

GAUSSRATE Quickstart Guide

February 27, 2007

Benjamin A. Ellingson and Donald G. Truhlar

Based on:

GAUSSRATE 9.5

José C. Corchado, Yao-Yuan Chuang, Elena L. Coitiño, Benjamin A. Ellingson, and Donald G. Truhlar

Department of Chemistry and Supercomputer Institute,
University of Minnesota, Minneapolis, MN 55455-0431, USA

GAUSSRATE provides an interface between two other programs:

POLYRATE 9.5, by

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

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

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

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

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

and

GAUSSIAN 03, by M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery, Jr., T. Vreven, K. N. Kudin,

J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, and J. A. Pople, Gaussian, Inc., Pittsburgh PA, 2003.

or

GAUSSIAN 98, by M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, V. G. Zakrzewski, J. A. Montgomery, R. E. Stratmann, J. C. Burant, S. Dapprich, J. M. Millam, A. D. Daniels, K. N. Kudin, M. C. Strain, O. Farkas, J. Tomasi, V. Barone, M. Cossi, R. Cammi, B. Mennucci, C. Pomelli, C. Adamo, S. Clifford, J. Ochterski, G. A. Petersson, P. Y. Ayala, Q. Cui, K. Morokuma, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. Cioslowski, J. V. Ortiz, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. Gomperts, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, C. Gonzalez, M. Challacombe, P. M. W. Gill, B. G. Johnson, W. Chen, M. W. Wong, J. L. Andres, M. Head-Gordon, E. S. Replogle and J. A. Pople, Gaussian, Inc., Pittsburgh PA, 1999.

or

GAUSSIAN 94, by M. J. Frisch, G. W. Trucks, H. B. Schlegel, P. M. W. Gill, B. G. Johnson, M. A. Robb, J. R. Cheeseman, T. Keith, G. A. Petersson, J. A. Montgomery, K. Raghavachari, M. A. Al-Laham, V. G. Zakrzewski, J. V. Ortiz, J. B. Foresman, J. Cioslowski, B. B. Stefanov, A. Nanayakkara, M. Challacombe, C. Y. Peng, P. Y. Ayala, W. Chen, M. W. Wong, J. L. Andres, E. S. Replogle, R. Gomperts, R. L. Martin, D. J. Fox, J. S. Binkley, D. J. Defrees, J. Baker, J. P. Stewart, M. Head-Gordon, C. Gonzalez, and J. A. Pople, Gaussian, Inc., Pittsburgh PA, 1995.

GAUSSRATE program version: 9.5/P9.5-G03G98G94

GAUSSRATE program version date: January 10, 2007

Quickstart version date: February 27, 2007

Copyright 2007

1. Introduction

This guide is intended to give the user a starting knowledge of how to install and use GAUSSRATE. Common difficulties will be highlighted, and ways to overcome them will be suggested.

2. Installation

GAUSSRATE is an interface program between POLYRATE and *Gaussian 03*. (Older versions of *Gaussian* are compatible, but this guide will focus on the newest version.) *Gaussian 03* is a separate commercial product that must be installed first. For *Gaussian 03*, follow their installation procedure if it is not already available on your system. Both POLYRATE and GAUSSRATE can be acquired from the website <http://comp.chem.umn.edu/software>. Fill out and return the license forms and you will be given a password to download the programs; there is no fee for these two programs. The full instructions for installing POLYRATE are in the POLYRATE manual, but here we give quickstart instructions.

Once the POLYRATE and GAUSSRATE programs have been acquired, install POLYRATE first. Transfer the downloaded POLYRATE file to the desired installation directory. (The home directory is acceptable, because a subdirectory will automatically be created.) If the file ends in *.Z*, use the uncompress command (i.e., ‘uncompress polyrate8.7.2.tar.Z’). If the file ends in *.gz*, use the gunzip command (i.e., ‘gunzip polyrate9.5.tar.gz’). Now the file may be untarred:

```
[~] % tar -xvf polyrate9.5.tar
```

Now go to the POLYRATE directory and run the ‘configure’ script, where you may simply hit ‘Enter’ at the prompts:

```
[~] % cd polyrate9.5/  
[~/polyrate9.5] % ./configure
```

```
Welcome to POLYRATE 9.5!
```

```
Do you want POLYRATE 9.5 to be your  
new default version of POLYRATE? [ yes ]:
```

```
... processor is powerpc  
... OS is AIX  
....found XLF F77 compiler :/usr/bin/fort77  
....found MPXLF MPI compiler :/usr/bin/mpxlf  
....found C compiler :/usr/vacpp/bin/cc
```

```
Do the options chosen above look OK? [ yes ]:  
... installing
```

```
.... creating Makefile
.... done with Makefile

---- POLYRATE INSTALLATION COMPLETE ----

Have a nice day! :)
```

POLYRATE is now installed, so it is time to install GAUSSRATE. Transfer the downloaded GAUSSRATE file to the newly created POLYRATE directory and gunzip and untar it.

```
[~/polyrate9.5] % mv ../gaussrate9.5.tar.gz .
[~/polyrate9.5] % gunzip gaussrate9.5.tar.gz
[~/polyrate9.5] % tar -xvf gaussrate9.4.tar
```

Now go to the GAUSSRATE directory and edit the 'shuttle' script:

```
[~/polyrate9.5] % cd gaussrate
[~/polyrate9.5/gaussrate] % vi shuttle
```

The 'shuttle' script needs to be told where the *Gaussian 03* executable file is located. Edit the following line in the script to do this:

```
set gausspath=/usr/local/g03/g03.d01/g03
```

The 'configure' script will copy the 'shuttle' script to the polyrate/exe directory. GAUSSRATE uses the file located there, so any future changes to the script need to be made in that location.

The configure script will install and compile the program. This is done by copying the GAUSSRATE source files to the ~/polyrate9.5/src directory and then adding a GAUSSRATE section to the Makefile located in that same directory. Then the configure script uses the gmake command compile the program. Changes to the FORTRAN compiler can be made in the first line of the Makefile, if desired. Now run the configure script.

```
[~/polyrate9.5/gaussrate] % ./configure
```

Once this is complete, you may proceed to the test runs. Running the full test suite is strongly encouraged. Running a single test run is a good starting point:

```
[~/polyrate9.5/gaussrate] % cd testrun/ch4oh/
[~/polyrate9.5/gaussrate/testrun/ch4oh] % ./ch4ohtr1.jc
```

The last line of the *.fu6 file will contain 'END OF POLYRATE LONG OUTPUT' if the calculation completes. To test the whole test suite, run the 'run_all.jc' script in the testrun directory. It may take several hours.

3. Advice on applications

When making applications with GAUSSRATE, one must check that the results are reasonable and do not suffer from numerical problems, instabilities, or errors. It is good to check the results in a particular order, starting with the results that use the least information about the potential energy surface and reaction path, and proceeding on to the more complicated calculations. Thus it is recommended that one check the results in this order:

- reactant properties (i.e., geometries and frequencies)
- product properties and energy of reaction
- saddle point properties, including barrier height and magnitude of imaginary frequency
- conventional transition state theory without tunneling (TST)
- CVT with zero-curvature tunneling (CCT/ZCT)
- CVT with small-curvature tunneling (CVT/SCT)
- CVT with LCT tunneling limited to ground vibrational state (CVT/(LCT(0))
- CVT with full LCT and muOMT tunneling (CVT/LCT and CVT/muOMT)

If a problem or unreasonable results or numerical instability is encountered at any of these levels, it is smart to resolve this before proceeding to consider the next level.

If there is a problem with curvilinear coordinates or torsional anharmonicity, it may be useful to first converge the calculations, if possible, with harmonic rectilinear frequencies. However, since rectilinear frequencies are more likely than curvilinear frequencies to lead to spurious imaginary frequencies, it is sometimes better to use curvilinear coordinates right from the outset. Published results should usually be based on curvilinear coordinates since they correspond to a more reasonable physical model for the transition state dividing surface.

Note that some reactions do not require large-curvature tunneling, and it is sufficient to stop with CVT/SCT. One problem though is that it is often hard to know whether LCT tunneling is important for a given case without at least approximately converging it.

Vibrational partition functions are very sensitive to the values of the frequencies. Therefore, the following steps should be taken to ensure that the most precise frequencies are calculated for a given level of theory. First, when using a DFT method, the `ultrafine` grid should be used because it yields more precise frequencies than the default grid. Second, optimizations should occur at the `vtight` level. The default convergence criteria allows a significant non-zero gradient to contaminate the calculated frequencies. Finally, nearly every electronic structure methods systematically overestimates the frequencies when using the harmonic oscillator approximation. The frequencies should be scaled by an appropriate number using the `FREQSCALE` option. A database of frequency scaling factors is maintained at http://comp.chem.umn.edu/database/freq_scale.htm.

4. A quick overview of how the program works

The POLYRATE program takes information about a potential energy surface and uses it to calculate reaction rates and/or related properties such as saddle point geometries and vibrational

frequencies. For an introductory article on variational transition state theory (VTST), please see the "Transition State Theory" book chapter, freely available in pdf form at <http://t1.chem.umn.edu/Truhlar/bookchap.htm> (listed as #58). The standalone POLYRATE program requires a potential energy surface (PES), usually an analytic function, for the system to be compiled and linked with the program.

GAUSSRATE is a program that allows *Gaussian 03* to calculate the required information about the PES "on the fly." This is called direct dynamics.¹⁻⁹ Rather than compiling and linking an analytic PES with POLYRATE, the GAUSSRATE program is compiled and linked with POLYRATE. When POLYRATE needs information about the PES, in particular energies, gradients, or Hessians at specific geometries, it calls the GAUSSRATE subroutines, which run *Gaussian 03* calculations and pass the information back to POLYRATE. GAUSSRATE can also be used to optimize stationary points using either the POLYRATE optimizer or the *Gaussian 03* ones, which are more highly recommended.

This basic understanding of the programs will help with troubleshooting and creating your own input files. The main input file, *.dat, is read in by the POLYRATE routines. The keywords found in this file are in the POLYRATE manual. A few keywords related to how the PES is obtained can also be found in the GAUSSRATE manual. The *.70 file contains *Gaussian 03* related information which is read in with GAUSSRATE subroutines. The *.71 through *.75 files are *Gaussian 03* input files which are called to obtain geometries. Information specifically related to *Gaussian 03* should be obtained from the *Gaussian 03* manual, available online at http://www.gaussian.com/g_ur/g03mantop.htm.

5. The input files

Below are the input files used for the ch4ohtr1 test run, with comments added on the right. Specific keywords in the ch4ohtr1.dat file can be found in the POLYRATE manual, and keywords in the ch4ohtr1.70 file are found in the GAUSSRATE manual. The ch4ohtr1.71 file is a *Gaussian 03* input file that optimizes the geometry of the first reactant, and any information about this file should be obtained from the *Gaussian 03* manual; an example of this file is given later in this section.

The ch4ohtr1 test run optimizes the first reactant and then stops.

The ch4ohtr1.dat file:

```
*General
```

```
TITLE
CH4 + OH -> CH3 + H2O           Title of the calculation
Calculation of first reactant (CH4)
END

ATOMS                             The atoms are assigned
  1      H                         numbers here.
```

```
2 C
3 H
4 H
5 H
6 O
7 H
END
```

```
NOSUPERMOL
```

```
*OPTIMIZATION
```

```
OPTMIN OHOOK
OPTTS OHOOK
```

```
*SECOND
```

```
HESSCAL HHOOK
```

```
*REACT1
```

```
INITGEO HOOKS
```

```
GEOM
```

```
1
2
3
4
5
END
```

```
SPECIES NONLINRP
```

```
# end of react1 section
```

The ch4ohtr1.70 file:

```
*GRGENERAL
```

```
NOGRRESTART
```

Required keyword

This tells POLYRATE to call GAUSSRATE for optimization.

Use GAUSSRATE for Hessians.

Tells POLYRATE to “call” GAUSSRATE for the geometry optimization. GAUSSRATE obtains the optimized geometry by running the *.71 file with *Gaussian 03*.

The atoms found in the first reactant, note that the *.71 file uses the same order, and this consistency is required.

Tells POLYRATE that it is a nonlinear reactant/product

Use ‘#’ to comment lines.

The restart option is not

turned on. It should usually be turned on when calculating an MEP.

*GRCOMMON

GRENER

#PM3 UNITS=AU FCHK NOSYMM

Options for energy calc.

END

GRFIRST

#PM3 FORCE UNITS=AU FCHK NOSYMM

Options for gradient calc.

END

GRSEC

Options for Hessian calc.

#PM3 FREQ=NORAMAN UNITS=AU FCHK NOSYMM

END

These *Gaussian 03* keywords are always required for energy, gradient, and Hessian calculations: “UNITS=AU FCHK NOSYMM”. Even though your *Gaussian 03* input file may be in angstroms, atomic units are required by the POLYRATE subroutines, and UNITS=AU tells *Gaussian 03* to use atomic units for communication with GAUSSRATE. The FCHK keyword tells *Gaussian 03* to write the file from which GAUSSRATE extracts the data from. The NOSYMM keyword forces *Gaussian 03* to write out the information for the original orientation of the geometry. This issue is discussed further in Section 7.4.3 of the GAUSSRATE manual.

Also required in this section is the level of theory, which is PM3 in this example, but any other level may also be used. We recommend the BB1K or MPWB1K levels of density functional theory. See <http://comp.chem.umn.edu/info/DFT.htm> for further information about these density functionals. An energy calculation is the default in *Gaussian 03*, which is why there are not any extra keywords in the GRENER section. The GRFIRST section requires the FORCE keyword, which tells *Gaussian 03* to do a gradient calculation, and the GRSEC section requires the FREQ keyword, which tells *Gaussian 03* to do a Hessian calculation.

The ch4ohtr1.71 file:

#PM3 OPT FCHK

Gaussian 03 keywords

Methane

Arbitrary title

0 1

Charge, Multiplicity

H

Geometry

C 1 r1

H 2 r1 1 a1

H 2 r1 1 a1 3 d1

H 2 r1 1 a1 3 -d1


```
r1 1.087
a1 109.471
d1 120.
```

This is simply a standard *Gaussian 03* input file. Note the FCHK keyword, which is required. If this keyword is missing, everything will appear to run normally, but then the calculation will die because the file that GAUSSRATE needs to read is never created. Also note that the order of the atoms corresponds exactly to the order defined in the ch4ohtr1.dat file. It is important to keep the atom ordering in the *.dat file and the Gaussian files consistent.

The job control script, *.jc, is the most cryptic looking file. It is a C shell script. The first comment line in a C shell script has a standard form, and it should not be changed. Any other line beginning with a # is a comment. Additional comments have been added here on the right where changes may be needed. The ch4ohtr1.jc file:

```
#!/bin/csh -f
#
# Usage ch4ohtr1.jc [working_directory]
#
set name = ch4ohtr1                               Name used for files is here.
if ($#argv != 0) then
    set actdir = `pwd`
    set wrkdir = $argv[1]
else
    set actdir = `pwd`
    set wrkdir = $actdir/$name
check:
    if (-e $wrkdir) then
        set tmp = $wrkdir/$name
        set wrkdir = $tmp
        goto check
    endif
    mkdir $wrkdir
endif
#
cd $wrkdir
#
# Set the name of the executable
#
set polydir = `cat ~/.poly_path`
set exedir  = $polydir/exe
set exe     = $exedir/gaussrate.exe
#
# Delete the fu# files left previously if any
#
rm poly.fu* >& /dev/null
#
```

```

# Copy the data files to POLYRATE filenames.
#
if (-e esp.fu81) then
/bin/rm esp.fu81
endif
if (-e shuttle) goto input
cp $exedir/shuttle $wrkdir/shuttle
input:
cp $actdir/$name.dat poly.fu5           Copies files to standard
cp $actdir/$name.70  esp.fu70          names to be read in by
cp $actdir/$name.71  esp.fu71          the program; see below for
#                                       discussions of directory
# Write system message to .time file    management.
#
(time $exe) >& $actdir/$name.time
echo " " >> $actdir/$name.time
#
# Copy the POLYRATE output files to appropriate filenames
#
rm poly.fu5                             Files are deleted or moved
rm esp.fu70                             back to the original directory.
rm esp.fu71
mv poly.fu6  $actdir/$name.fu6
mv esp.fu61  $actdir/$name.61
if (-e $actdir/GRTEMPFILES) goto exist
mkdir $actdir/GRTEMPFILES
exist:
mv * $actdir/GRTEMPFILES
cd $actdir
rm -rf $wrkdir
#
# End of script
exit

```

There are only a few places in the script that would ever need to be modified. The first is the 'set name' line, which is where the name used for the files is defined and stored with the variable \$name. In this example, the files being used are named ch4ohtr1.dat, ch4ohtr1.70, and ch4ohtr1.71, so the name is set to ch4ohtr1.

The second place that may need to be changed is after the 'input:' line, which copies the files from the active directory to their standard names in the working directory. By default, the active directory is the directory in which the *.jc file is being run, and the working directory is a newly created subdirectory entitled \$name. In this example, a directory called ch4ohtr1 will be created, and this is where the actual GAUSSRATE calculation will take place. Any additional files being used, such as the *.72 file for the second reactant, would also need to be added to this location of the script for copying.

The final place that may require change is the section towards the bottom that copies and removes files. For example, if you would like to keep the poly.fu15 file (which contains the rate constant summary), this line can be added: `mv poly.fu15 $actdir/$name.fu15`. This command will rename the summary file using the `$name` variable, and copy the file from the temporary directory to the active directory (which is the directory from which the program was run.)

6. A standard rate constant calculation

The ch4oh test set, found in the `polyrate9.5/gaussrate/testrun/ch4oh/` directory, is a good example of how to go about a rate constant calculation. The first test run, which has been used as the example thus far, calculates the first reactant. This calculation produces the `ch4ohtr1.61` file, which contains the optimized geometry, energy, and frequencies of the first reactant.

The second test run, `ch4ohtr2`, is a continuation of the first. In this test run, the information from the `ch4ohtr1.61` file is copied to the `*REACT1` section of the `ch4ohtr2.dat` file, and the keyword `'STATUS 6'` is added in the `*REACT1` section to tell the program not to recalculate the information for the first reactant. The `ch4ohtr2` test run calculates the information for the second reactant and both products.

The third test run is a TST rate constant calculation. Once again, the previously calculated data is copied from the `*.61` file to the `*.dat` file, and the `STATUS` is set to 6. The transition state is denoted by the `*START` section. It is setup in a similar manner to the reactants and products, with the `SPECIES` now being set to `NONLINTS`, rather than `NONLINRP`. There is also a `*RATE` section, which contains the options for the rate constant calculation. In this test run, the `FORWARDK` keyword declares that a forward rate constant is to be calculated. The `SIGMAF` keyword denotes the symmetry number for the forward reaction. The `NOCVT` keyword tells the program not to variationally optimize the location of the dividing surface. This is turned off because a minimum energy path (MEP) is not being calculated at this time. And finally, the temperatures that the rate constant is to be calculated at are listed using the `TEMP` keyword.

The fourth test run is a CVT rate constant calculation. There are many options in this test run that go beyond a quickstart guide, so just the additional steps needed for CVT and tunneling will be highlighted. Once again, information is copied from the `*.61` file to the `*.dat` file, this time for the `*START` section. In the `ch4ohtr4` test run, the geometry, energy, and Hessian are copied over, and the `STATUS` is set to 4. With this option, the frequencies and vibrational normal mode eigenvectors will be calculated from the Hessian. Alternatively, the geometry, energy, frequencies, Hessian, and eigenvectors can be copied from the `*.61` file and the `STATUS` set to 6; this option does not require any recalculation for the transition state. The MEP is calculated in this test run. Here are the standard options that should be listed to calculate an MEP:

*PATH

SCALEMASS	1.00	Mass used for mass-scaled coordinates
SSTEP	0.002	Step size in bohr; 0.002 – 0.005 is standard
INH	10	How many steps per Hessian (The recommend number is 9, but this example uses 10)
SRANGE		
SLP	1.00	Upper bound on s , the MEP coordinate
SLM	-2.00	Lower bound on s
END		
RPM	ESD	Reaction path method, steepest descent

Internal coordinates with the CURV3 option are recommended. They are discussed in the next section. The reaction coordinate is called s , where positive s leads toward products and negative s leads towards reactants. This MEP calculation will start at $s = 0$, which is the saddle point, and make steps of 0.002 bohr with Hessian calculations every 0.020 bohr until $s = 1.0$ bohr. Then a step is made from the saddle point in the opposite direction, and the path is calculated to $s = -2.0$ bohr in the same manner.

Also, tunneling can be defined in the *TUNNEL section. The SCT option is recommended for initial work and often is also the final method for publication. It is the best method that does not require any additional calculations by *Gaussian 03*. A description of tunneling methods accompanied by references is found in Section 6.D. of the POLYRATE manual and in the “Transition State Theory” book chapter.

Finally, the *RATE section now says CVT instead of NOCVT. The CVT calculation will use the MEP and find the optimal dividing surface.

7. Imaginary frequencies along the MEP

Imaginary frequencies are a common problem. Imaginary frequencies can either result from a bifurcating reaction path or a bad coordinate system; the latter is much more common. Imaginary frequencies resulting from a nonphysical coordinate system are considered spurious. The mostly likely cause for spurious imaginary frequencies is the use of Cartesian coordinates, which is the default. User-defined redundant internal coordinates are much better for calculating frequencies along the MEP. This can be done with the CURV3 option in the PATH section of the *.dat file. This following example comes from the ch4ohtr4 test run. First the CURV3 option is chosen using the COORD keyword, then the coordinates are defined with the INTDEF keyword.

```
COORD  CURV3

INTDEF
  2-1      2-3      2-4      2-5      3-6      6-7
```

```

1-2-3      1-2-4      1-2-5      3-2-4      3-2-5      4-2-5
2=3=6      3-6-7
7-6-2-1    7-6-2-4    7-6-2-5
END

```

The first line defines the stretches, the second and third lines define the bends, and the fourth line defines the torsions. The order is not important to the program. The notation 2=3=6 refers to a doubly degenerate linear bend, which is why it uses an equal sign.

The calculation of frequencies on the MEP is coordinate dependent, except at the saddle point. Curvilinear internal coordinates are more natural coordinates for describing molecule vibrations, especially vibrations associated with bending and torsions. The use of a complete set of curvilinear internal coordinates will likely eliminate any imaginary frequencies. Although the use of a physical coordinate system usually eliminates the spurious imaginary frequencies, it does not always eliminate nonphysical real frequencies.¹⁰ If nonphysical frequencies remain, the next option is to try the `RODS` keyword. `RODS` stands to 'reorientation of the dividing surface'. When this option is used, at every point where a Hessian is calculated, the surface is reoriented to maximize the free energy. (The default is for the surface to be normal to the gradient.)

The previous two options (i.e., curvilinear internal coordinate and `RODS`) can be used with an MEP that has already been calculated. The next option for calculating improved frequencies, `VRP (ESD)`, is a reaction path method, so restart data from a run with different reaction path methods cannot be used. The `VRP(ESD)` algorithm uses the `RODS` algorithm to not only reorient the surface, but also to change the direction of the next step. This helps the reaction coordinate readjust back to a physical path kly if it is off track.

8. Solvation

The `SMx` method is the recommended solvation model for applications to liquid-phase reactions, and it is available in the `SMXGAUSS` and `GAMESSPLUS` programs. Additionally, there is a `GAMESSPLUSRATE` program to interface `GAMESSPLUS` with `POLYRATE`. Please see <http://comp.chem.umn.edu/solvation/> for more information.

The `PCM` solvation model included in the *Gaussian 03* program can be used with `GAUSSRATE`. However, the method will typically fail for generalized transition states when using the default values. A working `PCM` calculation has been shown in test run `h3tr2`. A choice that has been found to improve convergence when using `PCM` for transition state theory is using the 'pauling radii'. First, the option 'read' must appear in the 'SCRF' as shown here:

```

#hf/6-31G(d,p) opt=(calcfc,ts) nosymm fchk
SCRF=(PCM,read,Solvent=water)

```

Secondly, there must be a blank line after the geometry, with this option following the blank line:

```

radii=pauling

```

While this option can be added simply for the initial *Gaussian 03* files (*.71-*.75), a trick must be employed to use this option with files that GAUSSRATE creates. GAUSSRATE reads in options for the *Gaussian 03* input files from the *.70 file. Lines needed at the end of a *Gaussian 03* input file are put in the GRBASIS section. The GRBASIS keyword was originally used for reading in basis sets, but it can be used for anything that needs to go at the bottom. The *.70 file needs to have a section like this:

```
GRBASIS
  radii=pauling
END
```

9. User-defined basis sets

If there is a user-defined basis set that needs to be inputted, it needs to be placed in the GRBASIS section of the *.70 file, as well as at the end of the initial *Gaussian 03* files (*.71-*.75). Rather than pasting the entire basis set in the file, a link to the absolute path name of a file can be given. For example,

```
GRBASIS
@/home/smpa/elling/basis/mg3s
END
```

where the mg3s basis set is in a text file located at /home/smpa/elling/basis/mg3s. Instructions for user-defined basis sets in *Gaussian 03* files are found at www.gaussian.com/g_ur/k_gen.htm.

10. Linear Molecules

The H3 test runs show that GAUSSRATE can be used for linear systems. However, problems may arise with *Gaussian 03* due to symmetry-related issues. The keyword 'nosymm' must be supplied, otherwise *Gaussian 03* automatically changes the input to a standard orientation to run calculations. When using the standard orientation, the gradient and Hessian components are calculated and printed out for this orientation, rather than the original orientation. The original orientation must be used when calculating a reaction path, thus the 'nosymm' keyword is required.

Numerical problems in *Gaussian 03* are possible due to a linear molecule being calculated without symmetry. In the H3 test runs, only TST is calculated because of numerical errors that occur if one tries to calculate a reaction path. GAUSSRATE itself does not have any problems treating linear systems, so any system that *Gaussian 03* can calculate with the 'nosymm' keyword can be calculated with GAUSSRATE.

11. "Error reading Gaussian 03/98/94 checkpoint file"

A common error to receive in the *.fu6 file when a GAUSSRATE calculation dies is "Error reading Gaussian 03/98/94 checkpoint file". This error is given anytime that GAUSSRATE is unable to read the formatted checkpoint file, Test.FChk. More often than not, the program cannot read this file because the *Gaussian 03* calculation died and never created it.

The first step is to look at the esp.fu82 file in the GRTEMPFILES subdirectory. This is the generic output file for all *Gaussian 03* calculations. If the esp.fu82 file does not have "Normal termination of Gaussian 03" on the last line, then the *Gaussian 03* calculation associated with this output file is the source of the problem. The esp.fu82 file will likely show what went wrong. If the file ends suddenly without any sort of error message, it is probable that the job was killed by the computer that it was running on, possibly because it overran the requested processor time or storage limits.

Another place to look when trying to track down an error is the *.time file. This file contains any information that the system wrote to the standard output during the run. System messages relating to memory or drive space errors will be located in this file.

If these files do not give a clear picture of the problem, it may be helpful to edit the *.jc file. First make a copy of the file, and then delete everything after the "# Copy the POLYRATE output files to appropriate filenames" line. Now run the calculation again, and when it dies nothing will be renamed or moved, so all of the files will remain unchanged from the moment that the calculation died. Full inspection of the files should show why the calculation did not complete correctly.

12. Concluding Remarks

Once the initial GAUSSRATE calculations are running correctly, most information that the user will be interested in will be found in the POLYRATE manual. The POLYRATE manual covers all of the options related to dynamics, such as reaction path following method, reaction path extrapolation methods, tunneling methods, torsion methods, frequency scaling, output options for extra information, and so on. This manual is available in pdf form at <http://comp.chem.umn.edu/polyrate/>.

References

1. "Ab Initio Reaction Paths and Direct Dynamics Calculations," K. K. Baldrige, M. S. Gordon, R. Steckler, and D. G. Truhlar, *Journal of Physical Chemistry* **93**, 5107-5119 (1989).
2. "Direct Dynamics Calculations with Neglect of Diatomic Differential Overlap Molecular Orbital Theory with Specific Reaction Parameters," A. González-Lafont, T. N. Truong, and D. G. Truhlar, *Journal of Physical Chemistry* **95**, 4618-4627 (1991).
3. "MORATE: A Program for Direct Dynamics Calculations of Chemical Reaction Rates by Semiempirical Molecular Orbital Theory," T. N. Truong, D.-h. Lu, G. C. Lynch, Y.-P. Liu,

- V. S. Melissas, J. J. P. Stewart, R. Steckler, B. C. Garrett, A. D. Isaacson, A. González-Lafont, S. N. Rai, G. C. Hancock, T. Joseph, and D. G. Truhlar, *Computer Physics Communications* **75**, 143-159 (1993).
4. "Direct Dynamics Calculation of the Kinetic Isotope Effect for an Organic Hydrogen-Transfer Reaction, Including Corner-Cutting Tunneling in 21 Dimensions," Y.-P. Liu, D.-h. Lu, A. González-Lafont, D. G. Truhlar, and B. C. Garrett, *Journal of the American Chemical Society* **115**, 7806-7817 (1993).
 5. "Direct Dynamics Method for the Calculation of Reaction Rates," D. G. Truhlar, in *The Reaction Path in Chemistry: Current Approaches and Perspectives*, edited by D. Heidrich (Kluwer, Dordrecht, 1995), pp. 229-255. [Understanding Chem. React. **16**, 229-255 (1995).]
 6. "Parameterization of NDDO Wavefunctions using Genetic Algorithms: An Evolutionary Approach to Parameterizing Potential Energy Surfaces and Direct Dynamics Calculations for Organic Reactions," I. Rossi and D. G. Truhlar, *Chemical Physics Letters* **233**, 231-236 (1995).
 7. "Transition State Resonances in the Reaction $\text{Cl} + \text{H}_2 \rightarrow \text{HCl} + \text{H}$," J. Srinivasan, T. C. Allison, D. W. Schwenke, and D. G. Truhlar, *Journal of Physical Chemistry A* **103**, 1487-1503 (1999).
 8. "Parameterized Direct Dynamics Study of Rate Constants of H with CH_4 from 250 to 2400 K," J. Pu and D. G. Truhlar, *Journal of Chemical Physics* **116**, 1468-1478 (2002).
 9. "Variational Transition State Theory with Multidimensional Tunneling," A. Fernandez-Ramos, B. A. Ellingson, B. C. Garrett, and D. G. Truhlar, in *Reviews in Computational Chemistry*, Vol. 23, edited by K. B. Lipkowitz and T. R. Cundari (Wiley-VCH, Hoboken, NJ, 2007), pp. 125-232.
 10. "Variational Transition State Theory and Tunneling Calculations with Reorientation of the Generalized Transition States for Methyl Cation Transfer," A. González-Lafont, J. Villà, J. M. Lluch, J. Bertrán, and D. G. Truhlar, *Journal of Physical Chemistry* **102**, 3420-3428 (1998).