Abstract:
GAUSSRATE is a set of FORTRAN subroutines and Unix scripts for interfacing the POLYRATE and GAUSSIAN 09 (or 03 or 98 or 94) computer programs for the purpose of carrying out direct dynamics calculations of gas-phase chemical reaction rates of polyatomic species (and also atoms and diatoms as special cases) using the electronic structure methods available in GAUSSIAN 09 (or 03 or 98 or 94) 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 for tunneling and nonclassical reflection. Rate constants may be calculated by any of the methods available in the POLYRATE program for both tight and loose transition states.

For tight transition states, the methods are canonical, improved canonical, microcanonical variational or conventional transition state theory with multidimensional semiclassical approximations for tunneling and nonclassical reflection. Bimolecular and unimolecular reactions are included. Both single-level and dual-level calculations may be carried out. In single-level mode, optimized geometries, potential energies, gradients, and Hessians can be calculated by any of the ab initio or semiempirical methods in the GAUSSIAN 09 (or 03 or 98 or 94) packages. In dual-level mode, the lower-level data is calculated by GAUSSIAN 09 (or 03 or 98 or 94), and the higher-level data is read in from an external file.

For loose transition states rate constants may be calculated for canonical or microcanonical ensembles or energy and total angular momentum resolved.
microcanonical ensembles. Multifaceted dividing surfaces based on a variable reaction coordinate are used. For loose transition states, only bimolecular association reactions are included, and single-level calculations may be carried out.
GAUSSRATE–version 2015 provides an interface between two other programs:


and


or


or

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

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.
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.
Table of Contents

1. USER AGREEMENT ................................................................. 7
2. REQUIRED REFERENCES .......................................................... 8
3. INTRODUCTION ........................................................................ 11
4. DISTRIBUTION ........................................................................ 13
5. INSTALLATION .......................................................................... 16
6. PROGRAM DESCRIPTION ............................................................ 17
   6.1. Description of GAUSSRATE subprograms ............................... 17
   6.2. Restarting a calculation ....................................................... 19
   6.3. Parallel capabilities ........................................................... 22
7. DESCRIPTION OF INPUT FILES .................................................. 24
   7.1. File usage ....................................................................... 24
   7.2. Description of fu5 unit ....................................................... 26
   7.3. Description of esp.fu71 through esp.fu78 input files ............... 31
   7.4. Description of esp.fu70 input file ........................................ 32
   7.4.1. The GRGENERAL section .............................................. 33
   7.4.2. The GRCOMMON section ............................................. 34
   7.4.3. The GRREACT1, GRREACT2, GRPROD1, GRPROD2, GRWELLR,
          GRWELLP, GRSTART, and GRPATH sections ...................... 36
8. SPEED UP THE DIRECT DYNAMICS CALCULATIONS USING GAUSSRATE
   .............................................................................................. 39
9. TESTING GAUSSRATE ................................................................. 42
   9.1. Tests performed ................................................................ 42
   9.2. Test suite ....................................................................... 43
   9.2.1. Test run ch5tr1 ............................................................ 44
   9.2.2. Test run ch5tr2 ............................................................ 45
   9.2.3. Test run oh3tr1 ............................................................ 46
   9.2.4. Test run oh3tr2 ............................................................ 47
   9.2.5. Test run oh3tr3 ............................................................ 48
   9.2.6. Test run ch4ohtr1 .......................................................... 49
   9.2.7. Test run ch4ohtr2 .......................................................... 50
   9.2.8. Test run ch4ohtr3 .......................................................... 51
   9.2.9. Test run ch4ohtr4 .......................................................... 52
   9.2.10. Test run ch4ohtr5 .......................................................... 54
   9.2.11. Test run ch4ohtr6 .......................................................... 56
   9.2.12. Test run nh3ohtr1 .......................................................... 57
   9.2.13. Test run nh3ohtr2 .......................................................... 58
   9.2.14. Test run h3tr1 ............................................................... 59
   9.2.15. Test run h3tr2 ............................................................... 60
   9.2.16. Test run c2h6tr1 ............................................................ 61
10. COMPUTERS, OPERATING SYSTEMS, AND GAUSSIAN VERSIONS ON
    WHICH THE CODE HAS BEEN DEVELOPED AND TESTED ............... 62
11. VERSION INFORMATION ............................................................ 65
   11.1. GAUSSRATE revision numbers ........................................... 65
   11.2. GAUSSRATE revision history ............................................. 66
12. ACKNOWLEDGMENTS ........................................................................... 74
1. USER AGREEMENT

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

A. 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 GAUSSRATE software or in any copyrights for the GAUSSRATE 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).

B. Publications resulting from using this package or the POLYRATE or GAUSSIAN subsystems used by this package will cite the corresponding program. The required references are given in the documentation (see Section 2 for the required references for GAUSSRATE).

C. 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.

D. No consulting or maintenance services are guaranteed or implied.

E. The POLYRATE 2015, and GAUSSIAN 09, or GAUSSIAN 03, or GAUSSIAN 98, or GAUSSIAN 94 codes, required to use GAUSSRATE, are covered by separate licenses.
2. **REQUIRED REFERENCES**

Publications based on results obtained with this computer code should include the following references:


and


or


or


or


Additional references are given in the documentation for POLYRATE and GAUSSIAN.
3. INTRODUCTION

GAUSSRATE 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 in the GAUSSIAN 09 (or 03 or 98 or 94) packages to represent the potential energy of interaction and its first and second derivatives whenever they are needed. This version, 2015/P2015-G09G03G98G94, interfaces POLYRATE–version 2015, which is a program for dynamical rate calculations, with the electronic structure program GAUSSIAN 09 (or 03 or 98 or 94).

Using GAUSSRATE presuppose that the user has both POLYRATE 2015 and GAUSSIAN 09 (or 03 or 98 or 94) on the same computer working properly before attempting to run GAUSSRATE. No part of POLYRATE 2015 or GAUSSIAN 09 (or 03 or 98 or 94) is contained in GAUSSRATE or needs to be modified for running this program.

This GAUSSRATE manual covers only material that is not covered in the POLYRATE and GAUSSIAN 09 (or 03 or 98 or 94) manuals. The user of GAUSSRATE should read those manuals first.

The interface between POLYRATE and GAUSSIAN 09 (or 03 or 98 or 94) 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 2015 and by a C shell script for running GAUSSIAN 09 (or 03 or 98 or 94). This C shell script, named shuttle, is called from inside GAUSSRATE, and it has to be able to start running GAUSSIAN 09 (or 03 or 98 or 94) 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 shuttle script is provided, and in most cases the user will only need to change one or two line, in order to set the gausspath variable to the appropriate path where GAUSSIAN 09 (or 03 or 98 or 94) is located and to set the executable to g09 (or g03 or g98 or g94). In the provided script, the variable scratchdir, which indicates the directory where the temporary and scratch GAUSSIAN 09 (or 03 or 98 or 94) files are going to be located, is set to the working directory; if desired, the user can change this. In some systems, the g09.login (or g03.login or g98.login or g94.login) files have to be loaded previously to any GAUSSIAN 09 (or
03 or 98 or 94) calculations. In such a case, the user should uncomment the line in the shuttle script that loads it or contact the system manager in order to get instructions about how to load this file.

Note that running two GAUSSRATE calculations or a GAUSSIAN and a GAUSSRATE calculation in the same directory can cause problems due to overwriting of some of the temporary files.

GAUSSRATE uses a subroutine usually included in libraries provided with FORTRAN compilers. This subroutine is called system. The prospective user of GAUSSRATE should be aware that this utility program is needed for running GAUSSRATE.

GAUSSRATE starting from version 2008 can perform reaction-path VTST (RP-VTST) and variable-reaction-coordinate VTST (VRC-VTST) calculations. The RP-VTST can be used for arbitrary kinds of reactions with tight transition states; VRC-VTST should be used for loose transition states of bimolecular association reactions. The RP-VTST and VRC-VTST calculations use different executables, i.e. gaussrate.exe for RP-VTST and gaussrate-vrc.exe for VRC-VTST.
4. DISTRIBUTION

GAUSSRATE is a set of subroutines for interfacing POLYRATE and GAUSSIAN. 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

– GAUSSIAN: GAUSSIAN 09 (or GAUSSIAN 03 or GAUSSIAN 98 or GAUSSIAN 94) is a licensed code available from Gaussian Inc. For further information see:
  http://www.gaussian.com

– GAUSSRATE: The GAUSSRATE 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/gaussrate

– system: This utility code is also required. It is available in standard FORTRAN libraries:

The prospective user of GAUSSRATE must obtain and install POLYRATE and GAUSSIAN 09 (or 03 or 98 or 94) before proceeding with GAUSSRATE.

The GAUSSRATE program is distributed as a compressed tar file named gaussrate2015-A.tar.gz. The distributed file should be uncompressed and untarred in the polyrate2015 directory. To uncompress, enter:

  gunzip gaussrate2015.tar.gz

The uncompressed file will be named gaussrate2015.tar. After extracting the files from the tar file, which can be done with the command:
A new directory, `gaussrate`, is created. This directory contains all the files included in the distribution package. If these instructions are followed correctly, the directory structure should be the following tree structure:

```
polyrate2015
  |  |
  doc exe gaussrate obj poten script src testo testrun util
  |  |  |  |  |  |  |  |  |  |  |
  script source testo testrun
  |  |  |  |  |  |  |  |  |  |  |
g09  g03  g94  g98  ch4oh  ch5  oh3  nh3oh  h3  c2h6
```

The files will be distributed among the directories as follows:

`gaussrate`: the C shell script `shuttle` and five subdirectories:

- `script`: 1 files: `grcompile`
- `testo`: This directory contains the following subdirectories:
  - `g03`: 15 files: `ch4ohtr1.fu6`, `ch4ohtr2.fu6`, `ch4ohtr3.fu15`, `ch4ohtr4.fu15`, `ch4ohtr5.fu15`, `ch4ohtr6.fu15`, `ch5tr1.fu15`, `ch5tr2.fu15`, `oh3tr1.fu15`, `oh3tr2.fu15`, `oh3tr3.fu15`, `nh3ohtr1.fu15`, `h3tr1.fu15`, `h3tr2.fu15`, `c2h6tr1.fu15"

These subdirectories contain selected output files of the test runs.

testrun: 4 files: check03, check94, check98, and run_all.jc, and 6 subdirectories: c2h6, ch4oh, ch5, oh3, nh3oh, and h3 with the input files for the test calculations.
5. INSTALLATION

First the user should obtain licensed copies and install the POLYRATE 2015 and GAUSSIAN 09 (or 03 or 98 or 94) packages. Next the user should obtain a licensed copy of the GAUSSRATE package. After downloading, uncompressing, and untarring the gaussrate.tar file, the GAUSSRATE package should appear as a directory, as described in Section 4.

There are a few variables that have to be set large enough to accommodate the system or systems to be studied. These variables are set in the include file param.inc in the POLYRATE side of the interface. The user should consult the POLYRATE manual for additional details.

The installation of the GAUSSRATE program is carried out by running the Perl script configure. The script will copy hooks_gr.wks to the hooks_mdep.f, will compile the POLYRATE 2015 files substituting the hooks.f file of standard POLYRATE by hooks_gr.f and hooks_mdep.f, and will create the executable file gaussrate.exe in the exe subdirectory in the polyrate2015 directory. The executable gaussrate.exe will then be accessible for all the GAUSSRATE calculations.

When running the installation script configure, the script will interactively ask if you want to choose the option that compile the codes using variable reaction coordinate variational transition state theory with multifaceted dividing surfaces for barrierless association reactions. This option will generate a separate executable file (gaussrate-vrc.exe) that is only used for VRC-VTST calculations. This code is parallelized using MPI. This is a new capability beginning with version 2008.

Before running GAUSSRATE, two lines of the C shell script shuttle should be changed as described in Section 1., in order to set the gausspath variable to the appropriate path where GAUSSIAN 09 (or 03 or 98 or 94) is located and set the command g09 (or g03 or g98 or g94) used to run GAUSSIAN job.
6. PROGRAM DESCRIPTION

The GAUSSRATE program follows the POLYRATE hooks protocol. This means that if energy, gradient, and/or Hessian calculations are desired by POLYRATE, the program calls the appropriate hooks subroutine that will return that information. (The user may read more about hooks in the POLYRATE manual.)

6.1. Description of GAUSSRATE subprograms

This is a brief description of GAUSSRATE 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, it calculates the energy using GAUSSIAN 09 (or 03 or 98 or 94).

*enatin* Calculates the energy for atomic systems.

*escener* Carries out GAUSSIAN energy calculations.

*escfirst* Carries our gaussian energy and first derivative calculations.

*escsec* Carries out GAUSSIAN Hessian calculations.

*geoin* Carries out GAUSSIAN 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 GAUSSIAN input file for energy calculations.

*ginpftrs* Creates the GAUSSIAN input file for gradient calculations.

*ginpsec* Creates the GAUSSIAN input file for Hessian calculations.
**goutsec**
Reads the formatted checkpoint file Test.FChk generated by GAUSSIAN and extracts the geometry, energy, and first and second derivatives of the energy with respect to atomic coordinates.

**grest**
Reads and writes the information in the restart files of a GAUSSRATE 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 GAUSSIAN or POLYRATE methods.

**ohook**
Equivalent to the ohook subroutine in the POLYRATE hooks. Optimizes a given initial geometry, using either POLYRATE or GAUSSIAN 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.

**rchmul**
Reads the charge and multiplicity of a system from the GAUSSIAN formatted checkpoint file.

**rcodgr**
Reads the input option for the GAUSSIAN runs from file esp.fu70.

**rendef**
Reads the information about reactants, products and saddle point calculated in the prep subroutine.

**renergr**
Reads the input options for GAUSSIAN energy calculations.

**rfirstgr**
Reads the input options for GAUSSIAN gradient calculations.

**rgenbas**
Reads the basis set for GAUSSIAN calculations when a nonstandard basis set is used.
**rgengr**  
Reads the options in the general section of the *esp.fu70* input file.

**rinpgr**  
Reads the *esp.fu70* GAUSSRATE input file.

**rlincer**  
Reads the *Link0* options for GAUSSIAN calculations.

**rlingr**  
Utility program for reading in a line of the *esp.fu70* input file and parsing off comments and extra spaces.

**rsecgr**  
Reads the input options for GAUSSIAN Hessian calculations.

**rstatgr**  
Reads the input options given in the *esp.fu70* input file for the stationary points.

**secin**  
Carries out GAUSSIAN Hessian calculations for stationary points.

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

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

Note that GAUSSRATE does not use any of the utility routines (routines starting with *y*) of the generic hooks package of POLYRATE.

### 6.2. Restarting a calculation

The GAUSSRATE 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 GAUSSRATE, but GAUSSRATE also has its own, unique restart capability. The rest of this section concerns this GAUSSRATE restart capability.

The GAUSSRATE restart option is controlled by the GRRESTART keyword in the GRGENERAL section of the unit *esp.fu70* input file. The GAUSSRATE 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 \textsc{gaussian} 09 (or 03 or 98 or 94) 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: \texttt{esp.fu83}, which stores the information to be read, as originated in a previous \texttt{gaussrate} calculation, and \texttt{esp.fu84}, where the useful information read from \texttt{esp.fu83} is stored along with the information calculated for the first time in the present \texttt{gaussrate} run (\texttt{esp.fu84} is the file to be used for future restart runs). The \texttt{esp.fu84} file has to be created by \texttt{gaussrate}, 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 \texttt{gaussrate} 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 \texttt{gaussrate} run. One should rename the \texttt{esp.fu84} file provided by the first run as \texttt{esp.fu83} and then run again with \texttt{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 \texttt{gaussian} 09 (or 03 or 98 or 94) 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 SCALEMASS. Changing this parameter will cause a miscalculation of the frequencies. It is important to note that its default value is 1.0 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 GAUSSRATE.

A GAUSSRATE 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 test runs in the GAUSSRATE manual.

The GAUSSRATE restart option works in the following way: when POLYRATE needs an energy, gradient, and/or Hessian for a given geometry, it first checks if the geometry is stored in esp.fu83, and then it checks if the required information is stored in this same file. If one of these conditions is not fulfilled, a GAUSSIAN calculation is run to obtain the desired information. If both conditions are fulfilled, the information is read from esp.fu83, and no GAUSSIAN job is spawned. The geometry check is performed by comparing each of the Cartesian coordinates of the desired geometry with the Cartesian coordinates in esp.fu83. If the values differ by more than a criterion given by RSTTOL, the geometries are considered to be different. It is important to have an appropriate value for RSTTOL: if the criterion is too tight, the check might fail due to the numerical round-off and to the fact that the precision of the written information is limited, and redundant calculations could be carried out, increasing unnecessarily the cpu time of the calculation. However, if this criterion is too loose, the check could be passed for two geometries that are only slightly different, but different enough to matter, and the restart option can be giving to GAUSSRATE inaccurate information. The later case can be especially true when performing reaction-path calculations with very small step-sizes, numerical differentiations, or accurate geometry optimizations. Therefore, the user must be very careful about changing this parameter. The default value is $1.D-8$, which is very
conservative, but it is safe. For very expensive calculations, one might want to use a higher value, but a value that is too high can give unphysical results. If the user increases this parameter, he or she has to be very aware of the risk it poses in the calculation, and all the results must be carefully checked. Therefore, only experienced users are advised to change this parameter.

Note that the number of processors should not be changed when restarting a parallel calculation using variable reaction coordinate transition state theory for loose transition states.

6.3. Parallel capabilities

The VRC-VTST code in the POLYRATE program is highly parallelized using MPI. This portion of the program uses variable reaction coordinates and multifaceted dividing surfaces for association reactions with loose transition states. The most time-consuming part of the calculation is an integral on an eight-dimensional multifaceted hypersurface in the coordinate space of the interacting reactants. This integral is evaluated by the Monte Carlo method. The Monte Carlo integration requires thousands of samples of the energy to achieve a reasonably accurate result, and each sample requires a full electronic structure calculation by the GAUSSIAN program. Each core independently performs GAUSSIAN calculations for the Monte Carlo samples generated in that core. When the Monte Carlo sampling is finished on all cores, one core (rank 0 in MPI) is used to collect the data calculated on all the cores and to perform the rest of the calculations to get reaction rate constants.

GAUSSIAN has two parallel versions. One is an OpenMP or symmetric multiprocessing (SMP) version that is used in a shared memory system, and the other one is a Linda version that can be used in a distributed memory system. There are some technical difficulties to use the Linda implementation of the GAUSSIAN program together with the MPI POLYRATE code together. Therefore only the SMP version of GAUSSIAN is tested for GAUSSRATE.

To run a parallel VRC-VTST calculation, the number of the cores to be used should be specified as a flag of the mpirun command in the .jc file (job control file). The same number of the cores should also be specified in the queue submission script. For example, if $n$ cores are required for a parallel calculation, one should use the command “mpirun –
np n gausrate-vrc.exe” in a .jc file. Note that if the keyword “%nprocs=1” is set or omitted in the esp.70 file (see Section 7.4), each GAUSSIAN calculation is performed by one core, and n Monte Carlo samples are calculated simultaneously.

If desired, one can use two kinds of parallelism together in a VRC-VTST calculation. In particular, one can use multiple cores for each SMP GAUSSIAN calculation and also multiple cores for the MPI VRC-VTST code. Here is an example to show how to use this double parallel strategy at script level. Assume that n cores on each computer node are used in each SMP GAUSSIAN calculation and m computer nodes are used for the MPI VRC-VTST calculation.

- In the esp.70 file, one should add the “%nprocs=n” keyword that is used for each GAUSSIAN job.
- In the .jc file, one should use the command “mpirun –np m –bynode gausrate-vrc.exe”.
- In the .pbs submission file, one should specify the number of cores and nodes as “nodes=m: ppn=n”.

In this setting, there are totally n × m cores are used in the whole VRC-VTST direct dynamics calculation. In this calculation, each GAUSSIAN calculation is performed by n cores, and m Monte Carlo samples are calculated simultaneously. This hybrid parallel strategy is only recommended when a large number of cores are available, and of course it can only be used when each node has multiple cores.

The RP-VTST code in the POLYRATE program only performs direct dynamics sequentially. However, RP-VTST calculations can be performed using multiple cores for direct dynamics by using them in each SMP GAUSSIAN calculation. For example, if n cores are available for an SMP GAUSSIAN calculation, the user can specify the number of cores by the “%nprocs=n” keyword in the esp.fu70 file (see Section 7.4) and require n cores in the queue submission script for RP-VTST calculations. Therefore, the whole dynamics calculation is performed by the serial RP-VTST code sequentially, while each electronic structure calculation required for direct dynamics is performed in parallel.
7. **DESCRIPTION OF INPUT FILES**

We will assume that the user knows how to use POLYRATE 2015 and GAUSSIAN 09 (or 03 or 98 or 94) and is familiar with all their keywords. Therefore, when talking about the common input files, we will only discuss the differences between running POLYRATE 2015 and running GAUSSRATE 2015.

7.1. **File usage**

In addition to the POLYRATE 2015 and GAUSSIAN 09 (or 03 or 98 or 94) input files, GAUSSRATE 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.

- **esp.fu70** General GAUSSRATE input data.
- **esp.fu71** GAUSSIAN 09 (or 03 or 98 or 94) input file for geometry optimization of reactant 1, if such optimization is going to be carried out.
- **esp.fu72** GAUSSIAN 09 (or 03 or 98 or 94) 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 GAUSSIAN 09 (or 03 or 98 or 94) program.
- **esp.fu73** GAUSSIAN 09 (or 03 or 98 or 94) input file for geometry optimization of product 1, if such optimization is going to be carried out.
- **esp.fu74** GAUSSIAN 09 (or 03 or 98 or 94) 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 GAUSSIAN 09 (or 03 or 98 or 94) program.
- **esp.fu75** GAUSSIAN 09 (or 03 or 98 or 94) 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 GAUSSIAN 09 (or 03 or 98 or 94) 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 GAUSSIAN 09 (or 03 or 98 or 94).

esp.fu78 GAUSSIAN 09 (or 03 or 98 or 94) 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 GAUSSIAN 09 (or 03 or 98 or 94).

esp.fu81 GAUSSIAN 09 (or 03 or 98 or 94) temporary input file, generated by GAUSSRATE.

esp.fu82 GAUSSIAN 09 (or 03 or 98 or 94) temporary output file, generated by GAUSSIAN. If the GAUSSIAN 09 (or 03 or 98 or 94) 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 Section 6.1).

esp.fu84 Restart file to be written during a restart calculation (see Section 6.1).

esp.fu85 GAUSSRATE uses the formatted checkpoint file generated by GAUSSIAN 09 (or 03 or 98 or 94). GAUSSIAN 09 (or 03 or 98 or 94) calls this file Test.FChk, and GAUSSRATE does not delete it. GAUSSRATE reads this file from FORTRAN unit 85. This file will not be removed after finishing a calculation, so if the GAUSSIAN 09 (or 03 or 98 or 94) 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.
7.2. Description of fu5 unit

A GAUSSRATE 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 we 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 2015 manual. It is important to clarify some points about preparing the input file for unit fu5:

RESTART

If a POLYRATE 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 GAUSSRATE, controlled by the GRRESTART keyword in GRGENERAL section of the esp.fu70 input file, 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 GAUSSRATE 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 GAUSSRATE.
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 GAUSSRATE there are two ways to optimize the geometries: by means of GAUSSIAN 09 (or 03 or 98 or 94) methods or by means of POLYRATE methods. The choice between the two methods is done in the OPTIMIZATION section of the fu5 input file. Choosing the option BFGS, NR, or EF, for the OPTMIN or OPTTS keywords in the OPTIMIZATION section, will cause GAUSSRATE to use the methods native to POLYRATE, 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. In these cases, 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, GAUSSRATE will optimize the geometries of the stationary points using the GAUSSIAN 09 (or 03 or 98 or 94) 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 GAUSSIAN 09 (or 03 or 98 or 94) 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.

**SECOND**

The calculation of the Hessian matrices can be done by numerical differentiation of the gradients, by means of the POLYRATE methods, or by the GAUSSIAN 09 (or 03 or 98 or 94) 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 GAUSSIAN 09 (or 03 or 98 or 94) can perform Hessian calculations using any of the electronic structure methods available in GAUSSIAN, the option of performing the Hessian calculation using POLYRATE methods is implemented for completeness. However the user is advised for efficiency reasons against using POLYRATE for the Hessian calculations.

**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 GAUSSIAN 09 (or 03 or 98 or 94) 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 GAUSSIAN 09 (or 03 or 98 or 94)
If a user selects POLYRATE 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 2015 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 GAUSSIAN 09 (or 03 or 98 or 94) 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 GAUSSIAN 09 (or 03 or 98 or 94) optimization. The second one is by means of the GEOM keyword, if a POLYRATE optimization (BFGS, NR, or EF 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 GAUSSIAN 09 (or 03 or 98 or 94) 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 to be set to GEOM, 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 procedure of the initial and/or optimized geometries for the stationary points. The combinations are:

- For optimizing the geometries using the methods in GAUSSIAN: OPTTS and OPTMIN in the OPIMIZATION 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, NR and/or EF, 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 GAUSSIAN 09 (or 03 or 98 or 94) formatted checkpoint 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 2015, 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. The only anharmonicity option that do not require derivatives higher than second are Morse I and hindered rotator.

In addition, when calculating second or higher order derivatives using the numerical methods of POLYRATE 2015, GAUSSIAN 09 (or 03 or 98 or 94) is unable to successfully calculate some slightly distorted structures of linear molecules, such as the hydrogen...
molecule, stopping in the SYMM subroutine. Therefore, we recommend that Hessians be calculated with GAUSSIAN rather than with the methods native to POLYRATE.

**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 discussion of SCALEMASS in Section 6.2.

7.3. **Description of esp.fu71 through esp.fu78 input files**

The data files esp.fu71 through esp.fu78 are GAUSSIAN-type data files. They are used only as input files for GAUSSIAN optimization on reactants 1 and 2 (esp.fu71 and esp.fu72, respectively), products 1 and 2 (esp.fu73 and esp.fu74), of the well on the reactant side (esp.fu77), of the well on the product side (esp.fu78), and of the saddle point or other starting geometry (esp.fu75). If no optimization using GAUSSIAN is desired, these files are not needed.

For constructing these files, the user must check the GAUSSIAN manual. With only one restriction, any option available for GAUSSIAN can be used; the restriction is that GAUSSIAN must generate a formatted checkpoint file at the end of the optimization. The GAUSSIAN keyword FCHK (or any of its equivalent keywords, FORMCHECK or FCHECK) must therefore be specified, and options in GAUSSIAN that do not generate such a file (for example, a non-standard route) cannot be used.

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.

7.4. Description of esp.fu70 input file

Along with the input fu5 file, which controls the POLYRATE options, the fu70 input file will control all of the GAUSSRATE options and some of the GAUSSIAN 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.

<table>
<thead>
<tr>
<th>Section name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRGENERAL</td>
<td>Restart options</td>
</tr>
<tr>
<td>GRCOMMON</td>
<td>GAUSSIAN options common to all the stationary points</td>
</tr>
<tr>
<td>GRREACT1</td>
<td>GAUSSIAN options specific to the first reactant</td>
</tr>
<tr>
<td>GRREACT2</td>
<td>GAUSSIAN options specific to the second reactant, if any</td>
</tr>
<tr>
<td>GRPROD1</td>
<td>GAUSSIAN options specific to the first product</td>
</tr>
<tr>
<td>GRPROD2</td>
<td>GAUSSIAN options specific to the second product, if any</td>
</tr>
<tr>
<td>GRWELLR</td>
<td>GAUSSIAN options specific to the reactant well, if any</td>
</tr>
<tr>
<td>GRWELLP</td>
<td>GAUSSIAN options specific to the product well, if any</td>
</tr>
<tr>
<td>GRSTART</td>
<td>GAUSSIAN options specific to the starting (or saddle) point</td>
</tr>
</tbody>
</table>
GRPATH

GAUSSIAN options common to all the nonstationary points along the reaction path

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

7.4.1. The GRGENERAL section

The GRGENERAL section controls GAUSSRATE’s own options. There are only three keywords allowed in this section, GRRESTART and NOGRRESTART, with GRRESTART being the default, and RSTTOL.

**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, GAUSSRATE runs GAUSSIAN and writes the GAUSSIAN output information to unit *fu84*. If NOGRRESTART is selected, neither reading file *fu83* nor writing file *fu84* will be performed.

**RSTTOL**

RSTTOL is a variable keyword that must be followed by a real number, which is the value of the tolerance in the comparison of geometries between the geometry currently used by GAUSSRATE and the geometry found in a restart file *esp.fu83*. The meaning of this keyword is explained in detail in Section 6.2. The inexperienced user is not advise to change this value from its default value, $1.E-8$.

*Example:*

| RSTTOL | 0.00000001 |
7.4.2. The GRCOMMON section

The GRCOMMON section reads the information common to all the GAUSSIAN calculations for the stationary points and points along the reaction path, excepting the GAUSSIAN 09 (or 03 or 98 or 94) 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 GAUSSIAN 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 GAUSSIAN will use for carrying out an energy calculation. The default for this keyword is #HF/STO-3G UNITS=AU FCHK NOSYMM. In order to get the desired results, two particular keywords must always be supplied for this kind of calculation: UNITS=AU and FCHK (or their equivalent GAUSSIAN keywords; see the GAUSSIAN manual for a discussion of alternative forms of GAUSSIAN keywords). No blank lines can appear in this list, since GAUSSIAN would take those blank lines as the end of keyword inputs. Just as in GAUSSIAN inputs, the pound symbol (#) must precede the complete set of keywords.

*Example:*

```plaintext
GRENER
  #HF/6-31G SCF=QC UNITS=AU FCHK
END
```

**GRFIRST**

This accomplishes a similar purpose as the GRENER keyword, but it indicates the keywords to be used in gradient calculations. Again, UNITS=AU and FCHK are required. The default is #HF/STO-3G UNITS=AU FCHK NOSYMM FORCE.
GRSEC

This keyword is only used when the $HHOOK$ option is specified in the HESSCAL keyword of the SECOND section of the $fu_5$ input file. It accomplishes a purpose similar to the GRENER and GRFIRST keywords, but indicating the options for a Hessian calculation. $UNITS=AU$ and $FCHK$ are required, and the $NORAMAN$ option is recommended (since it saves computer time). The default is $\#HF/STO-3G$ $UNITS=AU$ $FCHK$ $NOSYM$ $FREQ=NORAMAN$.

GRLINK0

GRLINK0 is a list keyword for indicating the link0 options for GAUSSIAN calculations (note that the last character is zero). This keyword controls the options for renaming scratch files, creating checkpoint files, determining memory size, etc. If it is not indicated, no link0 options are assumed, since they are not needed for the calculations. If indicated, they are assumed to be the same for all the GAUSSIAN calculations, no matter which species or type of calculation is involved, the only exception being the GAUSSIAN 09 (or 03 or 98 or 94) optimizations which are fully controlled by the $fu_70$ through $fu_78$ input files. No blank lines are allowed in this keyword.

GRBASIS

When running GAUSSIAN, if a non-standard basis set is desired, the keyword GEN must be indicated, and the general basis set must follow the geometry specification in the GAUSSIAN input. If a non-standard basis set is desired in a GAUSSRATE calculation, it must be indicated in GRBASIS, and it must be the same for all the calculations on all the systems. Again, no blank lines are allowed, and (because of dimension limitations) a maximum of 100 lines are allowed for indicating the basis set. If a larger space is required, the basis set can be specified by an external file, using the symbol '@' (see the GAUSSIAN 09 (or 03 or 98 or 94) manual for further details of this option). For example, for selecting a modified STO-3G basis set, located in the msto3g.gbs file in the directory /usr/g94/basis, one should write:

Example:

GRBASIS
@/usr/g94/basis/msto3g.gbs
END
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 GAUSSIAN 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 GAUSSIAN inputs (except the optimizations, which are read directly from units fu71-fu78), and no blank lines are allowed in this keyword. The later 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 GAUSSIAN 09 (or 03 or 98 or 94) input file if nonstandard isotopes are desired, should never be included in a GAUSSRATE run. The choice of the isotopic masses has to be done in the fu5 input file of POLYRATE.

7.4.3. The GRREACT1, GRREACT2, GRPROD1, GRPROD2, GRWELLR, GRWELLP, GRSTART, and GRPATH sections

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 us 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 GAUSSIAN 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.

These sections can include one or more of the GRENER, GRFIRST, and GRSEC keywords explained above for the GRCOMMON section, with the same syntax that we explained in the previous section, and requiring the keywords UNITS=AU and FCHK in all the inputs.
In order to avoid problems related to the reorientation of the systems that GAUSSIAN 09 (or 03 or 98 or 94) performs, in addition to *UNITS=AU* and *FCHK*, the keyword *NOSYMM* is strongly recommended for its use in the GRPATH section; otherwise, the results could be wrong even though they would seem reasonable and no error would be detected without a careful study of the directions of the gradients (see the POLYRATE manual for a more thorough discussion of reorientation problems and GAUSSIAN reorientation).

The *NOSYMM* keyword is also recommended when third or fourth derivatives of the energy are needed (i.e., for anharmonic calculations or when using the cubic algorithm for starting the reaction path) and when the *HOOK* option is selected for the HESSCAL keyword in the SECOND section of the *fu5* input file is on, since for the numerical differentiation of the derivatives computed using GAUSSIAN it is necessary to have them oriented in a consistent way.

The options entered by means of GRBASIS and GRLINK0 keywords in the GRCOMMON section are used in all the GAUSSIAN calculations for which input is constructed by GAUSSRATE (all the calculations but geometry optimizations, for which the input is written by the user in units *fu71*-*fu78*); therefore, the GRBASIS and GRLINK0 keywords are only allowed in the GRCOMMON section and not in the .GRREACT1, GRREACT2, GRPROD1, GRPROD2, GRWELLR, GRWELLP, GRSTART, and GRPATH sections.

Besides using the GRENER, GRFIRST, and GRSEC keywords, some additional issues need to be clarified, and two new keywords need to be introduced:

**CHARGE** and **MULTIPLICITY**

If GAUSSIAN optimization is required, GAUSSRATE will read the charge and multiplicity of each species from the GAUSSIAN optimization output. Nevertheless, if no optimization is selected or if POLYRATE 2015 is selected for carrying out the optimizations, GAUSSRATE will need to know the charge and multiplicity of the system before making any GAUSSIAN calculation. The way to input these characteristics is by means of the CHARGE and MULTIPLICITY keywords. These keywords are not allowed in the GRCOMMON section since they are dependent on the species and are only required for the systems which will not be optimized using GAUSSIAN 09 (or 03 or 98 or 94), being ignored if they
are specified for a species that is going to be optimized using GAUSSIAN. For the points along the reaction path, the charge and multiplicity are assumed to be the same as the values for the saddle point or starting geometry, and if they are specified in the GRPATH section they will also be ignored.

CHARGE is a variable keyword that must be followed by an integer, which is the value of the charge of the system. The default value for CHARGE is 0. MULTIPLICITY is also a variable keyword followed by an integer indicating the multiplicity of the species. The default value for MULTIPLICITY is 1, which corresponds to a singlet. Values of 2 and 3 correspond to doublets and triplets.

Example:

```
CHARGE       -1
MULTIPLICITY  2
```
8. SPEED UP THE DIRECT DYNAMICS CALCULATIONS USING GAUSSRATE

The direct dynamics calculations can be speeded up by decreasing the time required to reach the SCF convergence threshold in the electronic structure calculations (i.e., when the energies, gradients, and Hessians are computed) of the non-stationary points along the minimum energy path (MEP). To accomplish this, the molecular orbitals guess generated in the previous electronic structure calculation must be used as the initial guess in the current calculation, instead of computing it from *ab initio*. In this way, problems with the SCF convergence along the MEP should be avoided. A couple of examples given in test run directory are provided to demonstrate this strategy. Below it is explained how to perform such calculations.

Because the Gaussian formatted checkpoint file (*Test.FChk*) is written for all electronic structure calculations, it can be used to generate its corresponding binary checkpoint file using the *unfchk* GAUSSIAN utility (i.e., the sentence *$gausspath/unfchk Test.FChk $filename.chk* must be added to the *shuttle* file).¹

*Example of shuttle:*

```
# $gausspath/unfchk Test.FChk guess.chk
/bin/rm $argv[2]
/bin/rm Test.FChk
#
```

That checkpoint file must be read by the following GAUSSIAN calculation, i.e. the keyword *%chk=guess.chk* and *guess=read* in the route section of Gaussian must be added in all the items of the GRPATH section of the *esp.fu70* file.

*Example of *esp.fu70* file:*

```
*GRPATH
  GRENER
    %chk=guess.chk
```

¹ This sentence is included by default in the *shuttle* file by default.
The first non-stationary point computed along the MEP uses the guess of the transition structure (TS) as its initial guess, and it must be generated in a previous step. There are two possible ways to do that.

In the first way GAUSSRATE computes the guess of the TS and writes it in a checkpoint file, i.e. the keyword `%chk=guess.chk` must be added to the esp.fu75 file. (Note that this way is valid only if the value of the STATUS keyword of the *START section in the POLYRATE.dat input file is equal to 0 or 2).

Example of esp.fu75:

```bash
%chk=guess.chk
# HF/6-31G OPT=(TS,NOEIGENTEST,CALCFC) FCHK
```

In the second way, the GAUSSIAN checkpoint file (filename.chk) that contains the initial guess of the TS must be computed in a previous step to run GAUSSRATE, and then copied to GAUSSRATE work directory by including the sentence `cp $actdir/$filename.chk guess.chk` in the .jc file. It should be noted that this method is designed for when the energy and the Hessian or the frequencies of the TS are provided by the user in the POLYRATE.dat input file using the STATUS keyword equal 4 or, 6 and the use of STATUS 6 for reactants and products is mandatory.
Example of `.jc` file:

```
cp $actdir/guess.chk guess.chk
#
# Write system message to .time file
#
(time $exe) >& $actdir/$name.time
echo " " >> $actdir/$name.time
```
9. TESTING GAUSSRATE

9.1. Tests performed

We tested the version 2015 of GAUSSRATE using version 2015 of POLYRATE and the indicated versions of GAUSSIAN 09/03 on the following machines (the operating system is indicated in parentheses):

1. HP Linux cluster with Intel Xeon X5560 "Nehalem EP" processors; GAUSSIAN 09 Rev. D01

The code is expected to work with all GAUSSIAN 09/03 versions that use input files and formatted checkpoint files with the same structure as the above mentioned versions.
9.2. Test suite

The purpose of the test suite is to provide some examples of the way GAUSSRATE works, not to check all GAUSSIAN 09 (or 03 or 98 or 94) or POLYRATE 2015 options. Therefore, the number of tests included is only ten, and the tests illustrate only the most basic features of GAUSSRATE. 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.

The test suite is located in the testrun directory that contains several subdirectories. Each subdirectory contains one or more tests based on the same reaction. The summary output files for these test runs that are distributed in the current version of the program are located in the testo directory. The Korn shell scripts check03, check98, and check94 in the testrun directory allow an easy comparison between the poly.fu15 output files obtained after running all the test runs using GAUSSIAN 09, GAUSSIAN 03, GAUSSIAN 98, and GAUSSIAN 94, respectively, and the equivalent poly.fu15 files in the testo directory.

Each directory in testrun includes one or more tests based on one of the following reactions:

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Directory</th>
<th>Test names</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH$_3$ + H$_2$ → CH$_4$ + H</td>
<td>ch5</td>
<td>ch5tr1, ch5tr2</td>
</tr>
<tr>
<td>OH + H$_2$ → H$_2$O + H</td>
<td>oh3</td>
<td>oh3tr1, oh3tr2, oh3tr3.</td>
</tr>
<tr>
<td>CH$_4$ + OH → CH$_3$ + H$_2$O</td>
<td>ch4oh</td>
<td>ch4ohtr1, ch4ohtr2, ch4ohtr3, ch4ohtr4, ch4ohtr5, ch4ohtr6.</td>
</tr>
<tr>
<td>NH$_3$ + OH → NH$_2$ + H$_2$O</td>
<td>nh3oh</td>
<td>nh3ohtr1</td>
</tr>
<tr>
<td>CH$_3$ + CH$_3$ → C$_2$H$_6$</td>
<td>c2h6</td>
<td>c2h6tr1</td>
</tr>
</tbody>
</table>

The test runs are run in a subdirectory created for each run. This way, problems that could be caused when running other GAUSSIAN or GAUSSRATE calculations in the same directory that GAUSSRATE is using are avoided. After the test run is completed, the directory is removed and the temporary files created are moved to the GRTEMPFILES directory.
9.2.1. Test run ch5tr1

Tunneling: SCT
Number of levels: single level
Hindered rotation: none

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 2015 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 GAUSSIAN calculations (which is the default for GAUSSIAN 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 GAUSSIAN 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).

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch5tr1.70</td>
<td>esp.fu70</td>
</tr>
<tr>
<td>ch5tr1.71</td>
<td>esp.fu71</td>
</tr>
<tr>
<td>ch5tr1.72</td>
<td>esp.fu72</td>
</tr>
<tr>
<td>ch5tr1.73</td>
<td>esp.fu73</td>
</tr>
<tr>
<td>ch5tr1.74</td>
<td>esp.fu74</td>
</tr>
<tr>
<td>ch5tr1.75</td>
<td>esp.fu75</td>
</tr>
<tr>
<td>ch5tr1.dat</td>
<td>poly.fu5</td>
</tr>
<tr>
<td>ch5tr1.61</td>
<td>esp.fu61</td>
</tr>
<tr>
<td>ch5tr1.fu6</td>
<td>poly.fu6</td>
</tr>
<tr>
<td>ch5tr1.fu15</td>
<td>poly.fu15</td>
</tr>
<tr>
<td>ch5tr1.rst</td>
<td>esp.fu84</td>
</tr>
</tbody>
</table>
9.2.2. Test run ch5tr2

| Tunneling: | SCT |
| Number of levels: | single level |
| Hindered rotation: | none |

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.

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch5tr2.51</td>
<td>poly.fu51</td>
</tr>
<tr>
<td>ch5tr2.70</td>
<td>esp.fu70</td>
</tr>
<tr>
<td>ch5tr2.71</td>
<td>esp.fu71</td>
</tr>
<tr>
<td>ch5tr2.72</td>
<td>esp.fu72</td>
</tr>
<tr>
<td>ch5tr2.73</td>
<td>esp.fu73</td>
</tr>
<tr>
<td>ch5tr2.74</td>
<td>esp.fu74</td>
</tr>
<tr>
<td>ch5tr2.75</td>
<td>esp.fu75</td>
</tr>
<tr>
<td>ch5tr2.dat</td>
<td>poly.fu5</td>
</tr>
<tr>
<td>ch5tr2.61</td>
<td>esp.fu61</td>
</tr>
<tr>
<td>ch5tr2.fu6</td>
<td>poly.fu6</td>
</tr>
<tr>
<td>ch5tr2.fu15</td>
<td>poly.fu15</td>
</tr>
<tr>
<td>ch5tr2.rst.old</td>
<td>esp.fu83</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>ch5tr2.rst</td>
<td>esp.fu84</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
9.2.3. Test run oh3tr1

Tunneling: none
Number of levels: single level
Hindered rotation: none

This is a test run for a conventional TST calculation for the reaction \( \text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H} \). The geometries for the reactants and products are given (it is assumed that they already are fully optimized) in the fu5 input file (INITGEO is set to geom, and STATUS set to 2 in the REACT1, REACT2, PROD1 and PROD2 sections of the fu5 input file), while the saddle point optimization is carried out by means of GAUSSIAN, requiring, therefore, the esp.fu75 input file, and STATUS set to 0 and the option hooks for the INITGEO keyword in the START section of the input file fu5. The GAUSSRATE restart file esp.fu84 is written. The level of electronic structure calculation used for this test run is MP2, using Dunning's aug-cc-pVDZ basis set, which is entered by means of the GRBASIS keyword. The unformatted checkpoint file generated for GAUSSIAN is renamed oh3tr1.chk by means of the GRLINK0 keyword.

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
</tr>
</thead>
<tbody>
<tr>
<td>oh3tr1.70</td>
<td>esp.fu70</td>
</tr>
<tr>
<td>oh3tr1.75</td>
<td>esp.fu75</td>
</tr>
<tr>
<td>oh3tr1.dat</td>
<td>poly.fu5</td>
</tr>
<tr>
<td>oh3tr1.61</td>
<td>esp.fu61</td>
</tr>
<tr>
<td>oh3tr1.fu6</td>
<td>poly.fu6</td>
</tr>
<tr>
<td>oh3tr1.fu15</td>
<td>poly.fu15</td>
</tr>
<tr>
<td>oh3tr1.rst</td>
<td>esp.fu84</td>
</tr>
</tbody>
</table>
9.2.4. **Test run oh3tr2**

- **Tunneling:** none  
- **Number of levels:** single level  
- **Hindered rotation:** none

This is a test run that, by making use of the `oh3tr1.rst` file previously written by the `oh3tr1` test run, restarts the calculation performed by the previous test run and makes a canonical variational theory (CVT) calculation, without tunneling, by following the reaction path using the variational reaction path algorithm based on the Euler method (`VRPE` option of RPM keyword) with a step size of 0.025 bohr. The reaction path is calculated between −0.3 bohr and +0.3 bohr. Nonredundant internal coordinates are used for describing the vibrations. The user must run the `oh3tr1` test run before running `oh3tr2`, since the restart file to be read, `oh3tr2.rst`, will be a copy of the `oh3tr1.rst` file generated by the previous run and located in the same directory. Otherwise, the `oh3tr2.rst` file will not exist, and the program will calculate the frequencies for all the stationary points and the optimum geometry for the saddle point instead of reading them from the restart file. It should be noted that the `fu75` file is required, and it is used by `GAUSSRATE` although not by `GAUSSIAN`. At the end of the calculation, a new restart file, including the information calculated for the first time in the present run, will take the name of `oh3tr2.rst`, while the old restart file will be renamed as `oh3tr2.rst.old` (and this will be unchanged from what was in file `oh3tr1.rst` at the start of the present run).

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>oh3tr2.70</td>
<td>esp.fu70</td>
<td>Input data for GAUSSRATE</td>
</tr>
<tr>
<td>oh3tr2.75</td>
<td>esp.fu75</td>
<td>Input data for the saddle point</td>
</tr>
<tr>
<td>oh3tr2.dat</td>
<td>poly.fu5</td>
<td>Input data for POLYRATE</td>
</tr>
<tr>
<td>oh3tr2.61</td>
<td>esp.fu61</td>
<td>POLYRATE output file containing information about the stationary points</td>
</tr>
<tr>
<td>oh3tr2.fu6</td>
<td>poly.fu6</td>
<td>Long output file</td>
</tr>
<tr>
<td>oh3tr2.fu15</td>
<td>poly.fu15</td>
<td>Summary output file</td>
</tr>
<tr>
<td>oh3tr2.rst.old</td>
<td>esp.fu83</td>
<td>GAUSSRATE restart file read by GAUSSRATE in this run</td>
</tr>
<tr>
<td>oh3tr2.rst</td>
<td>esp.fu84</td>
<td>GAUSSRATE restart file generated by GAUSSRATE in this run</td>
</tr>
</tbody>
</table>
### 9.2.5. Test run oh3tr3

Tunneling: none  
Number of levels: single level  
Hindered rotation: none

This test run uses the `oh3tr2.rst` file written by the `oh3tr2` test run, copying it as `oh3tr3.rst`, and it performs the same calculation as the `oh3tr1` test. Since more information than is required for the calculation is stored in the `oh3tr2.rst` file, the restart file created by this test run, `oh3tr3.rst`, will be smaller than the file `oh3tr2.rst`. This is the reason why it is convenient to keep the file from which a restart calculation reads the information (in the executable scripts of these test runs, such files are renamed with the extension `old`). No GAUSSIAN calculations are needed for this test run, so it is the fastest test run of the suite.

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
</tr>
</thead>
<tbody>
<tr>
<td>oh3tr3.70</td>
<td>esp.fu70</td>
</tr>
<tr>
<td>oh3tr3.75</td>
<td>esp.fu75</td>
</tr>
<tr>
<td>oh3tr3.dat</td>
<td>poly.fu5</td>
</tr>
<tr>
<td>oh3tr3.61</td>
<td>esp.fu61</td>
</tr>
<tr>
<td>oh3tr3.fu6</td>
<td>poly.fu6</td>
</tr>
<tr>
<td>oh3tr3.fu15</td>
<td>poly.fu15</td>
</tr>
<tr>
<td>oh3tr3.rst.old</td>
<td>esp.fu83</td>
</tr>
<tr>
<td>oh3tr3.rst</td>
<td>esp.fu84</td>
</tr>
</tbody>
</table>

- `Input data for GAUSSRATE`  
- `Input data for the saddle point`  
- `Input data for POLYRATE`  
- `POLYRATE output file containing information about the stationary points`  
- `Long output file`  
- `Summary output file`  
- `GAUSSRATE restart file read by GAUSSRATE in this run`  
- `GAUSSRATE restart file generated by GAUSSRATE in this run`
9.2.6. Test run ch4ohtr1

Tunneling: none
Number of levels: single level
Hindered rotation: none

This is the first of a series of tests that try to show how GAUSSRATE 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 as 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, \( \text{CH}_4 \). In the fu5 input file we only include the REACT1 section, and only the additional input files fu70 and fu71 are required. The GAUSSRATE 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 esp.fu61 will contain the information calculated for this reactant, and this information will be included in the fu5 input file for the following runs.

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch4ohtr1.70</td>
<td>esp.fu70</td>
</tr>
<tr>
<td>ch4ohtr1.71</td>
<td>esp.fu71</td>
</tr>
<tr>
<td>ch4ohtr1.dat</td>
<td>poly.fu5</td>
</tr>
<tr>
<td>ch4ohtr1.61</td>
<td>esp.fu61</td>
</tr>
<tr>
<td>ch4ohtr1.fu6</td>
<td>poly.fu6</td>
</tr>
</tbody>
</table>
9.2.7. Test run ch4ohtr2

Tunneling: none
Number of levels: single level
Hindered rotation: none

This is the second run in the four-stage single-level calculation for the \( \text{CH}_4 + \text{OH} \rightarrow \text{CH}_3 + \text{H}_2\text{O} \) 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, CH\(_3\) and H\(_2\)O, 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 GAUSSRATE 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.

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch4ohtr2.70</td>
<td>esp.fu70</td>
</tr>
<tr>
<td>ch4ohtr2.72</td>
<td>esp.fu72</td>
</tr>
<tr>
<td>ch4ohtr2.73</td>
<td>esp.fu73</td>
</tr>
<tr>
<td>ch4ohtr2.74</td>
<td>esp.fu74</td>
</tr>
<tr>
<td>ch4ohtr2.dat</td>
<td>poly.fu5</td>
</tr>
<tr>
<td>ch4ohtr2.61</td>
<td>esp.fu61</td>
</tr>
<tr>
<td>ch4ohtr2.fu6</td>
<td>poly.fu6</td>
</tr>
</tbody>
</table>
9.2.8. Test run ch4ohtr3

Tunneling: none
Number of levels: single level
Hindered rotation: none

This is the third run in the four-stage single-level calculation for the CH$_4$ + OH $\rightarrow$ CH$_3$ + H$_2$O 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.

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch4ohtr3.70</td>
<td>esp.fu70</td>
</tr>
<tr>
<td>ch4ohtr3.75</td>
<td>esp.fu75</td>
</tr>
<tr>
<td>ch4ohtr3.dat</td>
<td>poly.fu5</td>
</tr>
<tr>
<td>ch4ohtr3.61</td>
<td>esp.fu61</td>
</tr>
<tr>
<td>ch4ohtr3.fu6</td>
<td>poly.fu6</td>
</tr>
</tbody>
</table>
9.2.9. Test run ch4ohtr4

Tunneling: SCT
Number of levels: single level
Hindered rotation: mode 14

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 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 GAUSSRATE restart option was selected in the fu70 input file, writing the information for future calculations in the fu84 file. It is interesting to note that, since no GAUSSIAN calculation is carried out for the saddle point in this particular run, the charge and multiplicity for the nonstationary points along the reaction path must be provided in the START section of the fu70 file (since the program assumes that these nonstationary points have the same charge and multiplicity as the saddle point).

The Hessians along the reaction path are calculated using the keywords (specified in GRSEC list keyword in the GRPATH section of the fu70 input file):

```
#PM3 OPT=(NOEIGENTEST,CALCFC) UNITS=AU FCHK NOSYMM OPTCYC=1
```

The reason for using these keywords, rather than FREQ as in the other test runs, is that when using the FREQ keyword for calculating the frequencies, some calculations of points along the reaction path performed with revision B.1 of GAUSSIAN 94 stop when the program reaches the matcor subroutine. The keywords given above make GAUSSIAN follow a different route, avoiding the matcor subroutine, ending successfully, and providing GAUSSRATE with the required information.
<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch4ohtr4.70</td>
<td>esp.fu70</td>
<td>Input data for GAUSSRATE</td>
</tr>
<tr>
<td>ch4ohtr4.dat</td>
<td>poly.fu5</td>
<td>Input data for POLYRATE</td>
</tr>
<tr>
<td>ch4ohtr4.61</td>
<td>esp.fu61</td>
<td>POLYRATE output file containing information about the stationary points</td>
</tr>
<tr>
<td>ch4ohtr4.fu6</td>
<td>poly.fu6</td>
<td>Long output file</td>
</tr>
<tr>
<td>ch4ohtr4.fu15</td>
<td>poly.fu15</td>
<td>Summary output file</td>
</tr>
<tr>
<td>ch4ohtr4.rst</td>
<td>esp.fu84</td>
<td>GAUSSRATE restart file generated by GAUSSRATE in this run</td>
</tr>
</tbody>
</table>
9.2.10. Test run ch4ohtr5

Tunneling: LCT
Number of levels: dual level
Hindered rotation: mode 14

This test run completes an IOC dual-level calculation of the \( \text{CH}_4 + \text{OH} \rightarrow \text{CH}_3 + \text{H}_2\text{O} \) 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 (\(\mu\)OMT), 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. The frequencies are calculated using Cartesian coordinates, instead of the set of redundant curvilinear coordinates from the ch4ohtr4 run. Since the different choice of coordinates does not affect the reaction path calculation, the restart option does not prevent us from changing the coordinate system from one run to another. The hindered rotor approximation is used for computing the partition function of the lowest vibrational mode. The lowest four frequencies are interpolated directly by the IVTST–0 interpolation algorithm.

The restart file to be read is copied from the ch4ohtr4.rst file, generated by the previous run, ch4ohtr4. The user should, therefore, run the previous run (ch4ohtr4) in this set of tests before running this one. A new restart file, which includes all the previous information plus the additional calculations needed for estimating the LCT tunneling, is generated by this test run.

Since GAUSSRATE tries to minimize the number of calculations spooled to GAUSSIAN, any calculation involves several checking steps, which can actually slow the GAUSSRATE calculation when the GAUSSIAN jobs are as fast as in this set of test runs. This fact, added to the large requirements of input-output data transfer, make this calculation (and, in general, any calculation based on semiempirical methods) much slower when using GAUSSRATE than when using MORATE.
<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch4ohtr5.70</td>
<td>esp.fu70</td>
<td>Input data for GAUSSRATE</td>
</tr>
<tr>
<td>ch4ohtr4.rst</td>
<td>esp.fu83</td>
<td>Restart file read by GAUSSRATE</td>
</tr>
<tr>
<td>ch4ohtr5.dat</td>
<td>poly.fu5</td>
<td>Input data for POLYRATE</td>
</tr>
<tr>
<td>ch4ohtr5.50</td>
<td>poly.fu50</td>
<td>Additional input data for the POLYRATE VTST-IC calculation</td>
</tr>
<tr>
<td>ch4ohtr5.61</td>
<td>esp.fu61</td>
<td>POLYRATE output file containing information about the stationary points</td>
</tr>
<tr>
<td>ch4ohtr5.fu6</td>
<td>poly.fu6</td>
<td>Long output file</td>
</tr>
<tr>
<td>ch4ohtr5.fu15</td>
<td>poly.fu15</td>
<td>Summary output file</td>
</tr>
<tr>
<td>ch4ohtr5.rst</td>
<td>esp.fu84</td>
<td>GAUSSRATE restart file generated by GAUSSRATE in this run</td>
</tr>
</tbody>
</table>
9.2.11. Test run ch4ohtr6

Tunneling: LCT
Number of levels: dual level
Hindered rotation: mode 14

This test run completes a IOC dual-level calculation of the CH$_4$ + OH $\rightarrow$ CH$_3$ + H$_2$O reaction. The lower-level information is read from the restart file from the ch4ohtr4 run. The higher-level data is included by means of the fu50 file. Tunneling is calculated by the microcanonical optimized multidimensional tunneling method (muOMT), with LCT method being calculated using the LCG4 approach. The frequencies are calculated using Cartesian coordinates, instead of the set of redundant curvilinear coordinates from the ch4ohtr4 run. Since the different choice of coordinates does not affect the reaction path calculation, the restart option does not prevent us from changing the coordinate system from one run to another. The hindered rotor approximation is used for computing the partition function of the lowest vibrational mode. The lowest four frequencies are interpolated directly by the IVTST-0 interpolation algorithm.

The restart file to be read is copied from the ch4ohtr4.rst file, generated by the previous run, ch4ohtr4. The user should, therefore, run the previous run (ch4ohtr4) in this set of tests before running this one. A new restart file, which includes all the previous information plus the additional calculations needed for estimating the LCT tunneling, is generated by this test run.

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ch4ohtr6.70</td>
<td>esp.fu70</td>
</tr>
<tr>
<td>ch4ohtr4.rst</td>
<td>esp.fu83</td>
</tr>
<tr>
<td>ch4ohtr6.dat</td>
<td>poly.fu5</td>
</tr>
<tr>
<td>ch4ohtr6.50</td>
<td>poly.fu50</td>
</tr>
<tr>
<td>ch4ohtr6.61</td>
<td>esp.fu61</td>
</tr>
<tr>
<td>ch4ohtr6.fu6</td>
<td>poly.fu6</td>
</tr>
<tr>
<td>ch4ohtr6.fu15</td>
<td>poly.fu15</td>
</tr>
<tr>
<td>ch4ohtr6.rst</td>
<td>esp.fu84</td>
</tr>
</tbody>
</table>
9.2.12. Test run nh3ohtr1

Tunneling: none
Number of levels: single level
Hindered rotation: none

This is a test run for the NH$_3$ + OH → NH$_2$ + H$_2$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 0 in the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections of the fu5 input file. The Hessian matrices are also calculated, as recommended, using the methods in GAUSSIAN. With the information for the stationary points obtained in this way, a conventional TST rate constant without tunneling is calculated.

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
</tr>
</thead>
<tbody>
<tr>
<td>nh3ohtr1.70</td>
<td>esp.fu70</td>
</tr>
<tr>
<td>nh3ohtr1.71</td>
<td>esp.fu71</td>
</tr>
<tr>
<td>nh3ohtr1.72</td>
<td>esp.fu72</td>
</tr>
<tr>
<td>nh3ohtr1.73</td>
<td>esp.fu73</td>
</tr>
<tr>
<td>nh3ohtr1.74</td>
<td>esp.fu74</td>
</tr>
<tr>
<td>nh3ohtr1.75</td>
<td>esp.fu75</td>
</tr>
<tr>
<td>nh3ohtr1.77</td>
<td>esp.fu77</td>
</tr>
<tr>
<td>nh3ohtr1.78</td>
<td>esp.fu78</td>
</tr>
<tr>
<td>nh3ohtr1.dat</td>
<td>poly.fu5</td>
</tr>
<tr>
<td>nh3ohtr1.61</td>
<td>esp.fu61</td>
</tr>
<tr>
<td>nh3ohtr1.fu6</td>
<td>poly.fu6</td>
</tr>
<tr>
<td>nh3ohtr1.fu15</td>
<td>poly.fu15</td>
</tr>
</tbody>
</table>
9.2.13. Test run nh3ohtr2

Tunneling: ZCT, SCT
Number of levels: single level
Hindered rotation: none

This is a test run for the NH\textsubscript{3} + OH \rightarrow NH\textsubscript{2} + H\textsubscript{2}O reaction including a well on the reactant side and a well on the product side using IVTST-M method. The geometries are read from \texttt{fu5} file with INITGEO set to \texttt{geom} and STATUS to 0 in the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections of the \texttt{fu5} input file. The potential energy surface is provided by \texttt{fu31} file. Rate constants are calculated using conventional TST, CVT, CVT/ZCT, and CVT/SCT methods.

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
</tr>
</thead>
<tbody>
<tr>
<td>nh3ohtr1.70</td>
<td>esp.fu70</td>
</tr>
<tr>
<td>nh3ohtr1.dat</td>
<td>poly.fu5</td>
</tr>
<tr>
<td>nh3ohtr1.fu31</td>
<td>poly.fu31</td>
</tr>
<tr>
<td>nh3ohtr1.61</td>
<td>esp.fu61</td>
</tr>
<tr>
<td>nh3ohtr1.fu6</td>
<td>poly.fu6</td>
</tr>
<tr>
<td>nh3ohtr1.fu15</td>
<td>poly.fu15</td>
</tr>
</tbody>
</table>
9.2.14. Test run h3tr1

Tunneling: none
Number of levels: single level
Hindered rotation: none

This is a test run for the $\text{H}_2 + \text{H} \rightarrow \text{H} + \text{H}_2$ reaction. 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, and INITGEO is set to hooks and STATUS to 0 in the REACT1, REACT2, PROD1, PROD2, and START sections of the fu5 input file. The Hessian matrices are also calculated, as recommended, using the methods in GAUSSIAN. With the information for the stationary points obtained in this way, a conventional TST rate constant without tunneling is calculated. This test run demonstrates that a linear system may be calculated using GAUSSRATE.

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>h3tr1.70</td>
<td>esp.fu70</td>
<td>Input data for GAUSSRATE</td>
</tr>
<tr>
<td>h3tr1.71</td>
<td>esp.fu71</td>
<td>Input data for reactant 1</td>
</tr>
<tr>
<td>h3tr1.72</td>
<td>esp.fu72</td>
<td>Input data for reactant 2</td>
</tr>
<tr>
<td>h3tr1.73</td>
<td>esp.fu73</td>
<td>Input data for product 1</td>
</tr>
<tr>
<td>h3tr1.74</td>
<td>esp.fu74</td>
<td>Input data for product 2</td>
</tr>
<tr>
<td>h3tr1.75</td>
<td>esp.fu75</td>
<td>Input data for the saddle point</td>
</tr>
<tr>
<td>h3tr1.dat</td>
<td>poly.fu5</td>
<td>Input data for POLYRATE</td>
</tr>
<tr>
<td>h3tr1.61</td>
<td>esp.fu61</td>
<td>POLYRATE output file containing information about the stationary points</td>
</tr>
<tr>
<td>h3tr1.fu6</td>
<td>poly.fu6</td>
<td>Long output file</td>
</tr>
<tr>
<td>h3tr1.fu15</td>
<td>poly.fu15</td>
<td>Summary output file</td>
</tr>
</tbody>
</table>
9.2.15. Test run h3tr2

Tunneling: none
Number of levels: single level
Hindered rotation: none

This is a test run for the $\text{H}_2 + \text{H} \rightarrow \text{H} + \text{H}_2$ reaction using the PCM solvation model. 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, and INITGEO is set to hooks and STATUS to 0 in the REACT1, REACT2, PROD1, PROD2, and START sections of the fu5 input file. The Hessian matrices are also calculated, as recommended, using the methods in GAUSSIAN. With the information for the stationary points obtained in this way, a conventional TST rate constant without tunneling is calculated. This test run demonstrates that a linear system may be calculated using the PCM model found in GAUSSRATE.

<table>
<thead>
<tr>
<th>GAUSSRATE I/O files</th>
<th>GAUSSRATE file names</th>
</tr>
</thead>
<tbody>
<tr>
<td>h3tr2.70</td>
<td>esp.fu70</td>
</tr>
<tr>
<td>h3tr2.71</td>
<td>esp.fu71</td>
</tr>
<tr>
<td>h3tr2.72</td>
<td>esp.fu72</td>
</tr>
<tr>
<td>h3tr2.73</td>
<td>esp.fu73</td>
</tr>
<tr>
<td>h3tr2.74</td>
<td>esp.fu74</td>
</tr>
<tr>
<td>h3tr2.75</td>
<td>esp.fu75</td>
</tr>
<tr>
<td>h3tr2.dat</td>
<td>poly.fu5</td>
</tr>
<tr>
<td>h3tr2.61</td>
<td>esp.fu61</td>
</tr>
<tr>
<td>h3tr2.fu6</td>
<td>poly.fu6</td>
</tr>
<tr>
<td>h3tr2.fu15</td>
<td>poly.fu15</td>
</tr>
</tbody>
</table>
9.2.16. Test run c2h6tr1

Tunneling: none
Number of levels: single level
Hindered rotation: none; spectator degrees of freedom treated by VRC-VTST for transitional modes.

This is a test run for the CH$_3$ + CH$_3$ → C$_2$H$_6$ reaction. This reaction occurs without a potential energy barrier. The calculation of rate constants for this reaction is carried out by using variable reaction coordinate variational transition state theory with multifaceted dividing surfaces. The rate constants are calculated by canonical variational theory, microcanonical variational theory, and total energy and angular momentum resolved microcanonical variational theory. Tunneling is not included.
10. COMPUTERS, OPERATING SYSTEMS, AND GAUSSIAN VERSIONS ON WHICH THE CODE HAS BEEN DEVELOPED AND TESTED

In each case we give the GAUSSRATE version number, then the computers and operating system on which GAUSSRATE was tested. For each computer and operating system we also specify the GAUSSIAN version that was used for testing.

<table>
<thead>
<tr>
<th>GAUSSRATE Version</th>
<th>Machine (Operating system)</th>
<th>GAUSSIAN Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1/P7.1.1-G94</td>
<td>IBM RS/6000 model 550 (AIX 4.1)</td>
<td>B.1</td>
</tr>
<tr>
<td></td>
<td>Cray C90 (UNICOS 8.0.3)</td>
<td>D.3</td>
</tr>
<tr>
<td></td>
<td>Silicon Graphics IRIS-Power Challenge L R8000 (IRIX 6.2)</td>
<td>B.1</td>
</tr>
<tr>
<td>7.2/P7.2-G94</td>
<td>IBM RS/6000 model 550 (AIX 4.1)</td>
<td>B.1</td>
</tr>
<tr>
<td></td>
<td>Cray C90 (UNICOS 8.0.3)</td>
<td>D.3</td>
</tr>
<tr>
<td></td>
<td>Silicon Graphics IRIS-Power Challenge L R8000 (IRIX 6.2)</td>
<td>B.1</td>
</tr>
<tr>
<td>7.3/P7.3-G94</td>
<td>IBM RS/6000 model 550 (AIX 4.2)</td>
<td>B.1</td>
</tr>
<tr>
<td></td>
<td>Cray C90 (UNICOS 8.0.3)</td>
<td>D.4</td>
</tr>
<tr>
<td></td>
<td>Silicon Graphics IRIS-Power Challenge L R8000 (IRIX 6.2)</td>
<td>B.1</td>
</tr>
<tr>
<td>7.4/P7.4-G94 – 7.4.1/P7.4.1-G94</td>
<td>IBM RS/6000 model 550 (AIX 4.2)</td>
<td>B.1</td>
</tr>
<tr>
<td></td>
<td>Cray C90 (UNICOS 8.0.3)</td>
<td>D.4,E.1</td>
</tr>
<tr>
<td></td>
<td>Silicon Graphics IRIS-Power Challenge L R8000 (IRIX 6.2)</td>
<td>B.1</td>
</tr>
<tr>
<td>7.8/P7.8-G94 – 7.9.1/P7.9.1-G94</td>
<td>IBM RS/6000 model 550 (AIX 4.2)</td>
<td>B.1</td>
</tr>
<tr>
<td></td>
<td>Cray C90 (UNICOS 8.0.3)</td>
<td>E.1</td>
</tr>
<tr>
<td></td>
<td>Silicon Graphics IRIS-Power Challenge L R8000 (IRIX 6.2)</td>
<td>D.4</td>
</tr>
<tr>
<td>GAUSSRATE Version</td>
<td>Machine (Operating system)</td>
<td>GAUSSIAN Version</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>8.0/P8.0-G94</td>
<td>IBM RS/6000 model 550 (AIX 4.2)</td>
<td>B.1</td>
</tr>
<tr>
<td></td>
<td>IBM RS/6000 43P model 240 (AIX 4.3)</td>
<td>D.2</td>
</tr>
<tr>
<td></td>
<td>Silicon Graphics IRIS-Power Challenge L R10000 (IRIX 6.2)</td>
<td>D.4</td>
</tr>
<tr>
<td>8.1.1/P8.1-G94G98</td>
<td>IBM SP with 604e processor (AIX 4.3.1)</td>
<td>D.2 A.7</td>
</tr>
<tr>
<td></td>
<td>SGI-Origin 2000 with R10000 processor (IRIX 6.5)</td>
<td>D.4 A.7</td>
</tr>
<tr>
<td>8.6/P8.5.1-G94G98</td>
<td>SGI-Origin 2000 with R10000 processor (IRIX 6.5)</td>
<td>D.4</td>
</tr>
<tr>
<td>8.7/P8.7-G94G98</td>
<td>IBM SP with Power3 processor (AIX 4.3.4.0)</td>
<td>A.11</td>
</tr>
<tr>
<td></td>
<td>SGI Origin 2000 with R12000 processor (IRIX 6.5.12f)</td>
<td>A.11</td>
</tr>
<tr>
<td>8.9/P8.9-G94G98–9.0/P9.0-G94G98</td>
<td>IBM SP with Power3 processor (AIX 4.3.4.0)</td>
<td>A.11</td>
</tr>
<tr>
<td></td>
<td>SGI-Origin 2000 with R12000 processor (IRIX 6.5.12f)</td>
<td>E.2 A.11</td>
</tr>
<tr>
<td>9.1/P9.1-G03G98G94</td>
<td>IBM SP with Power3 processor (AIX 4.3.4.0)</td>
<td>A.11 B.01</td>
</tr>
<tr>
<td></td>
<td>SGI-Origin 2000 with R12000 processor (IRIX 6.5.12f)</td>
<td>E.2 A.11 B.01</td>
</tr>
<tr>
<td></td>
<td>Netfinity Linux Cluster with Pentium III (RedHat Linux 7.2)</td>
<td>A.11.3 B.01</td>
</tr>
<tr>
<td></td>
<td>SunBlade 2000 with UltraSparcIII processor (Solaris 8)</td>
<td>A.11 B.01</td>
</tr>
<tr>
<td>9.5/P9.5-G03G98G94</td>
<td>IBM Regatta with Power4 processor (AIX 5.2)</td>
<td>A.11 D.01</td>
</tr>
<tr>
<td></td>
<td>SGI Altix with Itanium 2 processor (SuSE Linux)</td>
<td>A.11 D.01</td>
</tr>
<tr>
<td></td>
<td>IBM Netfinity with Pentium III (RedHat Enterprise Linux 3)</td>
<td>A.11.3 D.01</td>
</tr>
<tr>
<td>9.7/P9.7-G03G98G94</td>
<td>IBM Regatta with Power4 processor (AIX 5.3)</td>
<td>A.11 D.01</td>
</tr>
<tr>
<td>Machine (Operating system)</td>
<td>GAUSSIAN Version</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------</td>
<td></td>
</tr>
<tr>
<td>SGI Altix with Itanium 2 processor (SuSE Linux)</td>
<td>A.11 D.01</td>
<td></td>
</tr>
<tr>
<td>IBM Netfinity with Pentium III (RedHat Enterprise Linux 3)</td>
<td>A.11.3 D.01</td>
<td></td>
</tr>
<tr>
<td>IBM BladeCenter with AMD Opteron processor (SuSE Linux)</td>
<td>D.01</td>
<td></td>
</tr>
</tbody>
</table>

**2008/P2008-G03G98G94**

<table>
<thead>
<tr>
<th>Machine (Operating system)</th>
<th>GAUSSIAN Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGI Altix with Itanium 2 processor (SuSE Linux)</td>
<td>A.11 E.01</td>
</tr>
<tr>
<td>IBM BladeCenter with AMD Opteron processor (SuSE Linux)</td>
<td>E.01</td>
</tr>
<tr>
<td>SGI Altix XE 1300 with 2.66 GHz Intel Xeon processor (SuSE Linux)</td>
<td>E.01</td>
</tr>
</tbody>
</table>

**2009/P2008-G09G03G98G94**

<table>
<thead>
<tr>
<th>Machine (Operating system)</th>
<th>GAUSSIAN Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGI Altix with Itanium 2 processor (SuSE Linux)</td>
<td>E.01 A.02</td>
</tr>
<tr>
<td>IBM BladeCenter with AMD Opteron processor (SuSE Linux)</td>
<td>E.01 A.02</td>
</tr>
<tr>
<td>SGI Altix XE 1300 with 2.66 GHz Intel Xeon processor (SuSE Linux)</td>
<td>E.01 A.02</td>
</tr>
</tbody>
</table>

**2009-A/P2008-G09G03G98G94**

<table>
<thead>
<tr>
<th>Machine (Operating system)</th>
<th>GAUSSIAN Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGI Altix with Itanium 2 processor (SuSE Linux)</td>
<td>E.01 A.02</td>
</tr>
<tr>
<td>IBM BladeCenter with AMD Opteron processor (SuSE Linux)</td>
<td>E.01 A.02</td>
</tr>
<tr>
<td>SGI Altix XE 1300 with 2.66 GHz Intel Xeon processor (SuSE Linux)</td>
<td>E.01 A.02</td>
</tr>
</tbody>
</table>

**2015/P2015-G09G03G98G94**

<table>
<thead>
<tr>
<th>Machine (Operating system)</th>
<th>GAUSSIAN Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP Linux cluster with Intel Xeon X5560 &quot;Nehalem EP&quot; processors (CentOS Linux)</td>
<td>D.02</td>
</tr>
</tbody>
</table>
11. VERSION INFORMATION

11.1. GAUSSRATE revision numbers

GAUSSRATE version numbers have the general form v/Pv'-Gv" where v is the overall version number, v' is the number of the version of POLYRATE on which it is based, and v" is the version of GAUSSIAN on which it is based. The part before the solidus, i.e., v, is unique, that is, it changes if the interface changes, or if the POLYRATE version changes, or if the GAUSSIAN version changes. Thus, the part after the solidus is redundant information, and it may be omitted when brevity is desirable. For example, the present version of the code may be referred to simply as GAUSSRATE 2015 or GAUSSRATE-version 2015.

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.
11.2. GAUSSRATE revision history

This section contains the revision history subsequent to version 7.1 (March 1997), which was the first version of the code.

Version 7.1.1/P7.1.1-G94 (March 1997)

1. A bug which caused the energy of the products (and therefore of the reverse rate constant) to be miscalculated for reactions with one product when the POLYRATE 7.1.1 methods were selected for the Hessian calculation optimization, and the GAUSSIAN 94 methods were selected for geometry optimizations has been fixed.

2. A bug which caused some restart calculations to abort when an atomic species is involved has been fixed.

Version 7.2/P7.2-G94 (April 1997)

1. The test runs have been modified to take account of changes in the new version of POLYRATE. This will lead to incompatibilities between the fu5 input files for GAUSSRATE 7.1 and the fu5 files for GAUSSRATE 7.2 if certain options are used. The user is advised to check Section 23.E of the POLYRATE 7.2 manual in order to modify the fu5 input file from previous calculations. No incompatibilities are present in the fu31 through fu35 input files. The interface has not changed.

2. The compiling scripts have been modified in order to make the compilation cleaner and safer.

Version 7.3/P7.3-G94 (July 1997)

1. This version is based on a new version of POLYRATE. The test runs have been changed in order to use the new set of keywords and options. This will lead to incompatibilities between the fu5 input files for previous versions of the program and the fu5 files for GAUSSRATE 7.3 if certain options are used. The user is advised to check the Section 23.E of the POLYRATE 7.3 manual in order to modify the fu5 input file from previous calculations.
2. The GAUSSRATE files have been renamed from gr.# to esp.#, for the sake of consistency with the POLYRATE and MORATE programs. Moreover, the numbering of these FORTRAN units was updated according to the new conventions established by POLYRATE 7.3; in particular, the files are assigned to FORTRAN units 71-76 instead of 31-36.

3. The input file fu70 is now a case sensitive file. Hence, when typing UNIX path names, a capital letter in the path name must be entered as a capital letter, and a lower case letter must be entered as lower case.

**Version 7.4/P7.4-G94 (August 1997)**

1. This version is based on a new version of POLYRATE. The test runs have been modified in order to interface with POLYRATE 7.4, using the new keywords and options. These changes in POLYRATE will lead to incompatibilities between the fu5 input files for previous versions of the GAUSSRATE program and the fu5 files for GAUSSRATE 7.4 if certain options are used. The user is advised to check Section 23.E of the POLYRATE 7.4 manual in order to see whether it is necessary to modify the fu5 input file from previous calculations.

2. Four new test runs for the CH₄OH system were added in order to illustrate the use of the STATUS keyword. In addition, some test runs were modified in order to apply the new POLYRATE capability for calculating vibrational frequencies by using redundant internal coordinates.

3. The code has been changed in order to interface with version E.1 of GAUSSIAN 94. Minor changes in the formatted checkpoint file written by GAUSSIAN make previous versions of GAUSSRATE incompatible with the E.1 revision of GAUSSIAN 94.

**Version 7.4.1/P7.4.1-G94 (September 1997)**

1. This version is based on a new version of POLYRATE. The interface between POLYRATE and GAUSSIAN has not changed. Please review the POLYRATE 7.4.1 revision history for details.
Version 7.8/P7.8-G94 (December 1997)

1. This version is based on a new version of POLYRATE. The user is advised to review section 23.D of the POLYRATE 7.8 manual to determine the possible incompatibilities in fu5 input files from previous POLYRATE versions.

2. This update also involves the possibility of including wells along the reaction path. With this aim, files esp.fu77 and esp.fu78 have been included in the program.

3. The keywords GROPT and NOGROPT have been removed. The code now determines the optimization options automatically from the information in the fu5 input file. This leads to incompatibilities with esp.fu70 input files from older versions of the program.

4. A new test, nh3ohtr1, has been added to illustrate the use of the files esp.fu77 and esp.fu78 for calculating wells on the reaction path. Moreover, the test run oh3tr2 has been modified in order to provide an example of the use of the RODS method.

5. Several minor modifications in the test runs have been introduced to reduce the timings.

Version 7.8.1/P7.8.1-G94 (February 1998)

1. This version is based on a new version of POLYRATE. The interface between POLYRATE and GAUSSIAN has not changed. Please review the POLYRATE 7.8.1 revision history for details.

Version 7.9/P7.9-G94 (March 1998)

1. This version is based on a new version of POLYRATE. The interface between POLYRATE and GAUSSIAN has not changed. The user is advised to review section 23.D of the POLYRATE 7.9 manual to determine the possible incompatibilities in fu5 input files from previous POLYRATE versions.
Version 7.9.1/P7.9.1-G94 (April 1998)

1. This version is based on a new version of POLYRATE. The interface between POLYRATE and GAUSSIAN has not changed. Please review the POLYRATE 7.9.1 revision history for details.

Version 7.9.2/P7.9.2-G94 (June 1998)

1. A bug in the print out of the gradients of some stationary points after their optimization using GAUSSIAN 94 methods has been fixed. This bug fix involved changes in the ohook subroutine.

Version 8.0/P8.0-G94 (September 1998)

1. This version is based on a new version of POLYRATE. The FORTRAN source for the interface routines has not changed. Please review the POLYRATE 8.0 revision history for details. The test run input files have been updated to interface with POLYRATE 8.0. A new test run, ch5tr2, has been added to test the new ISPE method; and test run oh3tr2 has been modified in order to use the new VRPE option for following the reaction path, allowing a larger step size.

2. A Korn shell script, check, has been included in the testrun directory to automatically compare the fu15 output files obtained after running the test suite with those in the testo directory.

3. The structure of the distribution files and the compiling scripts have been modified in an attempt to make the compilation more straightforward. In addition, a compiling script for IBM computers with the 604e processor has been added.

Version 8.1.1/P8.1.1-G94G98 (July 1999)

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

2. Modifications in the subroutine goutsec are added to make the interface work with a new version of GAUSSIAN (in particular, for GAUSSIAN 98 Rev. A.7). In this case, the
user only needs to indicate the directory of the GAUSSIAN 98 executable in the script shuttle.

3. The compilation script was modified to O3 with 64 bits for the IBM SP 604e processor.

Version 8.1.2/P8.1.2-G94G98 (August 1999)

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

Version 8.2/P8.2-G94G98 (August 1999)

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

Version 8.3/P8.3-G94G98 (December 1999)

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

Version 8.4/P8.4-G94G98 (August 2000)

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

2. A new keyword, RSTTOL, in the GENERAL section of input file poly.fu70 has been added.

3. Modifications in the test suite were added in order to make all the test runs to run in a subdirectory of each test directory, so that the calculations are performed in a more clear fashion and the possibility of interfering with other GAUSSIAN or GAUSSRATE runs were minimized.

4. A new test run, ch4ohtr6, is added to test the LCG4 method for LCT tunneling.

Version 8.5/P8.5-G94G98 (October 2000)

1. This version is based on a new version of POLYRATE.
Version 8.6/P8.5.1-G94G98 (November 2000)

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

Version 8.7/P8.7-G94G98 (October 2001)

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

Version 8.9/P8.9-G94G98 (May 2002)

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

2. The Korn shell script, check, has been renamed to check94. A new Korn shell script, check98, has been added to compare the fu15 output files obtained after running the test suite with those in the testo/g98 directory.

3. The SGI compilation script was modified to turn off some warnings during compilation.

Version 9.0/P9.0-G94G98 (May 2002)

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

Version 9.1/P9.1-G03G98G94 (July 2003)

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

2. This version allows interfacing POLYRATE with the GAUSSIAN 03 as well as GAUSSIAN 98 and GAUSSIAN 94. This new feature required modifications in the shuttle script and upgrading the headr subroutine. The distributed output files obtained by running GAUSSRATE with GAUSSIAN 03 are located in the new directory testo/g03. A new Korn shell script, check03, has been added to compare the fu15 output files obtained after running the test suite with those in the testo/g03 directory.

3. Test runs ch4ohtr4, ch4ohtr5, and ch4ohtr6 were updated to include the new SPECSTOP keyword of POLYRATE.
4. The GAUSSRATE manual is now distributed as Word and pdf files instead of a txt file.

5. A few old variables that were not used anymore have been removed from the code.

6. The Cray platform is no longer supported. The hooks_gr.cray and gausrate.cray files were removed.

7. The Sun and Linux platforms were added to the supported platforms. The gausrate.linux and gausrate.sun compiling scripts have been added in the script directory.

Version 9.4/P9.4-G03G98G94 (June 2006)

1. This version is based on a new version of POLYRATE. The code has been modified to accept the MPI processor number from POLYRATE, however, it does not yet take advantage of this information.

2. This version edits the POLYRATE Makefile for easy installation.

3. The code has been modified to work on systems that do not have ‘.’ in $PATH.

Version 9.5/P9.5-G03G98G94 (January 2007)

1. The H3 test runs have been added to demonstrate compatibility of linear systems and with the PCM solvation model.

Version 9.7/P9.7-G03G98G94 (June 2007)

1. This version is based on a new version of POLYRATE. The code has been modified to be compatible with the new keywords INPUNIT and OUTUNIT in POLYRATE version 9.7.

2. Most input files of test runs are modified by adding “INPUNIT AU” keyword to be compatible with POLYRATE version 9.7. Two test runs do not use this keyword, and these test runs use angstroms for input.
Version 2008/P2008-G03G98G94 (May 2009)
1. This version is based on a new version of POLYRATE. The code has been modified to be compatible with the POLYRATE 2008. This allows one, for example, to now carry out VRC-VTST calculations in parallel using MPI. To illustrate the new capability, we add a test run for methyl radical association.

Version 2009/P2008-G09G03G98G94 (December 2009)
1. The subroutine `goutsec` is modified so that it can read the Test.fchk file generated by either G03 or G09. This modification make the subroutine `goutsec` read the geometry, energy, gradient, and Hessian independent of their order stored in the Test.fchk file.

Version 2009-A/P2008-G09G03G98G94 (March 2010)
1. A bug that causes WELLR geometry is overwritten by WELLP geometry is fixed in `prep` subroutine in hooks_gr.F file.

Version 2015/P2015-G09G03G98G94 (January 2015)
1. The common blocks are replaced by modules in order to be compatible with the POLYRATE version 2015.
12. ACKNOWLEDGMENTS

The authors are grateful to Patton Fast for assistance in preparing GAUSSRATE and its documentation and to Titus V. Albu for help in updating GAUSSRATE and its documentation. The development of GAUSSRATE has been supported in part by grants from the U.S. Department of Energy, Office of Basic Energy Sciences.