemistry A **112**, 11509-11513 (2008). dx.doi.org/10.1021/jp806617m

78. "Mechanistic Analysis of the Base-Catalyzed HF Elimination from 4-Fluoro-4-(4'-nitrophenyl)butane-2-one Based on Liquid-Phase Kinetic Isotope Effects Calculated by Dynamics Modeling with Multidimensional Tunneling," Y. Kim, A. V. Marenich, J. Zheng, K. H. Kim, M. Kołodziejska-Huben, M. Rostkowski, P. Paneth, and D. G. Truhlar, Journal of Chemical Theory and Computation **5**, 59-67 (2009). dx.doi.org/10.1021/ct800345j

79. "Critical Role of Substrate Conformational Change in the Proton Transfer Process Catalyzed by 4-Oxalocrotonate Tautomerase," J. Ruiz-Pernía, M. Garcia-Viloca, S. Bhattacharyay, J. Gao D. G. Truhlar, and I. Tuñon, Journal of the American Chemical Society **131**, 2687-2698 (2009). dx.doi.org/10.101/ja8087423

80. "Direct Dynamics Study of Hydrogen-Transfer Isomerization of 1-Pentyl and 1-Hexyl Radicals," J. Zheng and D. G. Truhlar , Journal of Physical Chemistry A **113**, 11919-11925 (2009). dx.doi.org/10.1021/jp903345x

81. "Phase Space Prediction of Product Branching Ratios: Canonical Competitive Nonstatistical Model," J. Zheng, E. Papajak, and D. G. Truhlar, Journal of the American Chemical Society **131**, 15754-15760 (2009). x.doi.org/10.1021/ja904405v

82. "Least-Action Tunneling Transmission Coefficient for Polyatomic Reactions," R. Meana-Pañeda, D. G. Truhlar, and A. Fernández-Ramos, Journal of Chemical Theory and Computation **6**, 6-17 (2010). dx.doi.org/10.1021/ct900420e

83. "Kinetics of Hydrogen-Transfer Isomerizations of Butoxyl Radicals," J. Zheng and D. G. Truhlar, Physical Chemistry Chemical Physics **12**, 7782-7793 (2010). dx.doi.org/10.1039/b927504e.

84. "Free Energy Surfaces for Liquid-Phase Reactions and Their Use to Study the Border Between Concerted and Nonconcerted α,β-Elimination Reactions of Esters and Thioesters," Y. Kim, J. R. Mohrig, and D. G. Truhlar, Journal of the American Chemical Society **131**, 11071-11082 (2010). dx.doi.org/10.1021/ja101104q

85. "Computational Thermochemistry: Scale Factor Databases and Scale Factors for Vibrational Frequencies Obtained from Electronic Model Chemistries," I. M. Alecu, J. Zheng, Y. Zhao, and D. G. Truhlar, Journal of Chemical Theory and Computation **6**, 2872-2887 (2010). dx.doi.org/10.1021/ct100326h.

86. "Direct Dynamics Implementation of the Least-Action Tunneling Transmission Coefficient. Application to the $CH_4/CD_3H/CD_4+CF_3$ Abstraction Reactions," R. Meana-Pañeda, D. G. Truhlar, and A. Fernández-Ramos, Journal of Chemical Theory and Computation **6**, 3015-3025 (2010). dx.doi.org/10.1021/ct100285a

87. "Free Energy Surfaces for Liquid-Phase Reactions and Their Use to Study the Border Between Concerted and Nonconcerted α,β-Elimination Reactions of Esters and Thioesters," Y. Kim, J. R. Mohrig, and D. G. Truhlar, Journal of the American Chemical Society **132**, 11071-11082 (2010). doi.org/10.1021/ja101104q

88. "High-Level Direct-Dynamics Variational Transition State Theory Calculations Including Multidimensional Tunneling of the Thermal Rate Constants, Branching Ratios, and Kinetic Isotope Effects of the Hydrogen Abstraction Reactions from Methanol by Atomic Hydrogen," R. Meana-Pañeda, D. G. Truhlar and A. Fernández-Ramos, Journal of Chemical Physics **134**, 094302/1-14 (2011). dx.doi.org/10.1063/1.3555763

89. "Practical Methods for Including Torsional Anharmonicity in Thermochemical Calculations of Complex Molecules: The Internal-Coordinate Multi-Structural Approximation," J. Zheng, T. Yu, E. Papajak, I, M. Alecu, S. L. Mielke, and D. G. Truhlar, Physical Chemistry Chemical Physics **13**, 10885-10907 (2011). dx.doi.org/10.1039/C0CP02644A

90. "Multi-Structural Variational Transition State Theory. Kinetics of the 1,4-Hydrogen Shift Isomerization of the Pentyl Radical with Torsional Anharmonicity," T. Yu, J. Zheng, and D. G. Truhlar, Chemical Science **2**, 2199-2213 (2011). dx.doi.org/10.1039/ C1SC00225B

91. "Computational Study of the Reactions of Methanol with the Hydroperoxyl and Methyl Radicals. 2. Accurate Thermal Rate Constants," I. M. Alecu and D. G. Truhlar, Journal of Physical Chemistry A **115**, 14599-14611 (2011). dx.doi.org/10.1021/jp209029p

92. "Multipath Variational Transition State Theory. Rate Constant of the 1,4-Hydrogen Shift Isomerization of the 2-Cyclohexylethyl Radical," T. Yu, J. Zheng, and D. G. Truhlar, Journal of Physical Chemistry A **116**, 297-308 (2012). dx.doi.org/10.1021/jp209146b

93. "Kinetics of the Hydrogen Abstraction from Carbon-3 of 1-Butanol by Hydroperoxyl Radical: Multi-Structural Variational Transition State Calculations of a Reaction with 262 Conformations of the Transition State," P. Seal, E. Papajak, and D. G. Truhlar, Journal of Physical Chemistry Letters **3**, 264-271 (2012). dx.doi.org/10.1021/jz201546e

94. "Multi-Structural Variational Transition State Theory: Kinetics of the 1,5-Hydrogen Shift Isomerization of 1-Butoxyl Radical Including All Structures and Torsional Anharmonicity," X. Xu, E. Papajak, J. Zheng, and D. G. Truhlar, Physical Chemistry Chemical Physics **14**, 4204-4216 (2012). dx.doi.org/10.1039/c2cp23692c.

95. "Multi-path Variational Transition State Theory for Chemical Reaction Rates of Complex Polyatomic Species: Ethanol + OH Reactions," J. Zheng and D. G. Truhlar, Faraday Discussions **157**, 59-88 (2012). dx.doi.org/10.1039/C2FD20012K.

96. "MSTor: A program for calculating partition functions, free energies, enthalpies, entropies, and heat capacities of complex molecules including torsional anharmonicity," J. Zheng, S. L Mielke, K. L Clarkson, and D. G. Truhlar, Computer Physics Communications **183**, 1803-1812 (2012). dx.doi.org/10.1016/j.cpc.2012.03.007

97. "A Product Branching Ratio Controlled by Vibrational Adiabaticity and Variational Effects: Kinetics of the H + *trans*-N_2H_2 Reactions," J. Zheng, R. J. Rocha, M. Pelegrini, L. F. A. Ferrão, E. F. V. Carvalho, O. Roberto-Neto, F. B. C. Machado, and D. G. Truhlar, Journal of Chemical Physics **136**, 184310/1-10 (2012). dx.doi.org/10.1063/1.4707734

98. "Role of Conformational Structures and Torsional Anharmonicity in Controlling Chemical Reaction Rates and Relative Yields: Butanal + HO2 Reactions," J. Zheng, P. Seal, and D. G. Truhlar, Chemical Science **4**, 200-212 (2013). dx.doi.org/10.1039/c2sc21090h

99. "Multi-Structural Variational Transition State Theory: Kinetics of the Hydrogen Abstraction from Carbon-2 of 2-Methyl-1-propanol by Hydroperoxyl Radical Including All Structures and Torsional Anharmonicity," X. Xu, T. Yu, E. Papajak, and D. G. Truhlar, Journal of Physical Chemistry A **116**, 10480-10487 (2012). dx.doi.org/10.1021/jp307504p

100. "Biofuel Combustion. Energetics and Kinetics of Hydrogen Abstraction from Carbon-1 in n-Butanol by the Hydroperoxyl Radical Calculated by Coupled Cluster and Density Functional Theories and Multi-Structural Variational Transition State Theory with Multidimensional Tunneling," I. M. Alecu, J. Zheng, E. Papajak, T. Yu, and D. G. Truhlar, Journal of Physical Chemistry A **116**, 12206-12213 (2012). dx.doi.org/10.1021/jp308460y

101. "Kinetics of the Hydrogen Atom Abstraction Reactions from 1-Butanol by Hydroxyl Radical: Theory Matches Experiment and Then Some," P. Seal, G. Oyedepo, and D. G. Truhlar, Journal of Physical Chemistry A **117**, 275-282 (2013). dx.doi.org/10.1021/jp310910f

102. "Quantum Thermochemistry: Multi-Structural Method with Torsional Anharmonicity Based on a Coupled Torsional Potential," J. Zheng and D. G. Truhlar, Journal of Chemical Theory and Computation **9**, 1356-1367 (2013). dx.doi.org/10.1021/ct3010722

103. "Including Torsional Anharmonicity in Canonical and Microcanonical Reaction Path Calculations," J. Zheng and D. G. Truhlar, Journal of Chemical Theory and Computation **9**, 2875-2881(2013). dx.doi.org/10.1021/ct400231q

104. "*MSTor* version 2013*: A new version of the computer code for the multistructural torsional anharmonicity with a coupled torsional potential," J. Zheng, R. Meana-Pañeda, and D. G. Truhlar, Computer Physics Communications **184**, 2032-2033 (2013). [new version announcement] dx.doi.org/10.1016/j.cpc.2013.03.011

105. "Multi-Path Variational Transition State Theory for Chiral Molecules: The Site-Dependent Kinetics for Abstraction of Hydrogen from Hydroperoxyl Radical, Analysis of Hydrogen Bonding in the Transition State, and Dramatic Temperature Dependence of the Activation Energy," J. L. Bao, R. Meana-Pañeda, and D. G. Truhlar, Chemical Science **6**, 5866-5881 (2015). doi.org/10.1039/C5SC01848J

106. "Kinetics of Hydrogen Radical Reactions with Toluene by Chemical Activation Theory Employing System-Specific Quantum RRK Theory Calibrated by Variational Transition State Theory," J. L. Bao, J. Zheng, and D. G. Truhlar, Journal of the American Chemical Society **138**, 2690-2704 (2016). doi.org/10.1021/jacs.5b11938

107. "Silane-Initiated Nucleation in Chemically Active Plasmas: Validation of Density Functionals, Mechanisms, and Pressure-Dependent Variational Transition State Calculations," J. L. Bao and D. G. Truhlar, Physical Chemistry Chemical Physics **18**, 10097-10108 (2016). doi.org/10.1039/c6cp00816j

108. "Predicting Pressure-Dependent Unimolecular Rate Constants Using Variational Transition State Theory with Multidimensional Tunneling Combined with System-Specific Quantum RRK Theory: A Definitive Test for Fluoroform Dissociation," J. L. Bao, X. Zhang, and D. G. Truhlar, Physical Chemistry Chemical Physics **18**, 16659-16670 (2016). doi.org/10.1039/C6CP02765B

109. "Atmospheric Chemistry of Criegee Intermediates. Unimolecular Reactions and Reactions with Water," B. Long, J. L. Bao, and D. G. Truhlar, Journal of the American Chemical Society **138**, 14409-14422 (2016). doi.org/10.1021/jacs.6b08655

110. "Association of CF_2 and Dissociation of C_2F_4 by Variational Transition State Theory and System-Specific Quantum RRK Theory," J. L. Bao, X. Zhang, and D. G. Truhlar, Proceedings of the National Academy of Sciences U.S.A. **113**, 13606-13611 (2016). doi.org/10.1073/pnas.1616208113

111. "Computational Thermochemistry: Automated Generation of Scale Factors for Vibrational Frequencies Calculated by Electronic Structure Model Chemistries," H. S. Yu, L. J. Fiedler, I. M. Alecu, and D. G. Truhlar, Computer Physics Communications **210**, 132-138 (2017). doi.org/10.1016/j.cpc.2016.09.004

112. "Reaction of SO_2 with OH in the Atmosphere," B. Long, J. L. Bao, and D. G. Truhlar, Physical Chemistry Chemical Physics **19**, 8091-8100 (2017). doi.org/10.1039/c7cp00497d

113. "Dual-Level Method for Estimating Multi-Structural Partition Functions with Torsional Anharmonicity," J. L. Bao, L. Xing, and D. G Truhlar, Journal of Chemical Theory and Computation **13**, 2511-2522 (2017). doi.org/10.1021/acs.jctc.7b00232

114. "Variational Transition State Theory: Theoretical Framework and Recent Developments," J. L. Bao and D. G. Truhlar, Chemical Society Reviews **46**, 7548-7596 (2017). doi.org/10.1039/C7CS00602K

115. "Kinetics of the Methanol Reaction with OH at Interstellar, Atmospheric, and Combustion Temperature," L. G. Gao, J. Zheng, A. Fernández-Ramos, D. G. Truhlar, and X. Xu, Journal of the American Chemical Society **140**, 2906-2918 (2018). doi.org/10.1021/jacs.7b12773

116. "Unimolecular Reaction of Acetone Oxide and its Reaction with Water in the Atmosphere," B. Long, J. L. Bao, and D. G. Truhlar, Proceedings of the National Academy of Sciences U.S.A.**115**, 6135-6140 (2018). doi.org/10.1073/pnas.1804453115

117. "Hydrogen Shift Isomerizations in the Kinetics of the Second Oxidation Mechanism of Alkane Combustion. Reactions of the Hydroperoxypentylperoxy OOQOOH Radical," L. Xing, J. L. Bao, X. Wang, F. Zhang, and D. G. Truhlar, Combustion and Flame Combustion and Flame **197**, 88–101 (2018). doi.org/10.1016/j.combustflame.2018.07.013

118. "Relative Rates of Hydrogen Shift Isomerizations Depend Strongly on Multiple-Structure Anharmonicity," L. Xing, J. L. Bao, Z. Wang, X. Wang, and D. G. Truhlar, Journal of the American Chemical Society **140**, 17556-17570 (2018). doi.org/10.1021/jacs.8b09381

119. "Effect of Energy Dependence of the Density of States on Pressure-Dependent Rate Constants," J. L. Bao and D. G. Truhlar, Physical Chemistry Chemical Physics **20**, 30475-30479 (2018). doi.org/10.1039/C8CP05915B

120. "Kinetics of the Strongly Correlated CH_3O + O_2 Reaction: The Importance of Quadruple Excitations in Atmospheric and Combustion Chemistry," B. Long, J. L. Bao, and D. G. Truhlar, Journal of the American Chemical Society **141**, 611-617 (2019). doi.org/10.1021/jacs.8b11766

121. "Kinetics of the Toluene Reaction with OH Radical," R. M. Zhang, X. Xu, and D. G. Truhlar, Research **2019**, 5373785/1-19 (2019). (*Research* is the first journal in the Science Partner Journal (SPJ) program and is the official journal of the China Association for Science and Technology (CAST).) (Open access) doi.org/10.34133/2019/5373785

122. "Rapid Unimolecular Reaction of Stabilized Criegee Intermediates and Implications for Atmospheric Chemistry," B. Long*, J. L. Bao, and D. G. Truhlar, Nature Communications **10**, 2003/1-8 (2019). doi.org/10.1038/s41467-019-09948-7

123. "Quantum Effects on H_2 Diffusion in Zeolite RHO: Inverse Kinetic Isotope Effect for Sieving," L. G. Gao, R. M. Zhang, X. Xu, and D. G. Truhlar, Journal of the American Chemical Society **134,** 13635-13642 (2019). doi.org/10.1021/jacs.9b06506

124. "Multistructural Anharmonicity Controls the Radical Generation Process in Biofuel Combustion, L. Xing, Z. Wang, and D. G. Truhlar, Journal of the American Chemical Society **141**, 18531-18543 (2019). doi.org/10.1021/jacs.9b09194

125. "Association of Cl with C_2H_2 by Unified Variable-Reaction-Coordinate and Reaction-Path Variational Transition State Theory," L. Zhang, D. G. Truhlar, and S. Sun, Proceedings of the National Academy of Sciences U.S.A. **117**, 5610-5616 (2020). doi.org/10.1073/pnas.1920018117

126. "Water Catalysis of the Reaction of Methanol with OH Radical in the Atmosphere is Negligible," J. Wu, L. G. Gao, Z. Varga, X. Xu, W. Ren, D. G Truhlar, Angewandte Chemie International Edition **59**, 10826-10830 (2020). doi.org/10.1002/anie.202001065

127. "Large Anharmonic Effects on Tunneling and Kinetics: The Reaction of Propane with Muonium." L. Gao, D. G. Fleming, D. G. Truhlar, and X. Xu, Journal of Physical Chemistry Letters **17**, 4154-4159 (2021). doi.org/10.1021/acs.jpclett.1c01229

128. "Strong dependence on multistructural anharmonicity of the relative rates of intramolecular H-migration in alkylperoxyl and methylcyclohexylperoxyl radicals," L. Xing, L. Lian, and D. G. Truhlar, Combustion and Flame **231**, 11503/1-12 (2021). doi.org/10.1016/j.combustflame.2021.111503

129. Atmospheric Kinetics. Bimolecular Reactions of Carbonyl Oxide by a Triple-Level Strategy," B. Long, Y. Wang, Y. Xia, X. He, J. L. Bao, and D. G. Truhlar, Journal of the American Chemical Society **143**, 8402-8413 (2021). doi.org/10.1021/jacs.1c02029

22. ERRATA

All known errata to the GTST book chapter (Ref. 5 in Sect. 21) have been assembled into a PDF file. This file is available at the *Polyrate* Web site.

23. ACKNOWLEDGMENTS

24. INDEX