

GAMESSPLUSRATE Manual

Program version: 9.3/P9.3-G4.2
Version date: December 30, 2004
Date of most recent manual update: June 14, 2004
Copyright 2004

Yao-Yuan Chuang, Jose C. Corchado, Jingzhi Pu, and Donald G. Truhlar
Department of Chemistry and Supercomputer Institute
University of Minnesota, Minneapolis, MN 55455-0431

GAMESSPLUSRATE-version 9.3 provides an interface between two other programs:

POLYRATE-version 9.3, by J. C. Corchado, Y.-Y. Chuang, P. L. Fast,
J. Villa, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels,
V. S. Melissas, B. J. Lynch, I. Rossi, E. L. Coitino, A. Fernandez-Ramos,
J. Pu, T. V. Albu, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar

and

GAMESSPLUS-version 4.2, by J. Pu, J. D. Thompson, J. D. Xidos, J. Li,
T. Zhu, G. D. Hawkins, Y.-Y. Chuang, P. L. Fast, D. A. Liotard, D. Rinaldi,
J. Gao, C. J. Cramer, and D. G. Truhlar

Abstract: GAMESSPLUSRATE is a set of FORTRAN subroutines and Unix scripts for interfacing the POLYRATE and GAMESSPLUS computer programs for the purpose of carrying out direct dynamics calculations of gas-phase and liquid-phase chemical reaction rates of polyatomic species (and also atoms and diatoms as special cases) using the electronic structure methods available in GAMESSPLUS to calculate the potential energy surface and POLYRATE for the dynamics. The interface is based on the POLYRATE hooks protocol. The dynamical methods used are variational or conventional transition state theory and multidimensional semiclassical approximations by any of the POLYRATE methods for tunneling and nonclassical reflection. Gas-phase rate constants may be calculated by any of the methods available in the POLYRATE-version 9.3 program for canonical or microcanonical ensembles or for specific vibrational states of selected modes with translational, rotational, and other vibrational modes treated thermally. Liquid-phase rate constants may be calculated by the separable equilibrium solvation (SES), equilibrium solvation path (ESP) approximation, or nonequilibrium solvation (NES) approximation. Bimolecular and unimolecular reactions are included. Both single-level and dual-level methods for dynamics calculation are available. In single-level mode, optimized geometries, potential energies, gradients, and Hessians can be calculated by any of the size-consistent ab initio or semiempirical methods in the GAMESSPLUS package. In dual-level mode, the lower-level data is calculated by GAMESSPLUS, and the higher-level data is read from an external file. In addition, the liquid-solution reaction rate constants can be calculated using separable equilibrium solvation (SES), equilibrium solvation path (ESP), or nonequilibrium solvation (NES) approximations. GAMESSPLUSRATE also includes a linear coefficient method for mixing two electronic structure theories, and this capability may be used, for example, to

carry out calculations by the linear combination of ab initio Hartree-Fock and semiempirical Molecular Orbital Theory (HF||MO) approximation. The GAMESSPLUSRATE interface of GAMESSPLUS with POLYRATE is similar to the GAUSSRATE interface of GAUSSIAN with POLYRATE; however only GAMESSPLUSRATE contains the linear coefficient capability.

CONTENTS

	Page
TITLE PAGE	i
CONTENTS	iii
USER AGREEMENT	v
1. INTRODUCTION	1-1
2. INSTALLING GAMESSPLUSRATE	2-1
2.A. OBTAINING THE CODES	2-2
2.B. INSTALLATION	2-2
2.C. DESCRIPTION OF GAMESSPLUSRATE SUBPROGRAMS	2-5
3. REFERENCES FOR THE GAMESSPLUSRATE PROGRAM	3-1
3.A. Recommended reference	3-1
3.B. Background references	3-1
4. INPUT/OUTPUT FILE USAGE	4-1
5. RESTARTING A CALCULATION	5-1
6. DESCRIPTION OF INPUT DATA	6-1
6.A. UNIT fu5	6-1
6.B. esp.fu71 THROUGH esp.fu75 FILES AND AVOIDING REORIENTATION.	6-5
6.C. esp.fu70 FILE	6-6
6.C.1. SECTION GRGENERAL	6-7
6.C.2. SECTION GRDUAL	6-7
6.C.3. SECTION GRSOLV	6-7
6.C.4. SECTION GRBASIS	6-7
6.C.5. SECTION GRCOMMON	6-8
6.C.6. SECTIONS GRREACT1, GRREACT2, GRPROD1, GRPROD2, GRWELLR, GRWELLP, GRSTART AND GRPATH	6-9
7. TESTING GAMESSPLUSRATE	7-1
7.A. TESTS PERFORMED	7-1
7.B. GAS-PHASE REACTIONS	7-1
7.B.1. TEST RUN ch5tr1	7-2
7.B.2. TEST RUN ch5tr2	7-3
7.B.3. TEST RUN ch4ohtr1	7-4
7.B.4. TEST RUN ch4ohtr2	7-5
7.B.5. TEST RUN ch4ohtr3	7-6
7.B.6. TEST RUN ch4ohtr4	7-7
7.B.7. TEST RUN ch4ohtr5	7-8
7.B.8. TEST RUN nh3ohtr1	7-9
7.B.9. TEST RUN ch3ohhtr1	7-10
7.B.10. TEST RUN ch3ohhtr2	7-11
7.B.11. TEST RUN ch3cl2tr1	7-12
7.C. LIQUID-PHASE REACTIONS USING SES	7-13
7.C.1. TEST RUN ch3ohhtr1	7-13
7.C.2. TEST RUN ch3cl2tr1	7-14
7.C.3. TEST RUN ch3clnh3tr1	7-15
7.C.4. TEST RUN ch3clnh3tr2	7-16
7.C.5. TEST RUN 2hcoohtr1	7-17
7.D. LIQUID-PHASE REACTIONS USING ESP	7-18
7.D.1. TEST RUN ch3ohhtr1	7-18
7.D.2. TEST RUN ch3cl2tr1	7-19
7.D.3. TEST RUN ch3clnh3tr1	7-20
7.D.4. TEST RUN ch3clnh3tr2	7-21

7.D.5. TEST RUN ch3c2h4tr1	7-22
7.D.6. TEST RUN ch3chotr1	7-23
7.E. LIQUID-PHASE REACTIONS USING NES	7-24
7.E.1. TEST RUN ch3ohhtr1	7-24
7.E.2. TEST RUN ch3cl2tr1	7-25
8. COMPUTERS, OPERATING SYSTEMS FOR WHICH GAMESPLUSRATE HAS BEEN TESTED	8-1
9. TEST RUN TIMINGS	9-1
10. VERSION INFORMATION	10-1
10.A. GAMESPLUSRATE VERSION NUMBERS	10-1
10.B. GAMESPLUSRATE REVISION HISTORY	10-1
11. ACKNOWLEDGMENTS	11-1

USER AGREEMENT

GAMESSPLUSRATE is a licensed program, and use of this program implies acceptance of the terms of the license, which are repeated here for convenience:

1. No user or site will redistribute the source code or executable code to a third party in original or modified form without written permission of the principal investigator (Donald G. Truhlar). A license does not entitle the licensee to relicense the code or distribute it in original or modified form to parties not covered by the license. The licensee has no ownership rights in the GAMESSPLUSRATE software or in any copyrights for the GAMESSPLUSRATE software or documentation through this license. A user license covers the work of a single research group and the code may be shared and disseminated within a group without requiring permission. Site-license inquiries should be directed to the principal investigator (Donald G. Truhlar).
2. Publications resulting from using this package or the POLYRATE or GAMESSPLUS subsystems used by this package will cite the corresponding program. The recommended references are given in the documentation (see Section 3 for the recommended reference for GAMESSPLUSRATE).
3. No guarantee is made that this program is bug-free or suitable for specific applications, and no liability is accepted for any limitations in the mathematical methods and algorithms used within the program.
4. No consulting or maintenance services are guaranteed or implied.
5. The POLYRATE 9.3 and GAMESSPLUS 4.2 codes required to use GAMESSPLUSRATE are covered by separate licenses.

1. INTRODUCTION

GAMESSPLUSRATE is a program for the analysis of reactants, products, and transition states of chemical reactions and for direct dynamics calculations of variational transition state theory (VTST) rate constants and multi-dimensional semiclassical tunneling probabilities using the electronic structure methods of the GAMESSPLUS-version 4.1 package to represent the potential energy of interaction and its first and second derivatives wherever they are needed. This version, 9.3/P9.3-G4.2, interfaces POLYRATE-version 9.3, which is a program for dynamical rate calculations, with the electronic structure program GAMESSPLUS-version 4.1.

GAMESSPLUSRATE assumes that the user has both POLYRATE 9.3 and GAMESSPLUS-version 4.1 on the same computer working properly before attempting to run GAMESSPLUS. No part of POLYRATE 9.3 or GAMESSPLUS is contained in GAMESSPLUSRATE or needs to be modified for running this program.

This GAMESSPLUSRATE manual covers only material that is not covered in the POLYRATE and GAMESSPLUS manuals. The user of GAMESSPLUSRATE should read those manuals first.

The interface between POLYRATE and GAMESSPLUS is made by a series of FORTRAN subroutines included in the files 'hooks_gr.f' and 'hooks_mdep.f', which are specialized versions of the routines in the 'hooks.f' file in POLYRATE 9.1 and by a C shell script for running GAMESSPLUS. This C shell script, named 'shuttle', is called from inside GAMESSPLUSRATE, and it has to be able to start running GAMESSPLUS using an input file named as the first argument given to 'shuttle'. It produces an output file named as the second argument given to 'shuttle'. For the user's convenience, a well documented script is provided, and in most cases the user will only need to change a few variables. The variable VERNO is used to specify the version of GAMESSPLUS program, the EXE variable is used to indicate the location of the executable, and the SCR variable is used to indicate the scratch directory.

GAMESSPLUSRATE uses a subroutine that is usually included in libraries provided with FORTRAN compilers. This subroutine is called 'SYSTEM' on IBM computers.

GAMESSPLUS is an enhanced GAMESS package that contains all the capabilities of GAMESS plus (i) solvation module for SM5.42R and SM5.42 calculations in liquid solutions (ii) additional capabilities for using specific reaction parameters in semiempirical molecular orbital theory (iii) redistributed Lowdin population analysis (RLPA) (iv) the CM2 and CM3 charge models and (iv) combined quantum mechanics and molecular mechanics (QM/MM) with the generalized hybrid orbital (GHO) boundary treatment.

The GAMESSPLUSRATE interface of GAMESSPLUS with POLYRATE is similar to the GAUSSRATE interface of GAUSSIAN with POLYRATE. However, GAMESSPLUSRATE includes one capability not currently in GAUSSRATE, namely the linear coefficient capability to take linear combinations of two energies or gradients calculated by different methods and the ability to use such a linear combination as a potential energy surface or force field.

There is no separate GAMESSRATE interface. GAMESSPLUSRATE is intended to support gas-phase as well as liquid-phase calculations.

2. INSTALLING GAMESSPLUSRATE

2.A. OBTAINING THE CODES

The user needs to obtain four items of code:

POLYRATE: The POLYRATE code is a licensed code available from the University of Minnesota. Currently the license fee is waived for all users. For further information see:

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

GAMESSPLUSRATE: The GAMESSPLUSRATE code is a licensed code, also available from the University of Minnesota. Currently the license fee is waived for all users. For further information see

<http://comp.chem.umn.edu/gamesplusrate>

GAMESSPLUS: The GAMESSPLUS code is a licensed code available from the University of Minnesota. Currently the license fee is waived for all users. For further information see:

<http://comp.chem.umn.edu/gamesplus>

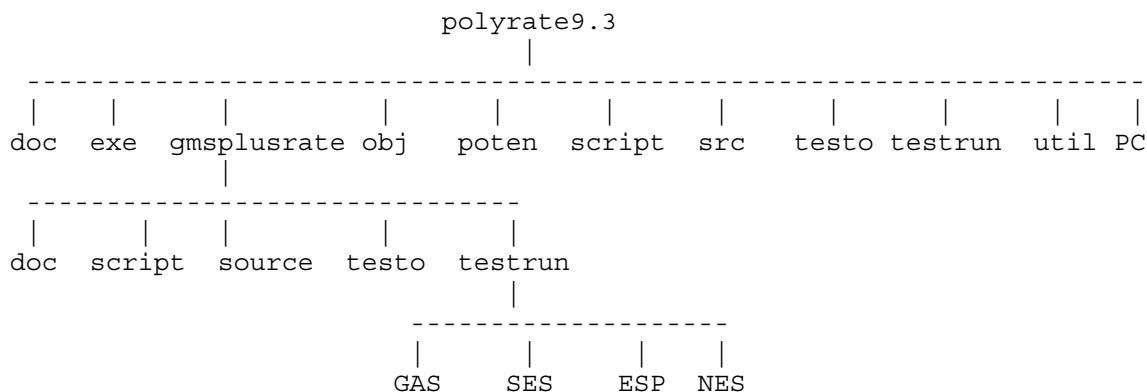
GAMESS: The GAMESS code is a licensed package available from Iowa State University. For further information see:

<http://www.msg.ameslab.gov/GAMESS/GAMESS.html/>

The prospective user of GAMESSPLUSRATE must obtain and install POLYRATE, GAMESS, and GAMESSPLUS before proceeding with GAMESSPLUSRATE.

2.B. INSTALLATION

The user should first install GAMESS, GAMESSPLUS, and POLYRATE. GAMESSPLUSRATE is distributed in tar format. The tar file needs to be untarred in the POLYRATE 9.3 directory. The command is 'tar -xvf gmsplusrate9.3.tar'. This command will create a directory called 'gmsplusrate'. In this directory are located the C shell sample script 'shuttle' and the directories 'script', 'source', 'doc', 'testo', and 'testrun'. A few variables of the C shell script 'shuttle' should be changed as described in Sect. 1 before compiling GAMESSPLUSRATE, in order to place a copy of it in the 'exe' POLYRATE subdirectory to make it accessible for all the GAMESSPLUSRATE calculations. If these instructions are followed correctly, the directory structure should be the following:



The files will be distributed among the directories as follows:

```

gmsplusrate:      shuttle

doc:              gmsplusrate.txt

script:           gmsplusrate.sgi, gmsplusrate.ibm, gmsplusrate.altix,
                  gmsplusrate.nf, gmsplusrate.sun

source:           hooks_gr.f, hooks_gr.wks, head_gr.f,
                  gmsplusrate.inc

testo:            ch5tr1.fu15, ch5tr2.fu15, oh3tr1.fu15, oh3tr2.fu15,
                  oh3tr3.fu15, ch4ohtr3.fu15, ch4ohtr4.fu15,
                  ch4ohtr5.fu15, nh3ohtr1.fu15

testrun:          check_test.jc, check_all.jc, run_all.jc and four
                  subdirectories

testrun/GAS/ch5:  ch5tr1.70, ch5tr1.71, ch5tr1.72, ch5tr1.73,
                  ch5tr1.74, ch5tr1.75, ch5tr1.dat, ch5tr1.jc
                  ch5tr2.51, ch5tr2.70, ch5tr2.71, ch5tr2.72,
                  ch5tr2.73, ch5tr2.74, ch5tr2.75, ch5tr2.dat,
                  ch5tr2.jc

testrun/GAS/ch4oh: ch4ohtr1.70, ch4ohtr1.71, ch4ohtr1.jc, ch4ohtr1.dat,
                  ch4ohtr2.70, ch4ohtr2.72, ch4ohtr2.73, ch4ohtr2.74,
                  ch4ohtr2.jc, ch4ohtr2.dat, ch4ohtr3.70, ch4ohtr3.75,
                  ch4ohtr3.dat, ch4ohtr3.jc, ch4ohtr4.70, ch4ohtr4.dat,
                  ch4ohtr4.jc, ch4ohtr5.50, ch4ohtr5.70, ch4ohtr5.dat,
                  ch4ohtr5.jc

testrun/GAS/nh3oh: nh3ohtr1.70, nh3ohtr1.71, nh3ohtr1.72, nh3ohtr1.73,
                  nh3ohtr1.74, nh3ohtr1.75, nh3ohtr1.77, nh3ohtr1.78,
                  nh3ohtr1.jc, nh3ohtr1.dat

testrun/GAS/ch3ohh: ch3ohhtr1.70, ch3ohhtr1.dat, ch3ohhtr1.jc, ,
                  ch3ohhtr2.70, ch3ohhtr2.dat, ch3ohhtr2.jc, SRP

```


testrun/GAS/ch3cl2:ch3cl2tr1.70, ch3cl2tr1.71, ch3cl2tr1.72,
ch3cl2tr1.73, ch3cl2tr1.74, ch3cl2tr1.75,
ch3cl2tr1.dat, ch3cl2tr1.jc

testrun/SES/ch3ohh:ch3ohhtr1.70, cheohhtr1.jc, ch3ohhtr1.dat, SRP

testrun/SES/ch3cl2:ch3cl2tr1.70, ch3cl2tr1.71, ch3cl2tr1.72,
ch3cl2tr1.73, ch3cl2tr1.74, ch3cl2tr1.75,
ch3cl2tr1.dat, ch3cl2tr1.jc

testrun/SES/ch3clnh3:ch3clnh3tr1.70, ch3clnh3tr1.71, ch3clnh3tr1.72,
ch3clnh3tr1.73, ch3clnh3tr1.74, ch3clnh3tr1.75,
ch3clnh3tr1.dat, ch3clnh3tr1.jc, ch3clnh3tr2.70,
ch3clnh3tr2.71, ch3clnh3tr2.72, ch3clnh3tr2.73,
ch3clnh3tr2.74, ch3clnh3tr2.75, ch3clnh3tr2.dat,
ch3clnh3tr2.jc

testrun/SES/2hcooh:2hcoohtr1.70, 2hcoohtr1.71, 2hcoohtr1.72,
2hcoohtr1.73, 2hcoohtr1.74, 2hcoohtr1.75,
2hcoohtr1.dat, 2hcoohtr1.jc

testrun/ESP/ch3ohh:ch3ohhtr1.70, ch3ohhtr1.dat, ch3ohhtr1.jc, TSRP

testrun/ESP/ch3cl2:ch3cl2tr1.70, ch3cl2tr1.71, ch3cl2tr1.72,
ch3cl2tr1.73, ch3cl2tr1.74, ch3cl2tr1.75,
ch3cl2tr1.dat, ch3cl2tr1.jc

testrun/ESP/ch3clnh3:ch3clnh3tr1.70, ch3clnh3tr1.71, ch3clnh3tr1.72,
ch3clnh3tr1.73, ch3clnh3tr1.74, ch3clnh3tr1.75,
ch3clnh3tr1.dat, ch3clnh3tr1.jc, ch3clnh3tr2.70,
ch3clnh3tr2.71, ch3clnh3tr2.72, ch3clnh3tr2.73,
ch3clnh3tr2.74, ch3clnh3tr2.75, ch3clnh3tr2.dat,
ch3clnh3tr2.jc

testrun/ESP/ch3c2h4:ch3c2h4tr1.70, ch3c2h4tr1.71, ch3c2h4tr1.72,
ch3c2h4tr1.73, ch3c2h4tr1.74, ch3c2h4tr1.75,
ch3c2h4tr1.dat, ch3c2h4tr1.jc

testrun/ESP/ch3cho:ch3chotr1.70, ch3chotr1.dat, ch3chotr1.jc

testrun/NES/ch3ohh:ch3ohhtr1.70, ch3ohhtr1.dat, ch3ohhtr1.jc, TSRP

testrun/NES/ch3cl2:ch3cl2tr1.70, ch3cl2tr1.71, ch3cl2tr1.72,
ch3cl2tr1.73, ch3cl2tr1.74, ch3cl2tr1.75,
ch3cl2tr1.dat, ch3cl2tr1.jc

The directory 'script' includes the machine-dependent compiling scripts 'gmsplusrate.sgi' (for Silicon Graphics Origin computers), 'gmsplusrate.ibm' (for IBM computers), 'gmsplusrate.altix' (for SGI Altix computers), 'gmsplusrate.nf' (for Netfinity Linux Cluster computers), and 'gmsplusrate.sun' (for Sun workstations). These scripts will create a 'wrk' temporary directory and the hooks_mdep.f code file by copying the machine-dependent file hooks_gr.wks to the hooks_mdep.f file, they will compile the POLYRATE 9.3 files substituting the 'hooks.f' file of standard POLYRATE by 'hooks_gr.f' and 'hooks_mdep.f', and they will create the executable file 'gmsplusrate.exe' in the 'exe' subdirectory in the POLYRATE 9.3 directory. When executing one of these scripts, one must supply the number of the param.inc file to be used in the GAMESSPLUSRATE calculation (see Sect. 15.A.2 of the POLYRATE 9.3 manual for a definition of the 'param.inc' file). For example, for compiling and linking GAMESSPLUSRATE on an IBM workstation using the param.inc file number 3, type

```
gmsplusrate.ibm 3
```

Note that GAMESSPLUSRATE does not use any of the utility routines (routines starting with Y) of the generic hooks package of POLYRATE.

The directory 'testrun' contains a test suite distributed among several subdirectories. Each subdirectory contains one or more tests based on the same reaction, as explained in Section 7 of this manual. The summary output files from these test runs are located in the 'testo' directory. The C shell scripts 'check_test.jc' allows the comparison between the poly.fu15 output files obtained after running all the test runs and the poly.fu15 files in the 'testo' directory. The "run_all.jc" script the whole test suite and the "check_all.jc" script checks the whole test suite.

The directory 'doc' contains the present ASCII manual for GAMESSPLUSRATE.

The directory 'source' contains the source code for GAMESSPLUSRATE.

2.C. DESCRIPTION OF GAMESSPLUSRATE SUBPROGRAMS

This is a brief description of GAMESSPLUSRATE subprograms. In most of the cases, a more detailed description is given in the comment cards at the beginning of the FORTRAN code for each subprogram.

EHOOK

Equivalent to the EHOOK subroutine in the POLYRATE hooks. For a given set of coordinates, EHOOK calculates the energy using GAMESSPLUS.

EFFBATH

Calculates the coupling constants between the solvent and solute coordinates.

ENATIN

Calculates the energy for atomic systems.

ESCENER

Carries out GAMESSPLUS energy calculations.

ESCFIRST

Carries out GAMESSPLUS energy and first derivative calculations.

ESCSEC

Carries out GAMESSPLUS Hessian calculations.

ESCSMX

Carries out GAMESSPLUS single-point liquid-solution free energy calculation.

GEOIN

Carries out GAMESSPLUS geometry optimizations.

GHOOK

Equivalent to the GHOOK subroutine in the POLYRATE hooks. For a given geometry, calculates the energy and first derivatives.

GINPENE

Creates the GAMESSPLUS input file for energy calculations.

GINPFRS

Creates the GAMESSPLUS input file for gradient calculations.

GINPSEC

Creates the GAMESSPLUS input file for Hessian calculations.

GINPSMX

Creates the GAMESSPLUS input file for single-point liquid-solution free energy calculations.

GOUTENG

Reads the archive file Test.FChk generated by GAMESSPLUS, and extracts the energy.

GOUTFIR

Reads the archive file Test.FChk generated by GAMESSPLUS, and extracts the energy and gradient.

GOUTOPT

Reads the archive file Test.FChk generated by GAMESSPLUS, and extracts the energy and geometry information.

GOUTSEC

Reads the archive file Test.FChk generated by GAMESSPLUS, and extracts energy, gradient, and Hessian.

GOUTSMX

Reads the archive file Test.FChk generated by GAMESSPLUS, and extracts energy in solution.

GREST

Reads and writes the information in the restart files of a GAMESSPLUSRATE restart calculation.

HEADR

Prints out the program header.

HHOOK

Equivalent to the HHOOK subroutine in the POLYRATE hooks. For a given geometry, calculates the Hessian matrix using whether GAMESSPLUS or POLYRATE methods.

OHOOK

Equivalent to the OHOOK subroutine in the POLYRATE hooks. Optimizes a given initial geometry, using either POLYRATE or GAMESSPLUS methods.

PREP

Prepares the calculation and, depending on the selected options, carries out certain geometry optimizations and Hessian calculations.

PREPJ

Initialize variables dependent on each stationary point.

RBASIS1

Reads the basis set information for the lower level.

RBASIS2

Reads the basis set information for the higher level.

RCODGR
Reads the input option for the GAMESSPLUS runs from file esp.fu70.

READLN1
Reads a line from a file and parses into words.

READ_GEOM
Reads geometry information from the GAMESSPLUS archive file Test.FChk.

READ_GEOM1
Reads geometry information from the GAMESSPLUS archive file Test.FChk for a optimization calculation.

READ_GRAD
Reads gradient and energy from the GAMESSPLUS archive file Test.FChk.

READ_SMX
Reads liquid-solution free energy from the GAMESSPLUS archive file Test.FChk.

READ_HESS
Reads Hessian from the GAMESSPLUS archive file Test.FChk.

REDINF
Reads the information about reactants, products, and saddle point calculated in the PREP subroutine.

RENERGR
Reads the input options for GAMESSPLUS energy calculations.

RFIRSTGR
Reads the input options for GAMESSPLUS gradient calculations.

RGDUAL
Reads the options in the GRADUAL section of the esp.fu70 input file.

RGENGR
Reads the options in the GENERAL section of the esp.fu70 input file.

RLEVEL
Reads the options in the LEVEL keyword of the esp.fu70 input file.

RGMETHOD
Reads the options in the METHOD keyword of the esp.fu70 input file.

RGRBASIS
Reads the options in the BASIS keyword of the esp.fu70 input file.

RGSOLV
Reads the options in the GRSOLV section of the esp.fu70 input file.

RINPGR
Reads the esp.fu70 GAMESSPLUSRATE input file.

RLINGR

Utility program for reading in a line of the input fu70 file and parsing out comments and extra spaces.

RSECGR

Reads the input options for GAMESSPLUS Hessian calculations.

RSTATGR

Reads the input options given in fu70 for the stationary points.

SECIN

Carries out GAMESSPLUS Hessian calculations for stationary points.

SETUP

Initializes some variables and calls the subroutines for reading in the fu70 input file.

STORE

Stores some information that could be useful in future calculations in order to avoid duplicate GAMESSPLUS calls.

3. REFERENCES FOR THE GAMESSPLUSRATE PROGRAM

3.A. Recommended reference

The recommended reference to be given when results based on the current version of the code are published:

GAMESSPLUSRATE-version 9.3/P9.3-G4.2 by Y.-Y. Chuang, J. C. Corchado, Jingzhi Pu, and D. G. Truhlar, University of Minnesota, Minneapolis, 2004, based on POLYRATE-version 9.3, by J. C. Corchado, Y.-Y. Chuang, P. L. Fast, J. Villa, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, V. S. Melissas, B. J. Lynch, I. Rossi, E. L. Coitino, A. Fernandez-Ramos, J. Pu, T. V. Albu, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar, University of Minnesota, Minneapolis, 2003 and on GAMESSPLUS-version 4.1 by J. Pu, J. D. Thompson, J. D. Xidos, J. Li, T. Zhu, G. D. Hawkins, Y.-Y. Chuang, P. L. Fast, D. A. Liotard, D. Rinaldi, J. Gao, C. J. Cramer, and D. G. Truhlar, University of Minnesota, Minneapolis, 2003. GAMESSPLUS is based on GAMESS: M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. J. Su, T. L. Windus, M. Dupuis, and J. A. Montgomery, *J. Comp. Chem.* 14, 1347 (1993).

Additional references are given in the documentation for POLYRATE and GAMESSPLUS.

3.B. Background references

1. "Tunneling in the Presence of a Bath: A Generalized Transition State Theory Approach," D. G. Truhlar, Y.-P. Liu, G. K. Schenter, and B. C. Garrett, *Journal of Physical Chemistry* 98, 8396-8405 (1994).
2. "The 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*, submitted for publication in *International Journal of Quantum Chemistry* 70, 887-896 (1998). (Sanibel issue)
3. "Direct dynamics for Free Radical Kinetics in Solution: Solvent Effect on the Rate Constant for the Reaction of Methanol with Atomic Hydrogen," Y.-Y. Chuang, M. L. Radhakrishnan, P. L. Fast, C. J. Cramer, and D. G. Truhlar, *Journal of Physical Chemistry A* 103, 4893-4909 (1999).
4. "Nonequilibrium Solvation Effects for a Polyatomic Reaction in Solution," Y.-Y. Chuang and D. G. Truhlar, *Journal of American Chemical Society* 121, 10157-10167 (1999).

4. INPUT/OUTPUT FILE USAGE

In addition to the POLYRATE 9.3 and GAMESSPLUS input files, GAMESSPLUSRATE uses several files of its own for input data, for storing restart information, and for output. All these files have the name esp.fu#, where # denotes an integer.

File	Usage
esp.fu70	General GAMESSPLUSRATE input data.
esp.fu71	GAMESSPLUS input file for geometry optimization of reactant 1, if such optimization is going to be carried out.
esp.fu72	GAMESSPLUS input file for geometry optimization of reactant 2, if the reaction studied is bimolecular and if such optimization is going to be carried out by means of the GAMESSPLUS program.
esp.fu73	GAMESSPLUS input file for geometry optimization of product 1, if such optimization is going to be carried out.
esp.fu74	GAMESSPLUS input file for geometry optimization of product 2, if the reaction studied has two products and if such optimization is going to be carried out by means of the GAMESSPLUS program.
esp.fu75	GAMESSPLUS input file for geometry optimization of the saddle point or constrained optimization of a reaction path starting geometry, if such optimization is desired.
esp.fu76	reserved
esp.fu77	GAMESSPLUS input file for geometry optimization of a well on the reactant side of the reaction path, if it exists and its optimization is going to be carried out by means of the methods in GAMESSPLUS.
esp.fu78	GAMESSPLUS input file for geometry optimization of a well on the product side of the reaction path, if it exists and its optimization is going to be carried out by means of the methods in GAMESSPLUS.
esp.fu81	GAMESSPLUS temporary input file, generated by GAMESSPLUSRATE.
esp.fu82	GAMESSPLUS temporary output file, generated by GAMESSPLUS. If the GAMESSPLUS calculation fails, it can be examined in order to find out why the failure occurred.
esp.fu83	Restart file to be read during a restart calculation (see Sect. 5).

- esp.fu84 Restart file to be written during a restart calculation (see Sect. 5).
- esp.fu85 GAMESSPLUSRATE uses the formatted archive file generated by GAMESSPLUS. The shuttle script calls this file Test.FChk, and GAMESSPLUSRATE does not delete it. GAMESSPLUSRATE reads this file from FORTRAN unit 85. This file will not be removed after finishing a calculation, so if the GAMESSPLUS calculation fails, it can be examined in order to find the cause of the failure.

Note that this file usage is consistent with Section 17 of the POLYRATE manual. Note also that esp stands for electronic structure package, and fu stands for FORTRAN unit. Finally note that esp.fu76 is reserved for future use, but it is not used in this version.

5. RESTARTING A CALCULATION

The GAMESSPLUSRATE restart option should not be confused with the POLYRATE restart options (see Section 7.C of the POLYRATE manual). The POLYRATE restart options can be used with GAMESSPLUSRATE, but GAMESSPLUSRATE also has its own, unique restart capability. The rest of this section concerns this GAMESSPLUSRATE restart capability.

The GAMESSPLUSRATE restart option is controlled by the GRRESTART keyword in the GRGENERAL section of the unit fu70 input file. The GAMESSPLUSRATE restart option is designed to take advantage of any previous calculations that may have been made on the same system, so that the number of GAMESSPLUS calls is minimized. As long as the information on the stationary points from previous calculations is still applicable, a restart calculation can be made (if there is a change in any stationary point input or option, the program will not read this information, and the results of trying a restart calculation are unpredictable). This restart option allows the user to change any parameter related to the reaction path calculations using as much information as possible from previous calculations.

Two restart files are used in this kind of restart calculation: esp.fu83, which stores the information to be read, as originated in a previous GAMESSPLUSRATE calculation, and esp.fu84, where the useful information read from esp.fu83 is stored along with the information calculated for the first time in the present GAMESSPLUSRATE run (esp.fu84 is the file to be used for future restart runs). The esp.fu84 file has to be created by GAMESSPLUSRATE, so if the program finds that this file exists, it will stop, to prevent overwriting a restart file.

One important practical point related to this restart option is that, if one finds that the range of the reaction path calculated in a GAMESSPLUSRATE run is not large enough to describe some effect (for example, tunneling), one can extend the calculated reaction path without repeating the calculation of the points included in the previous GAMESSPLUSRATE run. One should rename the esp.fu83 file provided by the first run as esp.fu84 and then run again with GRRESTART on. In a similar way, if one finds problems in the SCF convergence at some points along the reaction path, one can repeat the calculation including some of the options for improving SCF convergence available in GAMESSPLUS without recalculating the well converged points. Another possibility is computing one side of the reaction path with one computer (or one processor of a multiprocessor computer) and the other side with a different computer (or different processor); a new restart file combining the information from both calculations can be constructed by just appending one restart file to the end of the other restart file, allowing a calculation using the whole reaction path with no duplication of effort, at least in principle; in practice, due to the different precision of two different machines and round-off errors, some recalculation might be needed. One should be aware, though, that if a calculation is made using a step-size that is too large, giving a non-converged reaction path, a restart calculation using a different step-size would probably be unsuccessful, since the points along the two reaction paths would be different.

There is only one parameter related to the path that should not be changed between an original calculation and a restart calculation, namely the value of the POLYRATE input variable SCALEMASS, which determines the mass-scaled coordinates. Changing this parameter will cause a miscalculation of the frequencies. It is important to note that its default value is 1 amu, and this is the value assumed if no reaction path calculation is desired. Therefore, if a restart calculation including the reaction path is based on a restart file written by a TST-only calculation, SCALEMASS must be set to 1 amu in order to get the right frequencies. In general, we recommend always using a value of 1.0 amu for SCALEMASS in GAMESSPLUSRATE.

A GAMESSPLUSRATE restart run requires that all the information about all the stationary points be present on file esp.fu83, and this data cannot be updated in this kind of restart run. However, the use of the STATUS keyword in the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections of the POLYRATE fu5 input file can alleviate this inconvenience. Thus, the user can calculate reactants and products separately and include the information obtained in the fu5 input file. See the ch4oh testruns and Sections 7.C and 11.A of the POLYRATE manual.

6. DESCRIPTION OF INPUT DATA

We will assume that the user knows how to use POLYRATE 9.3 and GAMESSPLUS and is familiar with all their keywords. Therefore, when talking about the common input files, we will only discuss the differences between running POLYRATE 9.3 and running GAMESSPLUSRATE 9.3.

6.A. UNIT fu5

A GAMESSPLUSRATE calculation requires some input from input units for POLYRATE. At the very least, an fu5 input file is required for indicating several options such as electronic degeneracies, temperatures for the rate constant calculations, etc. Therefore, an fu5 file must be written, exactly as if one were using an analytical potential energy surface. If the calculation requires some additional input files, the user must provide them, following the instructions in the POLYRATE 9.3 manual. It is important to clarify some points about preparing the input file for unit fu5:

RESTART

If a POLYRATE 9.3 restart run is selected (option READFU1 or READFU1&2 for the RESTART keyword in the GENERAL section), the program will not modify any characteristic of the calculated reaction path. We therefore advise the user to use only the restart option in GAMESSPLUSRATE, controlled by the GRRESTART option in GRGENERAL, until the reaction path calculation is completely converged and all the desired tunneling options are calculated. When no more electronic structure calculations are required, the POLYRATE restart is recommended, since it is faster than the GAMESSPLUSRATE restart.

ATOMS

The ordering in the ATOMS keyword list of the GENERAL section of the input file for unit fu5 must be consistent with the ordering in unit fu75, if this unit is used in the calculation.

NOSUPERMOL

Because of the way the interface is constructed, the NOSUPERMOL keyword is required (note that SUPERMOL is the default option). Therefore, only size-consistent methods should be used in GAMESSPLUSRATE

EZERO

Because of the way the interface is constructed, the user must choose the CALCULATE option for the EZERO keyword of the ENERGETICS section, which is the default option.

POTENTIAL

The HOOKS option is required.

OPTTS AND OPTMIN

In GAMESSPLUSRATE there are two ways to optimize the geometries: by means of GAMESSPLUS methods or by means of POLYRATE 9.3 methods. The choice between the two methods is done in the OPTIMIZATION section of the fu5 input file. Choosing the option BFGS or NR for the OPTMIN or OPTTS keywords in the OPTIMIZATION section, will cause GAMESSPLUSRATE to use the methods native to POLYRATE 9.3, using as starting geometries those in the fu5 input file and controlling the optimization procedure by means of the keywords selected in the OPTIMIZATION section of the fu5 input file. The optimization options for GAMESSPLUS will be ignored, and the input files fu71 through fu75 will not be required. The INITGEO keyword needs to take the option GEOM in the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections of the fu5 input file.

If the option OHOOK is chosen for both the OPTTS and OPTMIN keywords in the OPTIMIZATION section of the fu5 input file, GAMESSPLUSRATE will optimize the geometries of the stationary points using the GAMESSPLUS program according to the options selected in the fu70 input file and the fu71-fu78 input files. The initial geometries for the optimization will be those indicated in the fu71-fu78 input files and the options selected in the OPTIMIZATION section in the fu5 input file will be ignored. INITGEO should take the value HOOKS in the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections of the fu5 input file.

If the user wishes to optimize the stationary points, we recommend the use of GAMESSPLUS methods; in particular this is accomplished by using the OHOOK option for both the OPTMIN and the OPTTS keyword in the OPTIMIZATION section of the fu5 input file, using initial geometries given in the fu71 through fu75 input files, and setting INITGEO equal to HOOKS. A further explanation of these keywords will be provided below.

However, if one uses the linear combination method of the Hartree Fock and Molecular Orbital Theory (HF||MO), geometry optimization has to be done by using an optimization algorithm in the POLYRATE package, in particular, by the Newton-Raphson method, the Newton-Raphson method with BFGS updates, or the Eigenvector Following method.

SECOND

The Hessian matrices can be calculated by numerical differentiation of the gradients, by means of the methods in POLYRATE 9.3, or by the GAMESSPLUS methods, with the latter option strongly recommended. The choice is made by means of the GHOOK or HHOOK options in the HESSCAL keyword in the SECOND section of the fu5 input file. If HHOOK is selected, the remaining keywords of the SECOND section are ignored and the Hessian calculations will be controlled by the options selected in the fu70 input file. If GHOOK is selected, the Hessian calculation will be carried out following the options indicated in the SECOND section, ignoring the options for the Hessian calculation selected in the fu70 input file.

Although GAMESSPLUS can perform Hessian calculations using any of the electronic structure methods available in GAMESS or GAMESSPLUS, the option of performing the Hessian calculation using POLYRATE methods is implemented for completeness. However, the user is advised to not use POLYRATE for the Hessian calculation because it is less efficient.

GEOM

The ordering of the atoms in the GEOM keyword in sections REACT1, REACT2, PROD1, PROD2, and START must be consistent with the ordering in units fu71, fu72, fu73, fu74, and fu75, when they are used; and also the ordering in fu75 must be consistent with the one in ATOMS in the GENERAL section.

CONSTANT

If the user selects GAMESSPLUS for optimizing the geometries (this is accomplishing with the OHOOK option for the OPTMIN and OPTTS keywords in the OPTIMIZATION section of the POLYRATE fu5 input), the CONSTANT keyword in the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP and START sections of the fu5 input file is meaningless. The frozen parameters must be indicated in units fu71-fu78, according to the instructions explained in the Chapter 3 of the GAMESSPLUS manual. If a user selects POLYRATE 9.3 for optimizing the geometries, this keyword has the same function as in any other POLYRATE calculation, fixing the value of the selected coordinates and avoiding their optimization, as explained in Section 11.A.6 of the POLYRATE 9.3 manual.

INITGEO AND STATUS

There are two options for reading the initial geometries of reactants, products, wells, and saddle point: the first one is by means of units fu71 through fu78, if a GAMESSPLUS optimization is desired (OHOOK option for the OPTMIN and OPTTS keywords in the OPTIMIZATION section). In this case, the INITGEO keyword has to be set to HOOKS, and the files fu71-fu78 should be the appropriate input files for a GAMESSPLUS optimization. The second one is by means of the GEOM keyword, if a POLYRATE 9.3 optimization (BFGS or NR options in the OPTMIN or OPTTS keyword in the OPTIMIZATION section) or no optimization is going to be carried out. Then, the INITGEO keyword needs to be set to GEOM. Therefore, the initial geometry for the optimization of those systems that are going to be optimized using GAMESSPLUS methods should be given in units fu71-fu78 and the INITGEO keyword in the respective section has to be set to HOOKS, while the initial geometry for those systems which are going to be optimized using POLYRATE methods or the optimized geometry of the systems which are not going to be further optimized should be given in unit fu5 and the INITGEO keyword in the respective sections has to be set to GEOM.

The combination of the keywords INITGEO, STATUS and OPTTS and OPTMIN, therefore, will indicate the precedence of the initial and/or optimized geometries for the stationary points. The combinations are:

- For optimizing the geometries using the methods in GAMESSPLUS: OPTTS and OPTMIN in the OPTIMIZATION section have to be set to OHOOKS, INITGEO has to be set to HOOKS and STATUS set to 0 in the corresponding REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and/or START sections. The initial geometry should be given in the files fu71 through fu78.

- If the geometry optimization of one of the stationary points is not desired, the INITGEO keyword has to be set to GEOM, and the STATUS keyword should be set to an allowed value greater than zero in the corresponding REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections of the fu5 input file. The optimized geometry will be read from the fu5 input file.

- For optimizing the geometries using the methods in POLYRATE (not recommended): OPTTS and OPTMIN in the OPTIMIZATION section have to be set to BFGS and/or NR, and INITGEO has to be set to GEOM and STATUS set to 0 in the corresponding REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and/or START sections of the fu5 input file. The initial geometry will be given in the appropriate section of the fu5 input file. If the geometry of one of the stationary points is already optimized, the only parameter that has to be changed is the value of the STATUS keyword in the corresponding section of the fu5 input file.

A value greater than 0 for the STATUS keyword in any of the REACT1, REACT2, PROD1, PROD2 and/or START sections involves no optimization of the geometry input in the fu5 file and, if a value greater than 2 is selected, no Hessian calculation. If STATUS is set to 4 in the START section of the fu5 input file, the Hessian matrix as it appears in a GAMESSPLUS archive file can be input, according to the instructions in the POLYRATE manual.

ANHARMONICITY

Although anharmonic calculations can be carried out, the user must keep in mind the extremely high cost of numerical third or fourth derivatives of the energy when using ab initio methods. With the anharmonicity options available in POLYRATE 9.3, this extra cost is not usually warranted. We therefore recommend not using anharmonic options in the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections when they require derivatives of order higher than two, at least until some later version of POLYRATE with better anharmonicity options becomes available. Note, however, that the Hindered rotor and the Morse I option do not require derivatives higher than second.

SCALEMASS

If restart calculations are desired, the value of the SCALEMASS variable keyword must be the same for the whole series of calculations. Note that SCALEMASS is the reduced mass to which all isoinertial coordinates are scaled. See also the paragraph about SCALEMASS in Section 5 of this manual.

6.B. esp.fu71 THROUGH esp.fu78 FILES AND AVOIDING REORIENTATION

The data files esp.fu71 through esp.fu78 are GAMESSPLUS-type data files. They are used only as input files for GAMESSPLUS optimization on reactants 1 and 2 (esp.fu71 and esp.fu72, respectively), on products 1 and 2 (esp.fu73 and esp.fu74), on the well on the reactant side (esp.fu77), on the well on the product side (esp.fu78), and the saddle point or other starting geometry (esp.fu75). If no optimization using GAMESSPLUS is desired, these files are not needed. For constructing these files, the user must check the GAMESSPLUS and GAMESS manuals. One important restriction is that the input geometry has to have C1 symmetry and it has to be given in Cartesian coordinates. The electronic structure package GAMESS reorients the molecule in order to take the advantage of symmetry to facilitate the calculations. This reorientating procedure will cause errors in the dynamics calculations, for example, in the reaction path following algorithm and curvature component computations. Therefore, it is necessary to use C1 symmetry and Cartesian coordinates in the fu71-fu78 input files. Notice that this program will not check on this "required" feature, it is the user's responsibility to have correct input format. Please refer to the POLYRATE manual for further comments on the reorientation issue, which is a major source of potential error for users who do not read manuals.

As mentioned above, when creating these files, the order of the atoms in unit fu75 must be consistent with the order of the atoms input in the ATOMS keyword in the GENERAL section of the POLYRATE fu5 input file, and the order of the atoms in units fu71-fu78 must be consistent with the index specifications in the GEOM keyword in the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections of the fu5 input file.

For the sake of consistency, the method chosen for the electronic structure calculations for a given reaction must be the same in all the esp.fu71 through esp.fu78 files representing the species for that reaction. Only if one of the species is the hydrogen atom and electronic correlation is included can the quantum methods be different (since no correlation energy can be calculated for the hydrogen atom). The electronic structure method must also be consistent with the options related to energy and first and second derivatives chosen in the fu70 input file.

When studying a reaction with no saddle point, the user must insure that the optimization procedures chosen for the starting point will not lead the system to reactants or products. This can be accomplished by freezing the appropriate geometrical parameters. If no geometrical parameter is kept fixed, the optimization will lead to a point where the gradients are zero and the procedure for following the reaction path will not work successfully.

6.C. esp.fu70 FILE

Along with the input fu5 file, which controls the POLYRATE options, the fu70 input file will control all of the GAMESSPLUSRATE options and some of the GAMESSPLUS input options. The fu70 input file is divided into sections which, like the fu5 sections, are denoted by a header preceded by an asterisk (*). There are eight sections, each starting with GR in order to avoid confusion with similar sections in the fu5 input file. These sections are GRGENERAL, GRCOMMON, GRREACT1, GRREACT2, GRPROD1, GRPROD2, GRWELLR, GRWELLP, GRSTART and GRPATH. In these sections, the appropriate keywords have to be specified.

There are three type of keywords: switch keywords (which do not have any argument), variable keywords (which require one argument following the keyword), and list keywords (which require a list of options, with an END line). See the POLYRATE manual for a more thorough discussion of the three types of keywords.

Notice that the keyword options for the GAMESSPLUS program are read by using the style of namelist format. Therefore, for the GAMESSPLUS keyword options, it is REQUIRED to have a blank space at the beginning of the line, which has maximum of 79 charaters.

Section name	Description
-----	-----
GRGENERAL	Restart option
GRDUAL	Dual level control
GRSOLV	Type of calculation
GRBASIS	Read in basis set information
GRCOMMON	GAMESSPLUS options common to all the stationary points
GRREACT1	GAMESSPLUS options specific to the first reactant
GRREACT2	GAMESSPLUS options specific to the second reactant, if any
GRPROD1	GAMESSPLUS options specific to the first product
GRPROD2	GAMESSPLUS options specific to the second product, if any
GRWELLR	GAMESSPLUS options specific to the reactant well, if any
GRWELLP	GAMESSPLUS options specific to the product well, if any
GRSTART	GAMESSPLUS options specific to the starting (or saddle) point
GRPATH	GAMESSPLUS options common to all the nonstationary points along the reaction path

6.C.1. SECTION GRGENERAL

The GRGENERAL section controls GAMESSPLUSRATE's own options. There are only two keywords allowed in this section, GRRESTART and NOGRRESTART, with GRRESTART being the default.

GRRESTART (NOGRRESTART)

This is a switch keyword that tells the program to check the restart file esp.fu83 and search for information about the point to be calculated. If it finds the required information about that point, the program writes it in the esp.fu84 file and uses the stored information instead of calculating it. If the information needed is not present in the fu83 file, GAMESSPLUSRATE runs GAMESSPLUS and writes the GAMESSPLUS output information to unit fu84. If NOGRRESTART is selected, neither reading file fu83 nor writing file fu84 will be performed.

6.C.2. SECTION GRDUAL

The GRDUAL section reads the information for the hybrid Hartree-Fock and Molecular Orbital (HF||MO) calculation. The mixing parameter x is specified by the keyword PARAMX. And the specific information for the molecular orbital method and Hartree-Fock calculations is indicated by using the LLEVEL keyword for the molecular orbital method (for example, AM1 or PM3) and the HLEVEL keyword for the Hartree-Fock method.

6.C.3. SECTION GRSOLV

The GRSOLV section indicates the type of calculation. First, it is required to indicate whether the calculation is GAS, SES, or ESP. If NES approximation is carried, ESP keyword is required in this section. Second, if the SES option is used, the keyword options HMETHOD and LMETHOD are used for indicating the solvation model for evaluating the the free energy of solvation using the linear combination of the HF and MO methods, or the keyword METHOD is used for a single-level calculation. If the calculation is of the ESP type, the \$CM2 namelist can be included either in the GRDUAL section (using HLEVEL and LLEVEL keywords) for a dual-level calculation or in the GRCOMMON, GRREACT1, GRREACT2, GRPROD1, GRPROD2, GRWELLR, GRWELLP, GRSTART, and/or GRPATH sections.

6.C.4. SECTION GRBASIS

The GRBASIS section reads in the user-defined basis set information. Notice that GAMESSPLUSRATE has GRBASIS as a separate section, whereas GAUSSRATE includes the similar option in the GRCOMMON section. Inside this section the basis functions for a specific atom type are defined as in the following example (which corresponds to the 21G basis for H):

```

BASIS
  <atomic number>
  <L value of contracted>  <# of primitive gaussians>
  <ordinal of primitive function>  <exponent> <coefficient>
  ...
END

```

For example, the MIDI! basis set of H atom

```

BASIS
  1
S   2
  1      4.50180000      0.07045200
  2      0.68144400      0.40782600
S   1
  1      0.15139800      1.000000
END

```

For a dual-level calculation, the basis set is defined by using the HBASIS and LBASIS keywords for the higher-level and lower-level respectively. For a single-level calculation, the basis set is defined by using the BASIS keyword. Also, the GRBASIS section must be read before the GRCOMMON section in order to keep the correct default values.

6.C.5. SECTION GRCOMMON

The GRCOMMON section reads the information common to all the GAMESSPLUS calculations for the stationary points and points along the reaction path, except for the GAMESSPLUS geometry optimizations, which are totally controlled by the fu71 through fu78 input file. The options selected in this section will be applied to all the GAMESSPLUS calculations unless a different keyword is specified in the section corresponding to a specific point. This section needs to precede the specific sections GRREACT1, GRREACT2, GRPROD1, GRPROD2, GRSTART, GRWELLR, GRWELLP, and GRPATH. The following keywords are allowed in the GRCOMMON section:

GRENER

This is a list type keyword and therefore it must include an END line to indicate the end of the list (see the POLYRATE manual for a description of list keywords). GRENER includes the keywords that GAMESSPLUS will use for carrying out an energy calculation. The default is set to be an RHF/STO-3G calculation. Notice that each line of the GAMESSPLUS keywords has to start with a blank space, because the GAMESSPLUS keywords are parsed in the style of namelist format. No blank lines can appear in this list, since GAMESSPLUS would take the blank lines as the end of its input.

For example:

```

*GRCOMMON
GRENER
  $CONTRL SCFTYP=RHF RUNTYP=ENERGY $END
  $GUESS  GUESS=HUCKEL $END
END

```

GRFIRST

This accomplishes a purpose similar to that of the GRENER keyword, but it indicates the keywords to be used in gradient calculations. The default is set to be an RHF/STO-3G calculation.

GRSEC

This keyword is only used when the HHOOK option is specified in the HESSCAL keyword of the SECOND section of the fu5 input file. It accomplishes a purpose similar to the GRENER and GRFIRST keywords, but indicating the options for a Hessian calculation. The default is set to be an RHF/STO-3G calculation.

Although the above example illustrates what will probably be the most common use for this keyword, actually any information that needs to be located in a GAMESSPLUS input file after the geometry specification can be included by means of the GRBASIS keyword. For example, electronic excited states require input information indicating the orbitals involved in the electronic excitation; this information can be entered by this keyword with two limitations: The information will be included in all the GAMESSPLUS inputs (except the optimizations, which are read directly from units 71-78), and no blank lines are allowed in this keyword. The second limitation can be overcome by using the external file option mentioned in the previous paragraph.

It should be noted that the isotope input, which must be contained in this section of the GAMESSPLUS input file if nonstandard isotopes are desired, should never be included in a GAMESSPLUSRATE run. The choice of the isotopic masses has to be done in the fu5 input file of POLYRATE.

6.C.6. SECTIONS GRREACT1, GRREACT2, GRPROD1, GRPROD2, GRWELLR, GRWELLP, GRSTART, AND GRPATH

Since in some cases, the options selected in the GRCOMMON section might not be the most useful for all the points to be calculated, these eight sections allow the user to select different keywords for the energy, gradients, and Hessian calculations by means of the GRENER, GRFIRST, and GRSEC list keywords. The options for the first reactant should be specified in the GRREACT1 section; in the GRREACT2 the options for the second reactant, if it exists; in GRPROD1 and GRPROD2 the options for the first product and, if it exists, the second one; in GRWELLR the options for a well on the reactant side of the reaction path, if it exists; in GRWELLP the options for a well on the product side, if it exists; section GRSTART will indicate the options for the saddle point or starting geometry, and GRPATH will indicate the options for GAMESSPLUS calculations on the non-stationary points along the reaction path. If no changes from the options selected in the GRCOMMON section are wanted, these sections can be skipped.

7. TESTING GAMESSPLUSRATE

7.A. TESTS PERFORMED

We tested version 9.3 of GAMESSPLUSRATE using version 9.3 of POLYRATE and the indicated versions of GAMESSPLUS, computing machines, and the operating systems listed on page 8-1.

7.B. TEST SUITE

The purpose of the test suite is to provide some examples of the way GAMESSPLUSRATE works, not to check all GAMESSPLUS or POLYRATE 9.3 options. Therefore, the number of tests included is only ten, and the tests illustrate only the most basic features of GAMESSPLUSRATE. The level of calculation, size of the systems, and computational requirements of the test runs have been chosen in such a way that running the whole suite does not consume an inordinate amount of computing time. All the test runs were run after compiling GAMESSPLUSRATE using the param4.inc file.

In the directory '~testrun/GAS' there are test runs corresponding to five reactions:

Reaction	Directory	Test names
CH3 + H2 -> CH4 + H	ch5	ch5tr1, ch5tr2
CH4 + OH -> CH3 + H2O	ch4oh	ch4ohtr1, ch4ohtr2, ch4ohtr3, ch4ohtr4, ch4ohtr5
NH3 + OH -> NH2 + H2O	nh3oh	nh3ohtr1
CH3OH + H -> H2 + CH3	ch3ohh	ch3ohhtr1, ch3ohhtr2
CH3Cl + Cl- -> Cl- + CH3Cl	ch3cl2	ch3cl2tr1

In the directory '~testrun/SES' there are test runs corresponding to four reactions:

Reaction	Directory	Test names
CH3OH + H -> H2 + CH3	ch3ohh	ch3ohhtr1
CH3Cl + Cl- -> Cl- + CH3Cl	ch3cl2	ch3cl2tr1
CH3Cl + NH3 -> Cl- + CH3NH3+	ch3clnh3	ch3nh3tr1
HCOOH + HCOOH' -> HCOOH' + HCOOH	2hcooh	2hcootr1

In the directory '~testrun/ESP' there are four test runs corresponding to four reactions and one test run corresponding to a geometry optimization based on SM5.43/HF/6-31G*.

Reaction (or system)	Directory	Test names
CH3OH + H -> H2 + CH3	ch3ohh	ch3ohhtr1
CH3Cl + Cl- -> Cl- + CH3Cl	ch3cl2	ch3cl2tr1
CH3Cl + NH3 -> Cl- + CH3NH3+	ch3clnh3	ch3nh3tr1
CH3 + C2H4 -> CH3C2H4	ch3c2h4	ch3c2h4tr1
CH3CHO	ch3cho	ch3chotr1

In the directory '~testrun/NES' there are test runs corresponding to two reactions:

Reaction	Directory	Test names
CH3OH + H -> H2 + CH3	ch3ohh	ch3ohhtr1
CH3Cl + Cl- -> Cl- + CH3Cl	ch3cl2	ch3cl2tr1

7.B.1 TEST RUN ch5tr1

Potential : HF/STO-3G
 Vibrations : harmonic rectilinear
 RODS : no
 Dynamics : TST, CVT
 Tunneling : ZCT, SCT
 Dual Level : no
 Solvation : no

This sample run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ gives the same results as the test run ch5fu30tr5 included in the POLYRATE 9.3 test suite. It performs calculations of the TST, CVT, and ICVT rate constants at the UHF/STO-3G level, and it is based on following the reaction path using the Euler steepest descents method (also called Euler single-step method). For all the stationary points symmetry is employed in all the GAMESSPLUS calculations (which is the default for GAMESSPLUS calculations); nevertheless, this option is turned off when calculating points along the reaction coordinate in order to avoid possible problems due to the reorientation of the system. A restart file is created (GRRESTART keyword), and all the input of the geometries is done by means of files fu71 through fu75, in order to allow GAMESSPLUS to optimize the stationary points (OHOOK option in OPTTS and OPTMIN in the OPTIMIZATION section of the fu5 input file, and INITGEO set to HOOKS in the REACT1, REACT2, PROD1, PROD2, and START sections in the input file fu5).

GAMESSPLUSRATE I/O FILES

GAMESSPLUSRATE FILENAMES

ch5tr1.70	esp.fu70	Input data for GAMESSPLUSRATE
ch5tr1.71	esp.fu71	Input data for reactant 1
ch5tr1.72	esp.fu72	Input data for reactant 2
ch5tr1.73	esp.fu73	Input data for product 1
ch5tr1.74	esp.fu74	Input data for product 2
ch5tr1.75	esp.fu75	Input data for the saddle point
ch5tr1.dat	poly.fu5	Input data for POLYRATE
ch5tr1.61	esp.fu61	POLYRATE output file containing information about the stationary points
ch5tr1.fu6	poly.fu6	Long output file
ch5tr1.fu15	poly.fu15	Summary output file
ch5tr1.rst	esp.fu84	Restart file written by

GAMESSPLUSRATE

7.B.2 TEST RUN ch5tr2

Potential : HF/STO-3G
 Vibrations : harmonic rectilinear
 RODS : no
 Dynamics : TST, CVT
 Tunneling : ZCT, SCT
 Dual Level : ISPE
 Solvation : no

This test run uses the restart file created by the previous run and performs a calculation using the ISPE technique by using ten MP2/6-31G** energies input in the esp.fu51 input file. Of these ten energies, two are for products and the saddle point, and eight are for nonstationary points. Since the ISPE technique does not modify the reaction path, the calculation can be restarted successfully. The user must run the ch5tr1 test run before running ch5tr2, in order to create the restart file to be read, ch5tr2.rst, which will be a copy of the ch5tr1.rst file generated by the previous run and located in the same directory.

GAMESSPLUSRATE I/O FILES

GAMESSPLUSRATE FILENAMES

ch5tr2.51	poly.fu51	Input file for ISPE information
ch5tr2.70	esp.fu70	Input data for GAMESSPLUSRATE
ch5tr2.71	esp.fu71	Input data for reactant 1
ch5tr2.72	esp.fu72	Input data for reactant 2
ch5tr2.73	esp.fu73	Input data for product 1
ch5tr2.74	esp.fu74	Input data for product 2
ch5tr2.75	esp.fu75	Input data for the saddle point
ch5tr2.dat	poly.fu5	Input data for POLYRATE
ch5tr2.61	esp.fu61	POLYRATE output file containing information about the stationary points
ch5tr2.fu6	poly.fu6	Long output file
ch5tr2.fu15	poly.fu15	Summary output file
ch5tr2.rst.old	esp.fu83	Restart file read by GAMESSPLUSRATE
ch5tr2.rst	esp.fu84	Restart file written by
GAMESSPLUSRATE		

7.B.3 TEST RUN ch4ohtr1

Potential : PM3
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST, CVT
Tunneling : ZCT, SCT
Dual Level : no
Solvation : no

This is the first of a series of tests that try to show how GAMESSPLUSRATE would usually be used. We study the $\text{CH}_4 + \text{OH} \rightarrow \text{CH}_3 + \text{H}_2\text{O}$ reaction. The sequence actually has five stages. The first four stages may be thought of a complete single-level calculation or as runs 1-4 of a five-run dual-level calculation. The first step we will take is the calculation of the first reactant, CH_4 . In the fu5 input file we only include the REACT1 section, and only the additional input files fu70 and fu71 are required. The GAMESSPLUSRATE restart option is off, since the restart is only available when all the information about the stationary points is read from the restart file or written in it.

The output file fu61 will contain the information calculated for this reactant, and this information will be included in the fu5 input file for the following runs.

GAMESSPLUSRATE I/O FILES	GAMESSPLUSRATE FILENAMES	
ch4ohtr1.70	esp.fu70	Input data for GAMESSPLUSRATE
ch4ohtr1.71	esp.fu71	Input data for reactant 1
ch4ohtr1.dat	poly.fu5	Input data for POLYRATE
ch4ohtr1.61	esp.fu61	POLYRATE output file containing information about reactant 1
ch4ohtr1.fu6	poly.fu6	Long output file

7.B.4 TEST RUN ch4ohtr2

Potential : PM3
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST, CVT
Tunneling : ZCT, SCT
Dual Level : no
Solvation : no

This is the second run in the four-stage single-level calculation for the CH₄ + OH → CH₃ + H₂O reaction. The information included in the fu61 output file from the previous run is cut and pasted to the fu5 input file for the present run. The properties of the other reactant, OH, and the products, CH₃ and H₂O, are calculated here. Therefore, the fu5 input file will not contain the START section, and the input files fu70, fu72, fu73, and fu74 are required. Once again, the GAMESSPLUSRATE restart option is off.

Usually, the user would calculate the stationary points one by one, instead of calculating three in a row. Thus, if some problem is encountered in any stationary point calculation, the user can check the error in that particular point, instead of repeating several optimizations.

The fu61 input file will contain all the information about the reactants and products needed for the next run. This includes both the information calculated in test run ch4ohtr1 and the information calculated in the present test run.

GAMESSPLUSRATE I/O FILES

GAMESSPLUSRATE FILENAMES

ch4ohtr2.70	esp.fu70	Input data for GAMESSPLUSRATE
ch4ohtr2.72	esp.fu72	Input data for reactant 2
ch4ohtr2.73	esp.fu73	Input data for product 1
ch4ohtr2.74	esp.fu74	Input data for product 2
ch4ohtr2.dat	poly.fu5	Input data for POLYRATE
ch4ohtr2.61	esp.fu61	POLYRATE output file containing information about the stationary points
ch4ohtr2.fu6	poly.fu6	Long output file

7.B.5 TEST RUN ch4ohtr3

Potential : PM3
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST, CVT
Tunneling : ZCT, SCT
Dual Level : no
Solvation : no

This is the third run in the four-stage single-level calculation for the CH₄ + OH → CH₃ + H₂O reaction. In this test run, we calculate the saddle point properties and evaluate the conventional TST rate constant, including all the information on reactants and products in the fu5 input file, as they appeared in the fu61 output file from the previous run.

GAMESSPLUSRATE I/O FILES

GAMESSPLUSRATE FILENAMES

ch4ohtr3.70	esp.fu70	Input data for GAMESSPLUSRATE
ch4ohtr3.75	esp.fu74	Input data for the saddle point
ch4ohtr3.dat	poly.fu5	Input data for POLYRATE
ch4ohtr3.61	esp.fu61	POLYRATE output file containing information about the stationary points
ch4ohtr3.fu6	poly.fu6	Long output file

Potential : PM3
Vibrations : harmonic redundant curvilinear, hindered rotor treatment of the
lowest vibrational mode
RODS : no
Dynamics : TST, CVT
Tunneling : ZCT, SCT
Dual Level : no
Solvation : no

7.B.6 TEST RUN ch4ohtr4

This is the fourth and final run in the four-stage calculation for the $\text{CH}_4 + \text{OH} \rightarrow \text{CH}_3 + \text{H}_2\text{O}$ reaction. Actually this fourth stage may be considered either as stage 4 of a complete four-stage single-level calculation or as the fourth stage in a five-stage dual-level calculation. In this test run we use the information about the stationary points appearing in the fu61 output file from the ch4ohtr3 test run and evaluate the reaction path, CVT, CVT/ZCT and CVT/SCT rate constants using Euler method.

We assumed the hindered rotor approximation for treating the lower vibrational mode along the reaction path, and we calculated the vibrational frequencies by using redundant internal coordinates. The reduced moment of inertia for the hindered rotor is evaluated using Pitzer's method.

GAMESSPLUSRATE I/O FILES	GAMESSPLUSRATE FILENAMES	
ch4ohtr4.70	esp.fu70	Input data for GAMESSPLUSRATE
ch4ohtr4.dat	poly.fu5	Input data for POLYRATE
ch4ohtr4.61	esp.fu61	POLYRATE output file containing information about the stationary points
ch4ohtr4.fu6	poly.fu6	Long output file
ch4ohtr4.fu15	poly.fu15	Summary output file
ch4ohtr4.rst	poly.fu84	Restart file

7.B.7 TEST RUN ch4ohtr5

Potential : PM3
Vibrations : harmonic rectilinear, hindered rotor treatment of the lowest vibrational mode
RODS : no
Dynamics : TST, CVT
Tunneling : ZCT, SCT, LCT
Dual Level : ICA/SECKART
Solvation : no

This test run completes an IOC dual-level calculation of the CH₄ + OH -> CH₃ + H₂O reaction. The lower-level information is read from the restart file from the previous run. The higher-level data is included by means of the fu50 file. Tunneling is calculated by the microcanonical optimized multidimensional tunneling method (muOMT), requiring therefore some calculation in the reaction-swath region that are not present in the restart file, since the tunneling for the previous run was calculated using the SCT approach.

GAMESSPLUSRATE I/O FILES

GAMESSPLUSRATE FILENAMES

ch4ohtr5.70	esp.fu70	Input data for GAMESSPLUSRATE
ch4ohtr4.rst	esp.fu83	Restart file read by GAMESSPLUSRATE
ch4ohtr5.dat	poly.fu5	Input data for POLYRATE
ch4ohtr5.50	poly.fu50	Additional input data for the POLYRATE VTST-IC calculation
ch4ohtr5.61	esp.fu61	POLYRATE output file containing information about the stationary points
ch4ohtr5.fu6	poly.fu6	Long output file
ch4ohtr5.fu15	poly.fu15	Summary output file
ch4ohtr5.rst	esp.fu84	Restart file written by
GAMESSPLUSRATE		

7.B.8 TEST RUN nh3ohtr1

Potential : PM3
Vibrations : harmonic rectilinear
RODS : no
Dynamics : TST
Tunneling : no
Dual Level : no
Solvation : no

This is a test run for the $\text{NH}_3 + \text{OH} \rightarrow \text{NH}_2 + \text{H}_2\text{O}$ reaction including a well on the reactant side and a well on the product side. The geometries are optimized by means of GAUSSIAN methods. Therefore, the initial geometries and keywords for the optimization are given in the input files esp.fu71-75, esp.fu77, and esp.fu78, and INITGEO is set to HOOKS and STATUS to zero in the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections of the fu5 input file. The Hessian matrices are also calculated, as recommended, by using the methods in GAMESSPLUS. With the information for the stationary points obtained in this way, a conventional TST rate constant without tunneling is calculated.

GAMESSPLUSRATE I/O FILES

GAMESSPLUSRATE FILENAMES

nh3ohtr1.70	esp.fu70	Input data for GAMESSPLUSRATE
nh3ohtr1.71	esp.fu71	Input data for reactant 1
nh3ohtr1.72	esp.fu72	Input data for reactant 2
nh3ohtr1.73	esp.fu73	Input data for product 1
nh3ohtr1.74	esp.fu74	Input data for product 2
nh3ohtr1.75	esp.fu75	Input data for the saddle point
nh3ohtr1.77	esp.fu77	Input data for the reactants well
nh3ohtr1.78	esp.fu78	Input data for the products well
nh3ohtr1.dat	poly.fu5	Input data for POLYRATE
nh3ohtr1.61	esp.fu61	POLYRATE output file containing information about the stationary points
nh3ohtr1.fu6	poly.fu6	Long output file
nh3ohtr1.fu15	poly.fu15	Summary output file

7.B.9 TEST RUN ch3ohhtr1

Potential : HF/MIDI!
Vibrations : harmonic rectilinear
RODS : no
Dynamics : TST
Tunneling : no
Dual Level : no
Solvation : no

This is a test run for the $\text{CH}_3\text{OH} + \text{H} \rightarrow \text{H}_2 + \text{CH}_2\text{OH}$ reaction at HF/MIDI!6D level. This is a TST-only calculation which serves as an example for reading the specific basis information using GRBASIS keyword option in esp.fu70 file. The gas-phase reaction rate constant is only evaluated at 298 K with the forward and backward symmetry numbers equal to 2.

GAMESSPLUSRATE I/O FILES

GAMESSPLUSRATE FILENAMES

ch3ohhtr1.70	esp.fu70	Input data for GAMESSPLUSRATE
ch3ohhtr1.dat	poly.fu5	Input data for POLYRATE
ch3ohhtr1.fu6	poly.fu6	Long output file
ch3ohhtr1.fu15	poly.fu15	Summary output file

7.B.9 TEST RUN ch3ohhtr2

Potential : HF/STO-3G||AM1-SRP
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST, CVT
Tunneling : ZCT, SCT, LCT
Dual Level : no
Solvation : no

This is a test run for the $\text{CH}_3\text{OH} + \text{H} \rightarrow \text{H}_2 + \text{CH}_2\text{OH}$ reaction at the HF/STO-3G||AM1-SRP ($x = 0.244$) level. In this calculation, the reaction path is followed using the Page-McIver algorithm with the gradient step size of 0.02 a0. The vibrational frequencies are evaluated at every 4th gradient step with redundant internal coordinates. The geometries of the stationary points are optimized using the EF algorithm. TST, CVT, CVT/ZCT, CVT/SCT, CVT/LCT, and CVT/mOMT dynamics calculations are carried out for a set of temperatures in the range from 200 K to 2400 K. The forward and backward symmetry numbers are equal to 2, and the SCALEMASS variable is set to 1.0 amu.

GAMESSPLUSRATE I/O FILES

GAMESSPLUSRATE FILENAMES

SRP	SRP	NDDO-SRP file
ch3ohhtr2.70	esp.fu70	Input data for GAMESSPLUSRATE
ch3ohhtr2.dat	poly.fu5	Input data for POLYRATE
ch3ohhtr2.fu6	poly.fu6	Long output file
ch3ohhtr2.fu15	poly.fu15	Summary output file

7.B.10 TEST RUN ch3cl2tr1

Potential : AM1
Vibrations : harmonic rectilinear
RODS : no
Dynamics : TST, CVT, ICVT
Tunneling : ZCT, SCT
Dual Level : no
Solvation : no

This sample input is for an AM1 direct dynamics calculation on the $\text{Cl}^- + \text{CH}_3\text{Cl} \rightarrow \text{CH}_3\text{Cl} + \text{Cl}^-$ exoergic $\text{S}_\text{N}2$ reaction. All normal modes and generalized normal modes are treated using the harmonic approximation with redundant curvilinear coordinates. The Page-McIver integration method with the cubic starting algorithm is used. A gradient step size of $5.E-3$ bohr is used to follow the MEP, and a save step size of $1.0E-2$ bohr is used between generalized normal mode analyses. A restart file containing reaction-path information is written to unit fu1. No tunneling contributions are computed, and only forward CVT reaction rate constants are calculated at two temperatures.

GAMESSPLUSRATE I/O Files

GAMESSPLUSRATE Filenames

ch3cl2tr1.70	esp.fu70	input data for phase
ch3cl2tr1.71	esp.fu71	input data for reactant 1
ch3cl2tr1.72	esp.fu72	input data for reactant 2
ch3cl2tr1.73	esp.fu73	input data for product 1
ch3cl2tr1.74	esp.fu74	input data for product 2
ch3cl2tr1.75	esp.fu75	input data for the saddle point
ch3cl2tr1.dat	poly.fu5	input data for GAMESSPLUSRATE
ch3cl2tr1.fu6	poly.fu6	long output file
ch3cl2tr1.fu1	poly.fu1	reaction path information
ch3cl2tr1.fu15	poly.fu15	summary output file

7.C. LIQUID-PHASE REACTIONS USING SES

7.C.1. TEST RUN ch3ohhtr1

Potential : HF/STO-3G||AM1-SRP
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST, CVT
Tunneling : ZCT, SCT, LCT
Dual Level : no
Solvation : SES, water

This is a test run for the $\text{CH}_3\text{OH} + \text{H} \rightarrow \text{H}_2 + \text{CH}_2\text{OH}$ reaction at the HF/STO-3G||AM1-SRP ($x = 0.244$) level. In this calculation, the reaction path is followed using the Page-McIver algorithm with the gradient step size set equal to 0.02 bohr. The vibrational frequencies are evaluated at every 4th gradient step with redundant internal coordinates. The geometries of the stationary points are optimized using the EF algorithm. TST, CVT, CVT/ZCT, CVT/SCT, CVT/LCT, and CVT/mOMT dynamics calculations are carried at a set of temperatures in the range from 200 K to 2400 K. The forward and backward symmetry numbers are equal to 2, and the SCALEMASS variable is set equal to 1.0 amu.

GAMESSPLUSRATE I/O FILES

GAMESSPLUSRATE FILENAMES

SRP	SRP	NDDO-SRP file for AM1-SRP
CM2SRPH	CM2SRPH	CM2SRP file for HF/STO-3G
CM2SRPL	CM2SRPL	CM2SRP file for AM1-SRP
ch3ohhtr1.70	esp.fu70	Input data for GAMESSPLUSRATE
ch3ohhtr1.dat	poly.fu5	Input data for POLYRATE
ch3ohhtr1.fu6	poly.fu6	Long output file
ch3ohhtr1.fu15	poly.fu15	Summary output file

7.C.2. TEST RUN ch3cl2tr1

Potential : AM1
Vibrations : harmonic rectilinear
RODS : no
Dynamics : TST, CVT, ICVT
Tunneling : ZCT, SCT
Dual Level : no
Solvation : SES, water

This sample input is for an SES dynamics calculation for the reaction $\text{CH}_3\text{Cl} + \text{Cl}^- \rightarrow \text{CH}_3\text{Cl} + \text{Cl}^-$ in water. The harmonic vibrational frequencies are evaluated using the SM5.42/AM1 model. The forward rate constant is calculated at 298 K. The reaction path is followed with the Page-McIver algorithm with a step size of 0.01 bohr over the range of $-0.5 \text{ bohr} < s < 0.5 \text{ bohr}$.

Both ICVT and CVT calculations are carried with SCT tunneling corrections.

GAMESSPLUSRATE I/O Files

GAMESSPLUSRATE Filenames

ch3cl2tr1.70	esp.fu70	input data for phase
ch3cl2tr1.71	esp.fu71	input data for reactant 1
ch3cl2tr1.72	esp.fu72	input data for reactant 2
ch3cl2tr1.73	esp.fu73	input data for product 1
ch3cl2tr1.74	esp.fu74	input data for product 2
ch3cl2tr1.75	esp.fu75	input data for the saddle point
ch3cl2tr1.dat	poly.fu5	input data for GAMESSPLUSRATE
ch3cl2tr1.fu6	poly.fu6	long output file
ch3cl2tr1.fu15	poly.fu15	summary output file

7.C.3. TEST RUN ch3clnh3tr1

Potential : PM3
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST
Tunneling : no
Dual Level : no
Solvation : SES, water

This sample input is for an SES dynamics calculation for the reaction
 $\text{CH}_3\text{Cl} + \text{NH}_3 \rightarrow \text{CH}_3\text{NH}_3^+ + \text{Cl}^-$ in water. The harmonic vibrational frequencies
are evaluated using the SM5.42/PM3 method. The forward rate constant is
calculated at 298 K.

GAMESSPLUSRATE I/O Files

GAMESSPLUSRATE Filenames

ch3clnh3tr1.70	esp.fu70	input data for phase
ch3clnh3tr1.71	esp.fu71	input data for reactant 1
ch3clnh3tr1.72	esp.fu72	input data for reactant 2
ch3clnh3tr1.73	esp.fu73	input data for product 1
ch3clnh3tr1.74	esp.fu74	input data for product 2
ch3clnh3tr1.75	esp.fu75	input data for the saddle point
ch3clnh3tr1.dat	poly.fu5	input data for GAMESSPLUSRATE
ch3clnh3tr1.fu6	poly.fu6	long output file
ch3clnh3tr1.fu15	poly.fu15	summary output file

7.C.4. TEST RUN ch3clnh3tr2

Potential : PM3
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST
Tunneling : no
Dual Level : no
Solvation : SES, water

This sample input is for an SES dynamics calculation for the reaction $\text{CH}_3\text{Cl} + \text{NH}_3 \rightarrow \text{CH}_3\text{NH}_3^+ + \text{Cl}^-$ in water. The harmonic vibrational frequencies are evaluated using the SM5.42/PM3 method. The forward rate constant is calculated at 298 K. Since this is a test for using the optimizer in POLYRATE, only TST reaction rate constants are calculated.

GAMESSPLUSRATE I/O Files

GAMESSPLUSRATE Filenames

ch3clnh3tr2.70	esp.fu70	input data for phase
ch3clnh3tr2.dat	poly.fu5	input data for GAMESSPLUSRATE
ch3clnh3tr2.fu6	poly.fu6	long output file
ch3clnh3tr2.fu15	poly.fu15	summary output file

7.C.5. TEST RUN 2hcoohtr1

Potential : PM3
Vibrations : harmonic rectilinear
RODS : no
Dynamics : TST, CVT
Tunneling : ZCT, SCT, LCT
Dual Level : no
Solvation : SES, water

This sample input is for an SES dynamics calculation for the double proton-transfer reaction $\text{HCOOH} + \text{HCOOH}' \rightarrow \text{HCOOH}' + \text{HCOOH}$ in water. The harmonic vibrational frequencies are evaluated using the PM3 model. The forward rate constant is calculated at 298 K. ZCT, SCT, LCT, and muOMT tunneling corrections are calculated with CVT. The reaction path is followed with the Page-McIver method with a step size of 0.01 bohr within the range $-1.5 \text{ bohr} < s < 1.5 \text{ bohr}$.

GAMESSPLUSRATE I/O Files

GAMESSPLUSRATE Filenames

2hcoohtr1.70	esp.fu70	input data for phase
2hcoohtr1.71	esp.fu71	input data for reactant 1
2hcoohtr1.72	esp.fu72	input data for reactant 2
2hcoohtr1.73	esp.fu73	input data for product 1
2hcoohtr1.74	esp.fu74	input data for product 2
2hcoohtr1.75	esp.fu75	input data for the saddle point
2hcoohtr1.dat	poly.fu5	input data for GAMESSPLUSRATE
2hcoohtr1.fu6	poly.fu6	long output file
2hcoohtr1.fu15	poly.fu15	summary output file

7.D. SOLUTION PHASE REACTIONS USING ESP

7.D.1. TEST RUN ch3ohhtr1

Potential : HF/3-21G||AM1-SRP
Vibrations : harmonic redundant curvilinear
RODS : yes
Dynamics : TST, CVT
Tunneling : ZCT, SCT, LCT
Dual Level : no
Solvation : ESP, water

This is a test run for the $\text{CH}_3\text{OH} + \text{H} \rightarrow \text{H}_2 + \text{CH}_2\text{OH}$ reaction at the SM5.42//HF/3-21G||AM1-SRP ($x = 0.276$) level. In this calculation, the reaction path is followed using the Page-McIver algorithm with the gradient step size of 0.02 bohr. The vibrational frequencies are evaluated at every 4th gradient step with redundant internal coordinates. The geometries of the stationary points are optimized using the EF algorithm. TST, CVT, CVT/ZCT, CVT/SCT, CVT/LCT, and CVT/mOMT dynamics calculations are carried at 298 K. The forward and backward symmetry numbers are set equal to 2, and the SCALEMASS variable is set equal to 1.0 amu.

GAMESSPLUSRATE I/O FILES

GAMESSPLUSRATE FILENAMES

TSRP	TSRP	NDDO-SRP file for AM1-SRP
CM2SRPH	CM2SRPH	CM2SRP file for HF/STO-3G
CM2SRPL	CM2SRPL	CM2SRP file for AM1-SRP
ch3ohhtr1.70	esp.fu70	Input data for GAMESSPLUSRATE
ch3ohhtr1.dat	poly.fu5	Input data for POLYRATE
ch3ohhtr1.fu6	poly.fu6	Long output file
ch3ohhtr1.fu15	poly.fu15	Summary output file

7.D.2. TEST RUN ch3cl2tr1

Potential : AM1
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST, CVT, ICVT
Tunneling : ZCT, SCT
Dual Level : no
Solvation : ESP, water

This sample input is for an ESP dynamics calculation for the reaction $\text{CH}_3\text{Cl} + \text{Cl}^- \rightarrow \text{CH}_3\text{Cl} + \text{Cl}^-$ in water. The harmonic vibrational frequencies are evaluated using the SM5.42/AM1 model. The forward rate constant is calculated at 298 K. The reaction path is followed with the Page-McIver algorithm with a step size of 0.01 bohr in the range of $-0.5 \text{ bohr} < s < 0.5 \text{ bohr}$.

Both ICVT and CVT calculations are carried with SCT tunneling corrections.

GAMESSPLUSRATE I/O Files

GAMESSPLUSRATE Filenames

ch3cl2tr1.70	esp.fu70	Input data for phase
ch3cl2tr1.71	esp.fu71	Input data for reactant 1
ch3cl2tr1.72	esp.fu72	Input data for reactant 2
ch3cl2tr1.73	esp.fu73	Input data for product 1
ch3cl2tr1.74	esp.fu74	Input data for product 2
ch3cl2tr1.75	esp.fu75	Input data for the saddle point
ch3cl2tr1.dat	poly.fu5	Input data for GAMESSPLUSRATE
ch3cl2tr1.fu6	poly.fu6	Long output file
ch3cl2tr1.fu15	poly.fu15	Summary output file

7.D.3. TEST RUN ch3clnh3tr1

Potential : PM3
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST
Tunneling : no
Dual Level : no
Solvation : ESP, water

This sample input is for an ESP dynamics calculation for the reaction
 $\text{CH}_3\text{Cl} + \text{NH}_3 \rightarrow \text{CH}_3\text{NH}_3^+ + \text{Cl}^-$ in water. The harmonic vibrational frequencies
are evaluated using the SM5.42/PM3 model. The forward rate constant is
calculated at 298 K.

GAMESSPLUSRATE I/O Files	GAMESSPLUSRATE Filenames	
ch3clnh3tr1.70	esp.fu70	Input data for phase
ch3clnh3tr1.71	esp.fu71	Input data for reactant 1
ch3clnh3tr1.72	esp.fu72	Input data for reactant 2
ch3clnh3tr1.73	esp.fu73	Input data for product 1
ch3clnh3tr1.74	esp.fu74	Input data for product 2
ch3clnh3tr1.75	esp.fu75	Input data for the saddle point
ch3clnh3tr1.dat	poly.fu5	Input data for GAMESSPLUSRATE
ch3clnh3tr1.fu6	poly.fu6	Long output file
ch3clnh3tr1.fu15	poly.fu15	Summary output file

7.D.4. TEST RUN ch3clnh3tr2

Potential : PM3
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST
Tunneling : no
Dual Level : no
Solvation : ESP

This sample input is for an ESP dynamics calculation for the reaction $\text{CH}_3\text{Cl} + \text{NH}_3 \rightarrow \text{CH}_3\text{NH}_3^+ + \text{Cl}^-$ in water. The harmonic vibrational frequencies are evaluated using the SM5.42/PM3 method. The forward rate constant is calculated at 298 K. Since this is a test for using the optimizer in POLYRATE, only TST reaction rate constants are calculated.

GAMESSPLUSRATE I/O Files

GAMESSPLUSRATE Filenames

ch3clnh3tr2.70	esp.fu70	input data for phase
ch3clnh3tr2.dat	poly.fu5	input data for GAMESSPLUSRATE
ch3clnh3tr2.fu6	poly.fu6	long output file
ch3clnh3tr2.fu15	poly.fu15	summary output file

7.D.5. TEST RUN ch3c2h4tr1

Potential : HF/6-31G*
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST, CVT
Tunneling : ZCT, SCT
Dual Level : no
Solvation : ESP, acetonitrile

This sample input is for an ESP dynamics calculation for the reaction $\text{CH}_3 + \text{C}_2\text{H}_4 \rightarrow \text{CH}_3\text{C}_2\text{H}_4$ in acetonitrile. The harmonic vibrational frequencies are evaluated using the SM5.42/HF/6-31G* model. The forward rate constant is calculated at 298 K. ZCT and SCT tunneling corrections are calculated with CVT. The reaction path is followed with the Page-McIver method with a gradient step size of 0.02 bohr within the range $-2.5 \text{ bohr} < s < 1.5 \text{ bohr}$.

GAMESSPLUSRATE I/O Files

GAMESSPLUSRATE Filenames

ch3c2h4tr1.70	esp.fu70	Input data for phase
ch3c2h4tr1.71	esp.fu71	Input data for reactant 1
ch3c2h4tr1.72	esp.fu72	Input data for reactant 2
ch3c2h4tr1.73	esp.fu73	Input data for product 1
ch3c2h4tr1.74	esp.fu74	Input data for product 2
ch3c2h4tr1.75	esp.fu75	Input data for the saddle point
ch3c2h4tr1.dat	poly.fu5	Input data for GAMESSPLUSRATE
ch3c2h4tr1.fu6	poly.fu6	Long output file
ch3c2h4tr1.fu15	poly.fu15	Summary output file

7.D.6. TEST RUN ch3chotr1

Potential : HF/6-31G*
Vibrations : harmonic cartesian
RODS : no
Dynamics : no
Tunneling : no
Dual Level : no
Solvation : SM5.43, ethanol

This sample input is for a geometry optimization of the molecule CH₃CHO (acetaldehyde) in ethanol. The harmonic vibrational frequencies are evaluated using the SM5.42/HF/6-31G* model. In this test run, the eigenvector following (EF) geometry optimizer in POLYRATE is used.

GAMESSPLUSRATE I/O Files

GAMESSPLUSRATE Filenames

ch3chotr1.70	esp.fu70	Input data for phase
ch3chotr1.dat	poly.fu5	Input data for GAMESSPLUSRATE
ch3chotr1.fu6	poly.fu6	Long output file

7.E. SOLUTION PHASE REACTIONS USING NES

7.E.1. TEST RUN ch3ohhtr1

Potential : HF/3-21G||AM1-SRP
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST, CVT
Tunneling : ZCT, SCT, LCT
Dual Level : no
Solvation : NES, water

This is a test run for the $\text{CH}_3\text{OH} + \text{H} \rightarrow \text{H}_2 + \text{CH}_2\text{OH}$ reaction at the SM5.42//HF/3-21G||AM1-SRP ($x = 0.276$) level. In this calculation, the reaction path is followed using the Page-McIver algorithm with the gradient step size of 0.02 bohr. The vibrational frequencies are evaluated at every 4th gradient step with redundant internal coordinates. The geometries of the stationary points are optimized using the EF algorithm. TST, CVT, CVT/ZCT, CVT/SCT, CVT/LCT, and CVT/mOMT dynamics calculations are carried at 298 K. The forward and backward symmetry numbers are set equal to 2, and the SCALEMASS variable is set equal to 1.0 amu. The solvent friction time is 10 fs with the diffusion coefficients determined using scheme G which is based on the free energy of solvation. See Ref. 4 in Section 3.B for further details.

GAMESSPLUSRATE I/O FILES

GAMESSPLUSRATE FILENAMES

TSRP	TSRP	NDDO-SRP file for AM1-SRP
CM2SRPH	CM2SRPH	CM2SRP file for HF/STO-3G
CM2SRPL	CM2SRPL	CM2SRP file for AM1-SRP
ch3ohhtr1.70	esp.fu70	Input data for GAMESSPLUSRATE
ch3ohhtr1.dat	poly.fu5	Input data for POLYRATE
ch3ohhtr1.fu6	poly.fu6	Long output file
ch3ohhtr1.fu15	poly.fu15	Summary output file

7.E.2. TEST RUN ch3cl2tr1

Potential : AM1
Vibrations : harmonic redundant curvilinear
RODS : no
Dynamics : TST, CVT, ICVT
Tunneling : ZCT, SCT
Dual Level : no
Solvation : NES, water

This sample input is for an NES dynamics calculation for the reaction $\text{CH}_3\text{Cl} + \text{Cl}^- \rightarrow \text{CH}_3\text{Cl} + \text{Cl}^-$ in water. The harmonic vibrational frequencies are evaluated using the SM5.42/AM1 model. The coordinates are scaled to a reduced mass of 20.58 amu. The forward rate constant is calculated at 298 K. The reaction path is followed with the Page-McIver algorithm with a step size of 0.01 bohr in the range of $-0.5 \text{ bohr} < s < 0.5 \text{ bohr}$. The solvent friction time is 10 fs, and the diffusion coefficients are determined using scheme A. Since this calculation uses solvation model SM5.42, the surface areas in scheme A are the exposed van der Waals (i.e., they are evaluated with a zero effective radius for the solvent) Scheme A and the linear-response nonequilibrium solvation model employed here are explained further in Ref. 4 of Section 3.B.

Both ICVT and CVT calculations are carried out with SCT tunneling corrections.

GAMESSPLUSRATE I/O Files	GAMESSPLUSRATE Filenames	
ch3cl2tr1.70	esp.fu70	Input data for phase
ch3cl2tr1.71	esp.fu71	Input data for reactant 1
ch3cl2tr1.72	esp.fu72	Input data for reactant 2
ch3cl2tr1.73	esp.fu73	Input data for product 1
ch3cl2tr1.74	esp.fu74	Input data for product 2
ch3cl2tr1.75	esp.fu75	Input data for the saddle point
ch3cl2tr1.dat	poly.fu5	Input data for GAMESSPLUSRATE
ch3cl2tr1.fu6	poly.fu6	Long output file
ch3cl2tr1.fu15	poly.fu15	Summary output file

8. COMPUTERS, OPERATING SYSTEMS FOR WHICH GAMESSPLUSRATE HAS BEEN TESTED

In each case we give the GAMESSPLUSRATE version number, then the computers and operating system on which GAMESSPLUSRATE was tested.

Version 8.1.1-8.4	IBM SP with PowerPC 604e processors (AIX 4.3.1)
Version 9.3	IBM SP with Power3 processors (AIX 5.1) IBM Regatta Power4 processors (AIX 5.1) SGI Altix with Madison 1.3GHz processors (Linux kernel 2.4.20) Netfinity Linux Cluster with Intel Pentium III processors (Linux kernel 2.4.20) Sun Blade 2000 wth UltraSparc-III processors (SunOS 5.8)

9. TEST RUN TIMINGS

Here are the serial timings for the test runs, in CPU seconds.
The following timings apply to version 9.3/P9.3-G4.2.

I. GAS PHASE

	IBM SP with 604e	Regatta	Altix	Netfinity	Sun
Test run ch5tr1	635.8	234.6	157.5	187.0	494.0
Test run ch5tr2	0.5	0.2	0.4	0.5	0.9
Test run ch4ohtr1	1.9	1.0	0.6	0.7	1.0
Test run ch4ohtr2	8.5	2.7	1.6	2.0	4.0
Test run ch4ohtr3	2.9	1.1	0.9	1.2	1.0
Test run ch4ohtr4	1348.7	696.6	574.5	668.7	1470.0
Test run ch4ohtr5	1600.3	667.5	477.1	625.7	1690.0
Test run nh3ohtr1	15.3	6.3	5.5	6.1	12.0
Test run ch3ohhtr1	4774.1	2307.5	1866.2	1846.0	5617.0
Test run ch3ohhtr2	6238.2	2595.5	1985.9	2206.2	6017.0
Test run ch3cl2tr1	1539.3	806.8	477.7	742.3	1659.0

II. SOLUTION PHASE USING SES

	IBM SP with 604e	Regatta	Altix	Netfinity	Sun
Test run ch3ohhtr1	17871.8	7446.6	5337.3	5808.3	18501.0
Test run ch3cl2tr1	3130.0	1329.3	1018.7	1129.9	3458.0
Test run ch3clnh3tr1	252.2	111.1	77.8	91.6	268.0
Test run ch3clnh3tr2	448.4	183.6	142.1	156.2	457.0
Test run 2hcoohtr1	5945.1	2294.9	2011.1	2111.3	6023.0

III. SOLUTION PHASE USING ESP

	IBM SP with 604e	Regatta	Altix	Netfinity	Sun
Test run ch3ohhtr1	15082.6	6284.4	6332.5	5970.2	16649.0
Test run ch3cl2tr1	1740.3	738.2	559.6	590.6	1843.0
Test run ch3clnh3tr1	158.5	66.5	70.6	58.1	140.0
Test run ch3clnh3tr2	319.9	156.4	142.4	140.8	306.0
Test run ch3c2h4tr1	44268.0	17588.2	14280.0	15829.4	40453.0
Test run ch3chotr1	3424.6	664.1	522.8	873.6	2545.0

IV. SOLUTION PHASE USING NES

	IBM SP with 604e	Regatta	Altix	Netfinity	Sun
Test run ch3ohhtr1	20310.9	12694.3	10365.6	12313.5	19041.5
Test run ch3cl2tr1	2180.6	782.1	580.4	696.1	1943.0

10. VERSION INFORMATION

10.A. GAMESSPLUSRATE VERSION NUMBERS

GAMESSPLUSRATE version numbers have the general form $x.y.z/Px'.y'.z'-Gx''$ where $x.y.z$ is the overall version number, $x'.y'.z'$ is the number of the version of POLYRATE on which it is based, and x'' is the number of the version of GAMESSPLUS on which it is based. The part before the solidus, i.e., $x.y.z$, is unique, i.e., it changes if the interface changes, the POLYRATE version changes, or the GAMESSPLUS version changes. Thus, the part after the solidus it is redundant information, and it may be omitted for simplicity if desired. In general, third-level revisions (z or z') are bug fixes, and first and second level revisions (x,y,x' , or y') are enhancements.

Since GAMESSPLUS versions 1.0 through 3.9 were called GAMESOL, GAMESSPLUSRATE based on earlier version of GAMESSPLUS was called GAMESOLRATE. The last version of GAMESOLRATE is GAMESOLRATE version 8.4/P8.4-G2.2.4. Beginning with version 9.3/P9.3-G4.2, the name of GAMESOLRATE is changed to GAMESSPLUSRATE.

Because the number before the virgule is unique (i.e., the part after the virgule is redundant information), it is equally acceptable to say, for example, GAMESSPLUSRATE-version 9.3 as to say GAMESSPLUSRATE version-9.3/P9.3-G4.2.

Note: we sometimes update one or more of the manuals without updating the version number. The manual version is determined by the date of its most recent change and is given on its first page. Any changes other than the manual(s) always involve a change in version number of the code.

10.B. REVISION HISTORY

Note that versions 8.1.1 through 8.4 were called GAMESOLRATE. Beginning with version 9.3, the name was changed to GAMESSPLUSRATE.

Version 8.1.1/P8.1.1-G2.2.3 (July 1999)

1. This was the first version of this program.

Version 8.1.2/P8.1.2-G2.2.3 (August 1999)

1. This is based on a new verison of POLYRATE.

Version 8.2/P8.2-G2.2.4 (September 1999)

1. Two new test runs NES/ch3cl2tr1 and NES/ch3ohhtr1 are added for nonequilibrium solvation calculations.
2. Two new test runs ESP/ch3clnh3tr2 and SES/ch3clnh3tr2 are added for testing the optimizer in POLYRATE.
3. Test runs SES/ch3cl2tr1 and ESP/ch3cl2tr1 are modified for calculating reaction rate constants at 298.0 K only.

4. Test runs SES/ch3clnh3tr1 and ESP/ch3clnh3tr1 are modified for evaluating TST reaction rates only.

Version 8.4/P8.4-G2.2.3 (December 1999)

1. This is based on a new version of POLYRATE.

Version 9.3/P9.3-G4.2 (June 2004)

1. This is based on a new version of POLYRATE and a new version of GAMESSPLUS.
2. The name of the program is changed from GAMESOLRATE to GAMESSPLUSRATE.
3. The 'shuttle' script for invoking a GAMESSPLUS calculation is modified to be compatible with the new version of GAMESSPLUS. The execution of the GAMESSPLUS executable is now carried out by utilizing the ddickick.x utility provided by GAMESS.
4. The installation scripts 'gamesolrate.ibm' and 'gamesolrate.sgi' under gmsplusrate/script/ have been changed to 'gmsplusrate.ibm' and 'gmsplusrate.sgi' respectively. Lines are added to these scripts to compile source files 'intbsv3.f' and 'rtpjac.f' in POLYRATE-version 9.3. These two files were not in the old version of POLYRATE, such as POLYRATE-version 8.4. Two new installation scripts 'gmsplusrate.altix' and 'gmsplusrate.nf' have been added for compiling GAMESSPLUSRATE on SGI Altix machines and Netfinity Linux Cluster machines, respectively.
5. Function calls of 'UPCASE' by subroutine READLN1 in 'hooks_gr.f' have been changed to 'UPCSE'. This function is provided by POLYRATE and used to convert a string from lower case to upper case. In POLYRATE-version 9.3, the appropriate function is 'UPCSE', not 'UPCASE'.
6. Subroutine GOUTENG in 'hooks_gr.f' has been modified. This subroutine is used to grab the energy from the GAMESSPLUS archive file Test.FChk. First, a bug in parsing the UHF energy is fixed, where the string label for locating the UHF energy is changed from 'E(UHF)' to 'E(UHF=)'. Second, the positions of the values of energies to be parsed are changed from the third word to the second word to be compatible with new version of GAMESS. Third, the string labels for locating AM1 and PM3 energies (R-AM1, U-AM1, RO-AM1, R-PM3, U-PM3, and RO-PM3) are added.
7. A bug in subroutine GREST in 'hooks_gr.f' has been fixed. For cases where GAMESSPLUSRATE writes a restart file, the dimensionality variable 'NEND' should be defined. In old versions of GAMESOLRATE, this variable was not defined unless GAMESOLRATE also read a restarting file. This bug caused GAMESOLRATE fail to write its restarting file in the correct format on some systems, such as the SGI Altix. To improve the portability of the code, we fixed this bug by explicitly assigning the correct dimensionality to 'NEND' for the problematic cases.

8. 'gamsolrate.inc' has been renamed to 'gmsplusrate.inc'. All INCLUDE statements of 'gamesolrate.inc' in files 'hook_gr.cray', 'hook_gr.f', 'hook_gr.wks', and 'hook_mdep.f' have been changed correspondingly to include 'gmsplusrate.inc'.
9. In the testrun ESP/ch3clnh3/ch3clnh3tr1, the GAMESSPLUS input file for geometry optimization of the saddle point (file ch3clnh3tr1.75) has been modified. The option 'IGEOM=1' has been added to the \$CM2 namelist to specify the use of the input geometry in ch3clnh3tr1.75 as the initial guess for the solution phase geometry optimization.

GAMESSPLUS provides two ways of optimizing geometries in the solution phase:

- (1) do a gas phase geometry optimization first (IGEOM=0, as a default) and use the gas-phase-optimized geometry as the initial geometry in the subsequent solution-phase geometry optimization.
- (2) use the structure given in the GAMESSPLUS input file as an initial geometry and do the solution-phase geometry optimization directly starting from this initial structure (IGEOM=1)

For the testrun mentioned above, the use of IGEOM=0 causes the saddle point being optimized to a wrong structure. Therefore we change to employ the second choice for this specific test run.

10. RODS has been removed from NES/ch3ohh/ch3ohhtr1.dat. In the current version of GAMESSPLUSRATE (actually POLYRATE), RODS or VRP is not compatible with BATH used for non-equilibrium solvation calculations. For carrying out RODS with NES, extra degrees of freedom associated with solvent coordinates must be included in the optimization of the reoriented dividing surface. This has not been implemented in the current version of POLYRATE.
11. A test run involving liquid phase optimization with SM5.43 has been added (testrun/ESP/ch3cho).
12. The code is tested on more platforms.

11. ACKNOWLEDGMENTS

The development of GAMESSPLUSRATE has been supported in part by grants from the National Science Foundation (liquid-phase dynamics) and U. S. Department of Energy, Office of Basic Energy Sciences (gas-phase dynamics).