Manual
NWCHEMRATE – version 2007

Jingjing Zheng, Mark A. Iron, Benjamin A. Ellingson, José C. Corchado, Yao-Yuan Chuang, and Donald G. Truhlar
Department of Chemistry and Supercomputing Institute
University of Minnesota, Minneapolis, Minnesota 55455

Program version: 2007/P9.7-NW5.0
Program version date: December 26, 2007
Manual version date: December 26, 2007

Abstract:

NWCHEMRATE is a set of FORTRAN subroutines and Unix scripts for interfacing the POLYRATE and NWCHEM 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 NWCHEM 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 POLYRATE for canonical or microcanonical ensembles or for specific vibrational states of selected modes with translational, rotational, and other vibrational modes treated thermally. Bimolecular and unimolecular reactions 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 density functionals in the NWCHEM package or with the MP2 *ab initio* method. In dual-level mode, the lower-level data is calculated by NWCHEM, and the higher-level data is read in from an external file.

NWCHEMRATE–version 2007 provides an interface between two programs:

POLYRATE–version 9.7, by J. C. Corchado, Y. -Y. Chuang, P. L. Fast, W. -P. Hu, Y. -P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, A. Fernandez-Ramos, B. A. Ellingson, B. J. Lynch, J. Zheng, V. S. Melissas, J. Villà, I. Rossi, E. L. Coitiño, J. Pu, T. V. Albu, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar, University of Minnesota, Minneapolis, MN, 2007.

and

E. J. Bylaska, W. A. de Jong, K. Kowalski, T. P. Straatsma, M. Valiev, D. Wang, E. Aprà, T. L. Windus, S. Hirata, M. T. Hackler, Y. Zhao, P.-D. Fan, R. J. Harrison, M. Dupuis, D. M. A. Smith, J. Nieplocha, V. Tipparaju, M. Krishnan, A. A. Auer, M. Nooijen, E. Brown, G. Cisneros, G. I. Fann, H. Früchtl, J. Garza, K. Hirao, R. Kendall, J. A. Nichols, K. Tsemekhman, K. Wolinski, J. Anchell, D. Bernholdt, P. Borowski, T. Clark, D. Clerc, H. Dachsel, M. Deegan, K. Dyall, D. Elwood, E. Glendening, M. Gutowski, A. Hess, J. Jaffe, B. Johnson, J. Ju, R. Kobayashi, R. Kutteh, Z. Lin, R. Littlefield, X. Long, B. Meng, T. Nakajima, S. Niu, L. Pollack, M. Rosing, G. Sandrone, M. Stave, H. Taylor, G. Thomas, J. van Lenthe, A. Wong, and Z. Zhang, *NWChem, A Computational Chemistry Package for Parallel Computers, Version 5.0* (2006), Pacific Northwest National Laboratory, Richland, Washington 99352-0999, USA.

High Performance Computational Chemistry: An Overview of NWChem A Distributed Parallel Application, Kendall, R.A.; Aprà, E.; Bernholdt, D.E.; Bylaska, E.J.; Dupuis, M.; Fann, G.I.; Harrison, R.J.; Ju, J.; Nichols, J.A.; Nieplocha, J.; Straatsma, T.P.; Windus, T.L.; Wong, A.T. Computer Phys. Comm., 2000, **128**, 260-283 .

Table of Contents

1. <u>USER AGREEMENT</u>

NWCHEMRATE 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 NWCHEMRATE software or in any copyrights for the NWCHEMRATE 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 licenses are also available.

B. Publications resulting from using this package or the POLYRATE or NWCHEM 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 NWCHEMRATE).

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 and NWCHEM codes, required to use NWCHEMRATE, are covered by separate licenses.

2. REQUIRED REFERENCES

Publications based on results obtained with this computer code should include the following references. (Additional references for specific methods used are given in the documentation for POLYRATE and NWCHEM.)

1. J. Zheng, M. A. Iron, B. A. Ellingson, J. C. Corchado, Y.-Y. Chuang, E. L. Coitiño, and D. G. Truhlar, NWCHEMRATE–version 2007, University of Minnesota, Minneapolis, MN, 2007.

2. J. C. Corchado, Y.-Y. Chuang, P. L. Fast, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, A. Fernandez-Ramos, B. A. Ellingson, B. J. Lynch, J. Zheng, V. S. Melissas, J. Villà, I. Rossi, E. L. Coitiño, J. Pu, T. V. Albu, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar, POLYRATE–version 9.7, University of Minnesota, Minneapolis, MN, 2007.

and either 3A or 3B (or both)

3A. E. J. Bylaska, W. A. de Jong, K. Kowalski, T. P. Straatsma, M. Valiev, D. Wang, E. Aprà, T. L. Windus, S. Hirata, M. T. Hackler, Y. Zhao, P.-D. Fan, R. J. Harrison, M. Dupuis, D. M. A. Smith, J. Nieplocha, V. Tipparaju, M. Krishnan, A. A. Auer, M. Nooijen, E. Brown, G. Cisneros, G. I. Fann, H. Früchtl, J. Garza, K. Hirao, R. Kendall, J. A. Nichols, K. Tsemekhman, K. Wolinski, J. Anchell, D. Bernholdt, P. Borowski, T. Clark, D. Clerc, H. Dachsel, M. Deegan, K. Dyall, D. Elwood, E. Glendening, M. Gutowski, A. Hess, J. Jaffe, B. Johnson, J. Ju, R. Kobayashi, R. Kutteh, Z. Lin, R. Littlefield, X. Long, B. Meng, T. Nakajima, S. Niu, L. Pollack, M. Rosing, G. Sandrone, M. Stave, H. Taylor, G. Thomas, J. van Lenthe, A. Wong, and Z. Zhang, *NWChem, A Computational Chemistry Package for Parallel Computers, Version 5.0* (2006), Pacific Northwest National Laboratory, Richland, Washington 99352-0999, USA.

3B. *High Performance Computational Chemistry: an Overview of NWChem a Distributed Parallel Application*, Kendall, R.A.; Aprà, E.; Bernholdt, D.E.; Bylaska, E.J.; Dupuis, M.; Fann, G.I.; Harrison, R.J.; Ju, J.; Nichols, J.A.; Nieplocha, J.; Straatsma, T.P.; Windus, T.L.; Wong, A.T. Computer Phys. Comm., 2000, **128**, 260-283 .

3. INTRODUCTION

NWCHEMRATE 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 NWCHEM package to represent the potential energy of interaction and its first and second derivatives whenever they are needed. This version, 2007/P9.7-NW5.0, interfaces POLYRATE–version 9.7, which is a program for dynamical rate calculations, with the electronic structure program NWCHEM (any version, although version 5.0 or later is recommended).

Using NWCHEMRATE presupposes that the user has both POLYRATE 9.7 and NWCHEM (version 5.0 preferred, but any version should suffice) on the same computer working properly before attempting to run NWCHEMRATE. No part of POLYRATE 9.7 or NWCHEM is contained in NWCHEMRATE or needs to be modified for running this program. Likewise, separate licenses for these codes need to be obtained before using NWCHEMRATE.

This NWCHEMRATE manual covers only material that is not covered in the POLYRATE and NWCHEM manuals. The user of NWCHEMRATE should first be familiar with these programs.

The interface between POLYRATE and NWCHEM is made by a series of FORTRAN subroutines included in the files `hooks_nwr.f` and `hooks_mdepnw.f`, which are specialized versions of the routines in the `hooks.f` file in POLYRATE 9.7, and by a C shell script for running NWCHEM. This C shell script, named `taxi` (equivalent to `shuttle` in GAUSSRATE – the name was changed to avoid confusion between the two codes), is called from inside NWCHEMRATE, and it has to be able to start running NWCHEM using an input file named as the first argument given to `taxi`. It produces an output file named as the second argument given to `taxi`. The third argument passed to `taxi` is the number of processors on which NWCHEM will be run. For the user's convenience, a well documented script is provided, and in most cases the user will only need to change one line: set the *`nwpath`* variable to the appropriate path where NWCHEM is located. In the provided script, the variable *`scratchdir`*, which indicates the directory where the temporary and scratch NWCHEM files are going to be located, is set to the working directory; if desired, the user can also change this.

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

NWCHEMRATE makes use of *system* calls in order to have NWCHEM perform the needed energy, gradient and/or Hessian calculations. The capability to make system call is available in most FORTRAN compilers. Any user of NWCHEMRATE should make sure that this utility subroutine is available with their compiler, and, if necessary, install any missing libraries.

4. DISTRIBUTION

NWCHEMRATE is a set of subroutines for interfacing POLYRATE and NWCHEM. 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`

– NWCHEM: NWCHEM is a licensed code available from Pacific Northwest National Laboratories. For further information, see
`http://www.emsl.pnl.gov/docs/nwchem/nwchem_main.html`

– NWCHEMRATE: The NWCHEMRATE 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/nwchemrate`

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

Prospective users of NWCHEMRATE must first obtain and install POLYRATE and NWCHEM before proceeding with NWCHEMRATE.

The NWCHEMRATE program is distributed as a compressed tar file named either `nwrate2007.tar.Z` or `nwrate2007.tar.gz`. The distributed file (either one of the above files) should be uncompressed and untarred in the `polyrate9.7` directory. To uncompress, type:

```
uncompress nwrate2007.tar.Z
```

or

```
gunzip nwrate2007.tar.gz
```

The uncompressed file will be named `nwrate2007.tar`. After extracting the files from the tar file, which can be done with the command:

```
tar -xvf nwrate2007.tar
```

a new directory, `nwrate`, 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:

```
              polyrate9.7
                |
 ----------------------------------------------------
 |   |    |     |    |     |      |   |      |     |
doc exe nwrate obj poten script src testo testrun util
          |
    ------------------------------------------
    |      |    |            |           |
    doc script source      testo      testrun
                                         |
                           ------------------
                           |    |    |   |
                          ch4oh ch5 oh3 nh3oh
```

The files will be distributed among the directories as follows:

`nwrate`: the C shell script `taxi` and five subdirectories:

`doc`: NWCHEMRATE manual: `nwchemrate.pdf`.

`script`: 1 file: `nwrcompile`

`source`: 4 files comprising the source code for NWCHEMRATE: `nwrate.inc`, `head_nwr.f`, `hooks_nwr.f`, and `hooks_nwr.wks.f`

`testo`: This directory contains selected output files of the test runs:
-15 files: `ch4ohtr1.fu6`, `ch4ohtr2.fu6`, `ch4ohtr3.fu15`, `ch4ohtr4.fu15`, `ch4ohtr5.fu15`, `ch4ohtr6.fu15`, `ch5tr1.fu15`, `ch5tr2.fu15`, `ch5tr3.fu15`,

ch5tr4.fu15, oh3tr1.fu15, oh3tr2.fu15, oh3tr3.fu15, oh3tr4.fu15, nh3ohtr1.fu15

testrun: 2 files: checknwr and run_all.jc, and 4 subdirectories: ch4oh, ch5, oh3, and nh3oh with the input files for the test calculations.

5. INSTALLATION

The user should first obtain and install licensed copies of the POLYRATE and NWCHEM packages. Next, the user should obtain a licensed copy of the NWCHEMRATE package. After downloading, uncompressing, and untarring the file (as described in Section 4), the NWCHEMRATE package should appear as a subdirectory of `polyrate9.7`.

There are a few parameters, such as NATOMS (maximum number of atoms), NSDM (maximum number of saved points along the minimum energy path), and so forth, 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 NWCHEMRATE program is carried out by running a C shell script, `configure`. The script will copy the file `hooks_nwr.wks.f` to the `hooks_mdepnw.f` file (the `nw` was appended to the file name to avoid potential conflicts with GAUSSRATE), will compile the POLYRATE 9.7 files substituting the `hooks.f` file of standard POLYRATE by `hooks_nwr.f` and `hooks_mdepnw.f`, and will create the executable file `nwrate.exe` in the `exe` subdirectory of the `polyrate9.7` directory. The executable `nwrate.exe` will then be accessible for all the NWCHEMRATE calculations.

In Section 9, the computers and operating systems are listed on which the code has been tested. The installation of NWCHEMRATE on other platforms should be straightforward, requiring only changes in the compiler and loader options in any one of the compilation scripts.

Before running NWCHEMRATE, one line of the C shell script `taxi` should be changed as described in Section 1: The user needs to set the *`nwpath`* variable to the appropriate path where NWCHEM is located. Likewise, one can also change the *scratchdir* from the working directory to one's own scratch space.

6. PROGRAM DESCRIPTION

NWCHEMRATE program performs direct dynamics calculations by variational transition state theory with multidimensional tunneling contributions (VTST/MT). Full details of VTST/MT are given in the POLYRATE manual and in "Variational Transition State Theory with Multidimensional Tunneling," A. Fernandez-Ramos, B. A. Ellingson, B. C. Garrett, and D. G. Truhlar, in *Reviews in Computational Chemistry*, Vol. 23, edited by K. B. Lipkowitz and T. R. Cundari (Wiley-VCH, Hoboken, NJ, 2007), pp. 125-232. The potential energy surface is obtained directly by electronic structure calculations employing density functional theory (DFT) or Møller-Plesset second order perturbation theory (MP2)

The NWCHEMRATE program follows the POLYRATE hooks protocol, which is explained in the POLYRATE manual. The hooks protocol means that if an energy, a gradient, and/or a Hessian calculation is required by POLYRATE, the program calls the appropriate hooks subroutine, which will return the requested information. (The user may read more about hooks in the POLYRATE manual.)

6.1. DESCRIPTION OF NWCHEMRATE SUBPROGRAMS

This is a brief description of the NWCHEMRATE subroutines. 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 NWCHEM.

enatin Calculates the energy for atomic systems.

escener Carries out NWCHEM energy calculations using NWCHEM.

escfirst Carries out energy and first derivative calculations using NWCHEM.

escsec Carries out Hessian calculations using NWCHEM.

geoin Carries out geometry optimizations using NWCHEM.

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

nwinpfl Creates the NWCHEM input file for all calculations.

rdnwout Reads the output file generated by NWCHEM 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 NWCHEMRATE 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 either NWCHEM or POLYRATE methods.

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

rcodnw Reads the input option for the NWCHEM runs from file `esp.fu70`.

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

rinpnw Reads the `esp.fu70` NWCHEMRATE input file.

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

rstatnw Reads the input options given in the `esp.fu70` input file for the stationary points.

rdnwtop Reads the NWCHEM input file section that precedes the geometry.

rdnwmid Reads the NWCHEM input file section that after the geometry but before the NWCHEM task statement(s).

rdnwmethop Reads the NWCHEM options for the method (dft, scf, ccsd, etc.) section of the NWCHEM input file.

rdnwoptop Reads the NWCHEM options for the geometry optimization section of the NWCHEM input file.

rdnwscfop Reads the NWCHEM options for the scf section of the NWCHEM input file for post–Hartree-Fock methods.

rgennw Reads the general NWCHEMRATE options.

secin Carries out NWCHEM 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 NWCHEM calls.

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

The NWCHEMRATE restart option is controlled by the NWRRESTART keyword in the NWRGENERAL section of the unit `esp.fu70` input file. The NWCHEMRATE 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 NWCHEM 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, as the results of trying to restart such calculation are unpredictable). This restart option allows the user to change any parameter related to the reaction path calculations using as much information as possible from previous calculations.

Two restart files are used in this kind of restart calculation: `esp.fu83`, which stores the information to be read, as originated in a previous NWCHEMRATE calculation, and `esp.fu84`, where the useful information read from `esp.fu83` is stored along with the information calculated for the first time in the present NWCHEMRATE run (`esp.fu84` is the file to be used for future restart runs). The `esp.fu84` file has to be created by NWCHEMRATE, 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 NWCHEMRATE run is not large enough to describe some effect (for example, tunneling), one can extend the calculated reaction path without having to repeat the calculation of the points included in the previous NWCHEMRATE run. One should rename the `esp.fu84` file provided by the first run as `esp.fu83` and then run again with the NWRRESTART option 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 NWCHEM 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 obtain the correct frequencies. In general, we recommend always using a value of 1.0 amu for SCALEMASS in NWCHEMRATE.

A NWCHEMRATE restart run requires that all the information about all the stationary points be present in 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 Section 8.2 and Sections 7.C and 11.A of the POLYRATE manual.

The 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 checks if the required information is stored in this same file. If one of these conditions is not fulfilled, an NWCHEM calculation is run to obtain the desired information. If both conditions are fulfilled, then the information is read from `esp.fu83`, and no NWCHEM 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 the fact that the precision of the written information is limited, and redundant calculations could be carried out unnecessarily increasing 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, thereby resulting in erroneous results from NWCHEMRATE. The latter 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 safe. For very expensive calculations, one might want to use a higher value, but too high a value can give unphysical results. If the user increases this parameter, he or she has to be very aware of the risk it posses in the calculation, and all the results must be carefully checked. Therefore, only experienced users are advised to change this parameter.

One should not confuse the restart options of NWCHEMRATE with those of NWCHEM. NWCHEM also has the ability to restart jobs that were stopped in the middle for various reasons. In principle, one could include NWCHEM's restart options in a NWCHEMRATE job, but this has not been tested. Therefore, this practice is not recommended.

6.3. LIMITATIONS OF NWCHEMRATE

There are a number of limitations in the initial release of NWCHEMRATE, some of which associated with this program, while others are associated with NWCHEM. We hope that most of these limitations will eventually be eliminated. The limitations of NWCHEM are listed here are for version 5.0; for full details see the NWCHEM User's Manual. Earlier versions of NWCHEM may have additional limitations.

- Due to potentially complicated input required to be passed between NWCHEMRATE and NWCHEM, certain model chemistries in NWCHEM are not compatible with NWCHEMRATE, and any attempt to use them will result in an error message from NWCHEMRATE. These currently include MCSCF, TDDFT, and PSPW.
- The use of dummy atoms (`Bq` or `X`) in the input specification may confuse NWCHEMRATE when it tries to read the output and is thus not recommended.
- So far, NWCHEMRATE has only been tested using DFT and MP2.
- In NWCHEM, a number of methods do not have analytical second derivatives (Hessians), *inter alia*, open-shell DFT and meta-GGA functionals. Likewise, the TCE engine, used for correlated post-Hartree–Fock methods, does not have analytical gradients. We generally recommend that NWCHEMRATE only be run with methods that have analytical gradients, but this is a recommendation, not a requirement.

7. DESCRIPTION OF INPUT FILES

We will assume that the user knows how to use POLYRATE 9.7 and NWCHEM and is familiar with their keywords. Therefore, when talking about the common input files, we will only discuss the differences between running POLYRATE 9.7 and running NWCHEMRATE 2007.

7.1. FILE USAGE

In addition to the POLYRATE 9.7 and NWCHEM input files, NWCHEMRATE 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 NWCHEMRATE input data.

`esp.fu71` NWCHEM input file for geometry optimization of reactant 1, if such optimization is going to be carried out.

`esp.fu72` NWCHEM 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 NWCHEM program.

`esp.fu73` NWCHEM input file for geometry optimization of product 1, if such optimization is going to be carried out.

`esp.fu74` NWCHEM 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 NWCHEM program.

`esp.fu75` NWCHEM 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` NWCHEM 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 NWCHEM.

`esp.fu78` NWCHEM 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 NWCHEM.

`esp.fu81` NWCHEM temporary input file, generated by NWCHEMRATE.

`esp.fu82` NWCHEM temporary output file, generated by NWCHEM. If the NWCHEM 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` NWCHEMRATE uses the NWCHEM database file to extract the final data after an NWCHEM run. This file, `esp.db`, is, however, machine-readable only. Therefore, a subsequent NWCHEM job (`esp.fu87`) is run in order to convert this file to a human-readable format; the result of this file is `esp.fu85`.

`esp.fu86` NWCHEM Hessian output file, generated by NWCHEM.

`esp.fu87` An NWCHEM input file with only two NWCHEM statements: `restart` and `task rtdbprint`. This file must be present in the work directory prior to starting the NWCHEMRATE job, although the supplied version of taxi will generate this file if it is missing.

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.

In addition, there are two additional files to be discussed. The first, discussed in Section 3, is `taxi`. This shell script is used by NWCHEMRATE to call NWCHEM. The second is the job control script (the `*jobname*.jc` file). This file prepares an NWCHEMRATE job for execution by making a working directory with a unique name, copying all files to the appropriate names, and making sure all necessary files are present and all environmental variables are defined. It also, after the NWCHEMRATE run, copies all the relevant output files back to their final destination and clean up any work directories. The use of such a file, although not necessary, is highly recommended. A general example is provided in the `exe` directory, and each test job (see Section 8) also has a `.jc` file that can be used as a template.

7.2. DESCRIPTION OF fu5 UNIT

A NWCHEMRATE 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, just 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 9.7 manual. Here we clarify some points about preparing the input file for unit fu5:

RESTART

If a POLYRATE 9.7 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 NWCHEMRATE, controlled by the NWRRESTART keyword in NWRGENERAL 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 NWCHEMRATE restart. See Section 6.2 of this manual for further information.

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 in POLYRATE). Therefore, only size-consistent methods should be used in NWCHEMRATE. (Both DFT and MP2 are size consistent.)

EZERO

Because of the way the interface is constructed, the user must use the *calculate* option for the EZERO keyword of the ENERGETICS section, note that *calculate* is the default option.

POTENTIAL

The *hooks* option is required if the program will perform any NWCHEM calculations. The option can also be unit29, unit30, unit31, or unit40, but the geometries of reactants, products, and saddle point should be given in fu5 file and keywords INITGEO GEOM and STATUS 2 must be used in order to avoid any NWCHEM calculations.

OPTTS and OPTMIN

In NWCHEMRATE there are two ways to optimize the geometries: by means of NWCHEM routines or by means of POLYRATE 9.7 routines. The choice between the two methods is made 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 NWCHEMRATE to use the methods native to POLYRATE 9.7, 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, NWCHEMRATE will optimize the geometries of the stationary points using NWCHEM 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 NWCHEM 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, either using the POLYRATE 9.7 methods or the NWCHEM modules. The latter option is 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 NWCHEM can calculate a Hessian for any of the electronic structure methods available in NWCHEM, the option of performing the Hessian calculation using POLYRATE methods is implemented for completeness. However the user is advised for efficiency reasons using NWCHEM for the Hessian calculations because frequent system calls can lower the program efficiency when using POLYRATE for the numerical Hessian calculations. The user should be aware that analytical Hessian is available in NWCHEM only for certain methods; for more details, see Section 6.3. For other method, NWCHEM will need to calculate numerical Hessians.

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` also must be consistent with the one in ATOMS in the GENERAL section.

CONSTANT

If the user selects NWCHEM 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 NWCHEM manual. If a user selects POLYRATE 9.7 for optimizing the geometries, this keyword has the same function as in any other POLYRATE calculation, fixing the value of the selected coordinates and avoiding their optimization, as explained in Section 11.A.6 of the POLYRATE 9.7 manual.

INITGEO and STATUS

There are two options for reading the initial geometries of reactants, products, wells, and saddle point: The first is by means of units fu71 through fu78 if a NWCHEM optimization is desired (*OHOOK* option for the OPTMIN and OPTTS keywords in the OPTIMIZATION section). In this case, the INITGEO keyword should be set to *HOOKS*, and the files fu71-fu78 should be the appropriate input files for a NWCHEM optimization. The second is by means of the GEOM keyword if a POLYRATE 9.7 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. In this case, the INITGEO keyword needs 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 NWCHEM: 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 geometries should be provided in the files fu71 through fu78.

– If the geometry optimization of one of the stationary points is not desired, the INITGEO keyword should be set to *GEOM*, and the STATUS keyword should be set to the appropriate values 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 should 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 geometries are given in the appropriate sections 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 results in no optimization of the input geometry in the `fu5` file. If a value greater than *`2`* is selected, no Hessian calculation is performed. If STATUS is set to *`4`* in the START section of the `fu5` input file, the Hessian matrix as it appears in a NWCHEM 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 or DFT methods. With the anharmonicity options available in POLYRATE 9.7, 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 options that do not require derivatives higher than second are Morse I and hindered rotator.

SCALEMASS

SCALEMASS is the reduced mass to which all isoinertial coordinates are scaled. If restart calculations are desired, the value of the SCALEMASS variable keyword must be the same for the whole series of calculations. 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 NWCHEM-type data files. They are used only as input files for NWCHEM optimization on reactants 1 and 2 (`esp.fu71` and `esp.fu72`, respectively), products 1 and 2 (`esp.fu73` and `esp.fu74`), of the wells on the reactant (`esp.fu77`) and product (`esp.fu78`) sides, and of the saddle point or other starting geometry (`esp.fu75`). If no optimization using NWCHEM is desired, these files are not needed.

For constructing these files, the user should consult the NWCHEM manual. The user can also use examples of these files in the NWCHEMRATE test suite.

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 NWCHEMRATE options and some of the NWCHEM 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 ten sections, each starting with NWR in order to avoid confusion with similar sections in the `fu5` input file. These sections are NWRGENERAL, NWRCOMMON, NWRREACT1, NWRREACT2, NWRPROD1, NWRPROD2, NWRWELLR, NWRWELLP, NWRSTART and NWRPATH. In these sections, the appropriate keywords need to be specified.

Section name	Description
NWRGENERAL	General options
NWRCOMMON	NWCHEM options common to all the stationary points
NWRREACT1	NWCHEM options specific to the first reactant
NWRREACT2	NWCHEM options specific to the second reactant, if any
NWRPROD1	NWCHEM options specific to the first product
NWRPROD2	NWCHEM options specific to the second product, if any
NWRWELLR	NWCHEM options specific to the reactant well, if any
NWRWELLP	NWCHEM options specific to the product well, if any
NWRSTART	NWCHEM options specific to the starting (or saddle) point
NWRPATH	NWCHEM options common to all the nonstationary points along the reaction path
NWRNWREND	signals last line of the `esp.fu70` file (mandatory)

There are three types 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 NWREND line). See the POLYRATE manual for a more thorough discussion of the three types of keywords. (Note that a list terminates with an NWREND line, rather than an END line, to avoid confusion with NWCHEM blocks that also end with an END statement.)

7.4.1. The NWGENERAL Section

The NWRGENERAL section controls NWCHEMRATE's own options. There are only four keywords allowed in this section, NWRRESTART and NONWRRESTART (with NWRRESTART being the default), NWPROC, and RSTTOL.

NWRRESTART (NONWRRESTART)

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, NWCHEMRATE runs NWCHEM and writes the NWCHEM output information to unit `fu84`. If NONWRRESTART is selected, neither reading file `fu83` nor writing file `fu84` will be performed.

NWPROC

NWPROC is the number of processors on which to run NWCHEM.

Example:

```
NWPROC  16
```

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 NWCHEMRATE 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 advise not to change this value from its default value, *`1.E-8`*.

Example:

```
RSTTOL        0.00000001
```

7.4.2. The NWRCOMMON Section

The NWRCOMMON section reads the information common to all the NWCHEM calculations for the stationary points and points along the reaction path, excepting the NWCHEM 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 NWCHEM calculations unless a different keyword is specified in the section corresponding to a specific point. This section needs to precede the specific sections NWRREACT1, NWRREACT2, NWRPROD1, NWRPROD2, NWRSTART, NWRWELLR, NWRWELLP, and NWRPATH. The following keywords are allowed in the NWRCOMMON section:

NWRMETHOD

This keyword determines which type of electronic structure calculation is to be performed by NWCHEM. This should be the option used in the NWCHEM task statement. NWCHEMRATE was primarily written to handle DFT calculations, by HF (SCF) and post-HF methods are also possible. In the case of DFT, the value of NWRMETHOD should be simply 'DFT' rather than a specific exchange-correlation functional, which should be defined in the NWRMETHOP section below. At this point, most of the electronic structure methods available in NWCHEM are available for use in NWCHEMRATE. Nonetheless, certain methods are not available, because they require specialized options. These include TDDFT, MCSCF, and PSPW. These methods may be implemented in future versions.

Example:

```
NWRMETHOD DFT
```

NWRTOP

This is a list type keyword and therefore it must include an NWREND line to indicate the end of the list (see the POLYRATE manual for a description of list keywords). NWRTOP includes many of the keywords that NWCHEM will use for carrying out an energy calculation. Everything here will be written before NWCHEMRATE writes the geometry and method sections.

Example:

```
NWRTOP
  memory 24 mw

  echo

  basis
    * library 6-311++g3df_3pd \
/usr/local/nwchem/nwchem-5.0/data/libraries/
  end

NWREND
```

Note that in NWCHEM, the backslash ('\') at the end of a line concatenates the current line with the next. This can be very useful as the maximum line length in NWCHEMRATE is 80 characters.

NWRMID

This optional list type keyword contains any part of the NWCHEM input file that may come between the geometry and methods sections, which are automatically generated by NWCHEMRATE, and the final NWCHEM task statement. The use of the NWRMID keyword is probably rare but occasionally necessary. For instance, one may want to do a single-point energy calculation in a smaller basis set before any calculation in a larger basis set. This is often required, for example, in order to improve SCF convergence when using large basis sets. In such a situation, the smaller basis set would go in the NWRTOP section while the NWRMID section would include, for example, a *`task dft energy`* statement followed by the larger basis set definition.

Example:

```
NWRMID
  basis
    * library aug-cc-pVTZ
  end

NWREND
```

NWRMETHOP

This list type keyword contains all the options required in the NWCHEM method section. This may include special convergence options or method specification options. For instance, when using DFT, one would define here the specific exchange-correlation method desired. One normally would also specify here the multiplicity of the system (in those cases where the system is not a singlet), but when using NWCHEMRATE, this option **must not** be set. Rather, the multiplicity statements are inserted automatically by NWCHEMRATE and unpredictable results may be obtained if the user were to also specify the multiplicity in the NWRMETHOP section. Also note that when using post-HF methods, this section refers to the post-HF method options and any options required for the SCF calculation should be specified in the NWRSCFOP section. Note that only the options should be provided and not the method as this is set with NWRMETHOD keyword.

Example:

```
NWRMETHOP
   XC M05
   ITERATIONS 200
NWREND
```

NWRSCFOP

NWRSCFOP is a list keyword for specifying any options related to the SCF part of a post-HF calculation. In all other circumstances, this section should be absent. If one were only doing an HF (SCF) calculation instead of post-HF calculation, then the SCF options should be given in the NWRMETHOP section and not here.

Example:

```
NWRSCFOP
      SCF
      Maxiter 200
      END
NWREND
```

NWROPTOP

This section lists any options needed for using NWCHEM to optimize geometries. The first one should be the selection of either the DRIVER or STEPPER NWCHEM module. In the current version, NWCHEMRATE does not generate NWCHEM optimization files but rather requires them to be already generated by the user prior to starting NWCHEMRATE. This feature will be implemented in the next version of NWCHEMRATE.

Example:

```
NWROPTOP
 DRIVER
   MAXITER 100
   XYZ
   INHESS 2
 END
NWREND
```

7.4.3. The NWRREACT1, NWRREACT2, NWRPROD1, NWRPROD2, NWRWELLR, NWRWELLP, NWRSTART, and NWRPATH Sections

Since in some cases, the options selected in the NWRCOMMON 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 NWRENER, NWRFIRST, and NWRSEC list keywords. The options for the first reactant should be specified in the NWRREACT1 section; in the NWRREACT2 the options for the second reactant, if it exists; in NWRPROD1 and NWRPROD2 the options for the first product and, if it exists, the second one; in NWRWELLR the options for a well on the reactant side of the reaction path, if it exists; in NWRWELLP the options for a well on the product side, if it exists; section NWRSTART will indicate the options for the saddle point or starting geometry, and NWRPATH will indicate the options for NWCHEM calculations on the non-stationary points along the reaction path. If no changes from the options selected in the NWRCOMMON section are wanted, these sections can be skipped.

Currently, only three keywords are allowed in these sections: CHARGE, MULTIPLICITY, and SCFTYPE.

CHARGE, MULTIPLICITY, and SCFTYPE

If NWCHEM optimization is required, NWCHEMRATE will read the charge, the multiplicity of each species, and the type of SCF (if HF or post-HF method is used) from the NWCHEM optimization output. Nevertheless, if no optimization is selected or if POLYRATE 9.7 is selected for carrying out the optimizations, NWCHEMRATE will need to know the charge and multiplicity of the system and type of SCF before making any NWCHEM calculation. The way to input these characteristics is by means of the CHARGE, MULTIPLICITY, and SCFTYPE keywords. These keywords are not allowed in the NWRCOMMON section since they are dependent on the species and are only required for the systems that will not be optimized using NWCHEM, being ignored if they are specified for a species that is going to be optimized using NWCHEM. 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 NWRPATH section they will also be ignored.

These are variable keywords that must be followed by an integer, which is the value of the charge and multiplicity of the system. The default values are `0` and `1`, respectively, which correspond to a neutrally charged singlet. Multiplicities of `2` and `3` correspond to doublets and triplets.

Example:

```
CHARGE          –1
MULTIPLICITY     2
SCFTYPE         UHF
```

7.5. DESCRIPTION OF INTERPOLATED VTST BY MAPPING (IVTST-M) USAGE

IVTST-M is a method for obtaining the information required for a VTST calculation all along the reaction-path from the information available at an arbitrary, but small, number of points. The only limitations on using the IVTST-M algorithm are that a saddle point exists and reaction path information is available at least one nonstationary point on each side of the saddle point. In addition, as in all IVTST methods, information is required for 3 – 5 stationary points (reactants, saddle point, products, and optionally one or two wells). The level of IVTST-M is denoted by two numbers, H and G, indicating respectively the number of nonstationary points at which Hessians are available and the number of nonstationary points at which gradients are available. Thus, IVTST-M-6/54 indicates that the interpolation is based in 6 nonstationary Hessian points, at which geometries, energies, gradients, and Hessians are available, and 48 points (which is 54 minus 6) for which only geometries, energies, and gradients are available.

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

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

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

The first way to perform IVTST-M calculations in NWCHEMRATE is to use electronic structure input file fu30, fu40, or fu31 (fu31 is recommended). This method is described fully in the POLYRATE manual. The input file fu31 can be generated by a preliminary run with the keyword `writefu31` in the `General` section. In this preliminary run, one only needs to use a very small range of *s* to get some points along the reaction path that will be used for interpolations in IVTST-M

The second way to perform IVTST-M calculations is in conjunction with the hooks; this requires some additional keywords, namely `EXFIRST` and `EXSECOND`. A calculation with the hooks involves two stages. At the end of the first stage, one has stored *s*, $V_{\mathrm{MEP}}(s)$, and the gradients on the fine grid, and *s*, $V_{\mathrm{MEP}}(s)$, $\omega(s)$, $B_{mF}\ (s)$, and $\left|\mathbf{I}(s)\right|$ are stored on the save grid. (The grids are defined in Sect. 7.A of the POLYRATE manual) Both grids extend from s = SLM on the reactant side to s = SLP on the product side, and there are G + 1 points (including the saddle point but excluding reactants and products) on the fine grid and H + 1 points (including the saddle point but excluding reactants and products) on the save grid. Then mapped interpolation is carried out to add EXNSTEP * EXSTEP points on each side of the grids. Furthermore SLM is replaced by SLM – EXNSTEP * EXSTEP, and SLP is replaced by SLP + EXNSTEP * EXSTEP.

8. TESTING NWCHEMRATE

8.1. TESTS PERFORMED

We tested version 2007 of NWCHEMRATE using version 9.7 of POLYRATE and the indicated versions of NWCHEM on the following machines (the operating system is indicated in parentheses):

1. IBM Regatta with Power4 processors (AIX 5.3): NWCHEM version 5.0
2. SGI Altix with Itanium 2 processors (Linux): NWCHEM version 5.0
4. IBM BladeCenter with AMD Opteron processor (SuSE Linux): NWCHEM version 5.0

The code is expected to work with all NWCHEM versions that use input and output files with the same structure as the above mentioned versions.

8.2. TEST SUITE

The purpose of the test suite is to provide some examples of the way NWCHEMRATE works, and to check the most important NWCHEM and POLYRATE 9.7 options for direct dynamics. Therefore, the number of tests included is only fifteen, and the tests illustrate only the most basic features of NWCHEMRATE. The level of calculation, size of the systems, and computational requirements of the test runs have been chosen in such a way that running the whole suite does not consume an inordinate amount of computing time. All the test runs were run after compiling NWCHEMRATE using the `param4.inc` file.

The test suite is located in the `testrun` directory, which contains four 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 output files were obtained from runs on an IBM Power4 Regatta computer.) The Korn shell script `checknw` in the `testrun` directory allows for an easy comparison between the `poly.fu15` output files obtained after running all the test runs using NWCHEM 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:

Reaction	**Directory**	**Test Names**
$OH + H_2 \rightarrow H_2O + H$	`oh3`	`oh3tr1`, `oh3tr2`, `oh3tr3`, `oh3tr4`
$CH_3 + H_2 \rightarrow CH_4 + H$	`ch5`	`ch5tr1`, `ch5tr2`, `ch5tr3`, `ch5tr4`
$CH_4 + OH \rightarrow CH_3 + H_2O$	`ch4oh`	`ch4ohtr1`, `ch4ohtr2`, `ch4ohtr3`, `ch4ohtr4`, `ch4ohtr5`, `ch4ohtr6`.
$NH_3 + OH \rightarrow NH_2 + H_2O$	`nh3oh`	`nh3ohtr1`

The test runs are run in a subdirectory created for each run. In this way, problems that could be caused when running other NWCHEM or NWCHEMRATE calculations in the same directory that NWCHEMRATE is using are avoided. After the test run is completed, the directory is removed, and the temporary files created are moved to the `NWRTEMPFILES` directory.

8.2.1. Test Run `oh3tr1`

Tunneling:	none
Number of levels:	single level (MPW1K/3-21G*)
Hindered rotation:	none
IVTST/M:	off

This is a test run for a conventional TST calculation for the reaction OH + H_2 → H_2O + 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 using NWCHEM, 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 NWCHEMRATE restart file `esp.fu84` is written.

NWCHEMRATE I/O files	NWCHEMRATE file names	
`oh3tr1.70`	`esp.fu70`	Input data for NWCHEMRATE
`oh3tr1.75`	`esp.fu75`	Input data for the saddle point optimization
`oh3tr1.dat`	`poly.fu5`	Input data for POLYRATE
`oh3tr1.61`	`esp.fu61`	POLYRATE output file containing information about the stationary points
`oh3tr1.fu6`	`poly.fu6`	Long output file
`oh3tr1.fu15`	`poly.fu15`	Summary output file
`oh3tr1.rst`	`esp.fu84`	Restart file written by NWCHEMRATE

8.2.2. Test Run `oh3tr2`

Tunneling:	none
Number of levels:	single level (MPW1K/3-21G*)
Hindered rotation:	none
IVTST/M:	off

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 0.012 Angstrom. The reaction path is calculated between –0.15 Angstrom and +0.15 Angstrom. 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 NWCHEMRATE although not by NWCHEM. 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).

NWCHEMRATE I/O files	**NWCHEMRATE file names**	
`oh3tr2.70`	`esp.fu70`	Input data for NWCHEMRATE
`oh3tr2.75`	`esp.fu75`	Input data for the saddle point
`oh3tr2.dat`	`poly.fu5`	Input data for POLYRATE
`oh3tr2.61`	`esp.fu61`	POLYRATE output file containing information about the stationary points
`oh3tr2.fu6`	`poly.fu6`	Long output file
`oh3tr2.fu15`	`poly.fu15`	Summary output file
`oh3tr2.rst.old`	`esp.fu83`	Restart file read by NWCHEMRATE
`oh3tr2.rst`	`esp.fu84`	Restart file written by NWCHEMRATE

8.2.3. Test Run oh3tr3

Tunneling:	none
Number of levels:	single level (MPW1K/3-21G*)
Hindered rotation:	none
IVTST/M:	off

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 is stored in the oh3tr2.rst file than is required for the calculation, 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 NWCHEM calculations are needed for this test run, so it is the fastest test run of the suite.

NWCHEMRATE I/O files	**NWCHEMRATE file names**	
oh3tr3.70	esp.fu70	Input data for NWCHEMRATE
oh3tr3.75	esp.fu75	Input data for the saddle point
oh3tr3.dat	poly.fu5	Input data for POLYRATE
oh3tr3.61	esp.fu61	POLYRATE output file containing information about the stationary points
oh3tr3.fu6	poly.fu6	Long output file
oh3tr3.fu15	poly.fu15	Summary output file
oh3tr3.rst.old	esp.fu83	Restart file read by NWCHEMRATE
oh3tr3.rst	esp.fu84	Restart file written by NWCHEMRATE

8.2.4. Test Run `oh3tr4`

Tunneling:	SCT
Number of levels:	single level (MP2/6-31G*)
Hindered rotation:	none
IVTST/M:	off

This is a test run that uses the MP2 method for direct dynamics with NWCHEMRATE. The reaction path is calculated between –0.60 Angstrom and +0.60 Angstrom. Nonredundant internal coordinates are used for describing the vibrations. The geometries of stationary points are provided in the fu5 input file. Because the current version of NWCHEM cannot perform a calculation for the one-electron hydrogen atom when specifying any post-HF method in the NWCHEM input file, the PROD 1, hydrogen atom, is calculated at the HF level although MP2 method is specified in .70 file.

NWCHEMRATE I/O files	**NWCHEMRATE file names**	
`oh3tr4.70`	`esp.fu70`	Input data for NWCHEMRATE
`oh3tr4.dat`	`poly.fu5`	Input data for POLYRATE
`oh3tr4.61`	`esp.fu61`	POLYRATE output file containing information about the stationary points
`oh3tr4.fu6`	`poly.fu6`	Long output file
`oh3tr4.fu15`	`poly.fu15`	Summary output file
`oh3tr4.rst`	`esp.fu84`	Restart file written by NWCHEMRATE

8.2.5. Test Run `ch5tr1`

Tunneling:	SCT
Number of levels:	single level (M05/6-31G)
Hindered rotation:	none
IVTST/M:	off

This sample run for the reaction $CH_3 + H_2 \rightarrow CH_4 + H$ is based on the test run `ch5fu30tr5` included in the POLYRATE 9.7 test suite. It performs calculations of the TST, CVT and ICVT rate constants, and it is based on following the reaction path using the Euler steepest descents method (also called Euler single-step method). For all of the stationary points, symmetry is employed in all the NWCHEM calculations (which is the default for NWCHEM 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 (NWRRESTART keyword), and all the input of the geometries is done by means of files `fu71` through `fu74`, in order to allow NWCHEM 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, and PROD2 in the input file `fu5`. INITGEO is set to GEOM and STATUS is set to 2 in START sections in the input file `fu5`).

NWCHEMRATE I/O files	NWCHEMRATE file names	
`ch5tr1.70`	`esp.fu70`	Input data for NWCHEMRATE
`ch5tr1.71`	`esp.fu71`	Input data for reactant 1 optimization
`ch5tr1.72`	`esp.fu72`	Input data for reactant 2 optimization
`ch5tr1.73`	`esp.fu73`	Input data for product 1 optimization
`ch5tr1.74`	`esp.fu74`	Input data for product 2 optimization
`ch5tr1.dat`	`poly.fu5`	Input data for POLYRATE
`ch5tr1.61`	`esp.fu61`	POLYRATE output file containing information about the stationary points
`ch5tr1.fu6`	`poly.fu6`	Long output file
`ch5tr1.fu15`	`poly.fu15`	Summary output file
`ch5tr1.rst`	`esp.fu84`	Restart file written by NWCHEMRATE

8.2.6. Test Run ch5tr2

Tunneling:	SCT
Number of levels:	dual level
Hindered rotation:	none
IVTST/M:	off

This test run uses the restart file created by the previous run and performs a calculation using the ISPE technique by using eight MP2/6-31G** energies input in the esp.fu51 input file. Of these eight energies, two are for products and the saddle point, and six 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.

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

8.2.7. Test Run `ch5tr3`

Tunneling:	SCT
Number of levels:	single level (M05/6-31G)
Hindered rotation:	none
IVTST/M:	on

The IVTST-M algorithm is used for this sample run with hooks. The step size of *s* is set to 0.01 angstrom, and *s* range is set to 0.20 to −0.20 angstrom. The keywords `EXFIRST` and `EXSECOND` are used for IVTST-M algorithm. The reaction coordinate s is extended to -2.0 − +2.0 angstrom with a 0.01 angstrom step size. The keyword `FIXMUEF` is used to apply a global interpolation to estimate the effective reduced mass of the SCT algorithm at the saddle point. The geometries for the reactants, products, and saddle point are given in the `fu5` input file (`INITGEO` is set to `GEOM`, and `STATUS` set to *2* in the `REACT1`, `REACT2`, `PROD1, PROD2`, and `START` sections of the `fu5` input file). `Fu31` file is written by using keyword `writefu31` in the `fu5` input file, which will be used for the next test run.

NWCHEMRATE I/O files	NWCHEMRATE file names	
`ch5tr3.dat`	`poly.fu5`	Input data for POLYRATE
`ch5tr3.61`	`esp.fu61`	POLYRATE output file containing information about the stationary points
`ch5tr3.fu6`	`poly.fu6`	Long output file
`ch5tr3.fu15`	`poly.fu15`	Summary output file
`ch5tr3.fu31`	`poly.fu31`	Input data for IVTST-M algorithm

8.2.8. Test Run `ch5tr4`

Tunneling:	SCT
Number of levels:	single level (M05/6-31G)
Hindered rotation:	none
IVTST/M:	on (with fu31)

The IVTST-M algorithm is used for this sample run with the `fu31` input file that is generated in the ch5tr3 test run. The information used for interpolations is read from fu31 input file by specifying the keyword `POTENTIAL UNIT31` in `ENERGETICS` section. The reaction path is calculated at the s range of -2.00 to 2.00 angstrom by using the interpolation of information provided in `fu31` file. There is no electronic structure calculation by nwchem in this test run.

NWCHEMRATE I/O files	NWCHEMRATE file names	
`ch5tr4.dat`	`poly.fu5`	Input data for POLYRATE
`ch5tr4.61`	`esp.fu61`	POLYRATE output file containing information about the stationary points
`ch5tr4.fu6`	`poly.fu6`	Long output file
`ch5tr4.fu15`	`poly.fu15`	Summary output file
`ch5tr3.fu31`	`poly.fu31`	Input data for IVTST-M algorithm

8.2.9. Test Run `ch4ohtr1`

Tunneling:	none
Number of levels:	single level (M05-2X/3-21G)
Hindered rotation:	none
IVTST/M:	off

This is the first of a series of tests that try to show how NWCHEMRATE would usually be used. We study the CH_4 + OH → CH_3 + H_2O reaction. The sequence actually has five stages. The first four stages may be thought of a complete single-level calculation or as runs 1-4 of a five-run dual-level calculation. The first step we will take is the calculation of the first reactant, CH_4. In the `fu5` input file we only include the REACT1 section, and only the additional input files `fu70` and `fu71` are required. The NWCHEMRATE 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.

NWCHEMRATE I/O files	**NWCHEMRATE file names**	
`ch4ohtr1.70`	`esp.fu70`	Input data for NWCHEMRATE
`ch4ohtr1.71`	`esp.fu71`	Input data for reactant 1 optimization
`ch4ohtr1.dat`	`poly.fu5`	Input data for POLYRATE
`ch4ohtr1.61`	`esp.fu61`	POLYRATE output file containing information about reactant 1
`ch4ohtr1.fu6`	`poly.fu6`	Long output file

8.2.10. Test Run `ch4ohtr2`

Tunneling:	none
Number of levels:	single level (M05-2X/3-21G)
Hindered rotation:	none
IVTST/M:	off

This is the second step in the four-stage single-level calculation for the CH_4 + OH → CH_3 + H_2O 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_2O, 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 NWCHEMRATE 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.

NWCHEMRATE I/O files	NWCHEMRATE file names	
`ch4ohtr2.70`	`esp.fu70`	Input data for NWCHEMRATE
`ch4ohtr2.72`	`esp.fu72`	Input data for reactant 2 optimization
`ch4ohtr2.73`	`esp.fu73`	Input data for product 1 optimization
`ch4ohtr2.74`	`esp.fu74`	Input data for product 2 optimization
`ch4ohtr2.dat`	`poly.fu5`	Input data for POLYRATE
`ch4ohtr2.61`	`esp.fu61`	POLYRATE output file containing information about the stationary points
`ch4ohtr2.fu6`	`poly.fu6`	Long output file

8.2.11. Test Run `ch4ohtr3`

Tunneling:	none
Number of levels:	single level (M05-2X/3-21G)
Hindered rotation:	none
IVTST/M:	off

This is the third step in the four-stage single-level calculation for the CH_4 + OH → CH_3 + H_2O 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.

NWCHEMRATE I/O files	NWCHEMRATE file names	
`ch4ohtr3.70`	`esp.fu70`	Input data for NWCHEMRATE
`ch4ohtr3.75`	`esp.fu75`	Input data for the saddle point optimization
`ch4ohtr3.dat`	`poly.fu5`	Input data for POLYRATE
`ch4ohtr3.61`	`esp.fu61`	POLYRATE output file containing information about the stationary points
`ch4ohtr3.fu6`	`poly.fu6`	Long output file

8.2.12. Test Run `ch4ohtr4`

Tunneling:	SCT
Number of levels:	single level (M05-2X/3-21G)
Hindered rotation:	mode 14
IVTST/M:	off

This is the fourth and final step in the four-stage calculation for the CH_4 + OH → CH_3 + H_2O 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 NWCHEMRATE restart option was selected in the `fu70` input file, writing the information for future calculations in the `fu84` file. It should be noted that, since no NWCHEM calculations are 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).

NWCHEMRATE I/O files	**NWCHEMRATE file names**	
`ch4ohtr4.70`	`esp.fu70`	Input data for NWCHEMRATE
`ch4ohtr4.dat`	`poly.fu5`	Input data for POLYRATE
`ch4ohtr4.61`	`esp.fu61`	POLYRATE output file containing information about the stationary points
`ch4ohtr4.fu6`	`poly.fu6`	Long output file
`ch4ohtr4.fu15`	`poly.fu15`	Summary output file
`ch4ohtr4.rst`	`esp.fu84`	Restart file written by NWCHEMRATE

8.2.13. Test Run `ch4ohtr5`

Tunneling:	LCT
Number of levels:	dual level
Hindered rotation:	mode 14
IVTST/M:	off

This test run completes a IOC dual-level calculation of the $CH_4 + OH \rightarrow CH_3 + H_2O$ reaction. The lower-level information is read from the restart file from the previous run. The higher-level data is included by means of the `fu50` file. Tunneling is calculated by the microcanonical optimized multidimensional tunneling method (muOMT), requiring therefore some calculation in the reaction-swath region that are not present in the restart file, since the tunneling for the previous run was calculated using the SCT approach. 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.

NWCHEMRATE I/O files	**NWCHEMRATE file names**	
`ch4ohtr5.70`	`esp.fu70`	Input data for NWCHEMRATE
`ch4ohtr4.rst`	`esp.fu83`	Restart file read by NWCHEMRATE
`ch4ohtr5.dat`	`poly.fu5`	Input data for POLYRATE
`ch4ohtr5.50`	`poly.fu50`	Additional input data for the POLYRATE VTST-IC calculation
`ch4ohtr5.61`	`esp.fu61`	POLYRATE output file containing information about the stationary points

`ch4ohtr5.fu6`	`poly.fu6`	Long output file
`ch4ohtr5.fu15`	`poly.fu15`	Summary output file
`ch4ohtr5.rst`	`esp.fu84`	Restart file written by NWCHEMRATE

8.2.14. Test Run ch4ohtr6

Tunneling:	LCT
Number of levels:	dual level
Hindered rotation:	mode 14
IVTST/M:	off

This test run completes an IOC dual-level calculation of the CH_4 + OH → CH_3 + H_2O 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.

NWCHEMRATE I/O files	NWCHEMRATE file names	
ch4ohtr6.70	esp.fu70	Input data for NWCHEMRATE
ch4ohtr4.rst	esp.fu83	Restart file read by NWCHEMRATE
ch4ohtr6.dat	poly.fu5	Input data for POLYRATE
ch4ohtr6.50	poly.fu50	Additional input data for the POLYRATE VTST-IC calculation
ch4ohtr6.61	esp.fu61	POLYRATE output file containing information about the stationary points
ch4ohtr6.fu6	poly.fu6	Long output file
ch4ohtr6.fu15	poly.fu15	Summary output file
ch4ohtr6.rst	esp.fu84	Restart file written by NWCHEMRATE

8.2.15. Test Run `nh3ohtr1`

Tunneling:	none
Number of levels:	single level (B3LYP/3-21G*)
Hindered rotation:	none
IVTST/M:	off

This is a test run for the NH_3 + OH → NH_2 + H_2O reaction including wells on the reactant and product sides. The geometries are optimized by NWCHEM. 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 NWCHEM. With the information for the stationary points obtained in this way, a conventional TST rate constant without tunneling is calculated.

NWCHEMRATE I/O files	**NWCHEMRATE file names**	
`nh3ohtr1.70`	`esp.fu70`	Input data for NWCHEMRATE
`nh3ohtr1.71`	`esp.fu71`	Input data for reactant 1 optimization
`nh3ohtr1.72`	`esp.fu72`	Input data for reactant 2 optimization
`nh3ohtr1.73`	`esp.fu73`	Input data for product 1 optimization
`nh3ohtr1.74`	`esp.fu74`	Input data for product 2 optimization
`nh3ohtr1.75`	`esp.fu75`	Input data for the saddle point optimization
`nh3ohtr1.77`	`esp.fu77`	Input data for the reactant well optimization
`nh3ohtr1.78`	`esp.fu78`	Input data for the product well optimization
`nh3ohtr1.dat`	`poly.fu5`	Input data for POLYRATE
`nh3ohtr1.61`	`esp.fu61`	POLYRATE output file containing information about the stationary points
`nh3ohtr1.fu6`	`poly.fu6`	Long output file
`nh3ohtr1.fu15`	`poly.fu15`	Summary output file

8.3. TEST RUN TIMINGS

Here are the timings for the test runs, in CPU seconds. These timings are sum of user and system time on a single processor.

The following timings apply to version 2007/P9.7 with NWCHEM version 5.0. These calculations are all run on a single processor.

	IBM Regatta Computer With Power4 Processors	**IBM BladeCenter with AMD Opteron Processors**
Test run ch5tr1	13036.5	6985.9
Test run ch5tr2	3.4	1.3
Test run ch5tr3	3143.6	1758.1
Test run ch5tr4	0.1	0.0
Test run oh3tr1	548.3	253.4
Test run oh3tr2	1875.8	823.3
Test run oh3tr3	0.0	0.0
Test run oh3tr4	995.4	523.4
Test run ch4ohtr1	287.8	167.6
Test run ch4ohtr2	413.8	241.4
Test run ch4ohtr3	2080.3	1433.4
Test run ch4ohtr4	9997.0	5750.0
Test run ch4ohtr5	10174.8	6804.0
Test run ch4ohtr6	11640.5	7100.3
Test run nh3ohtr1	3251.3	1795.4

8.4. COMPARISON BETWEEN NWCHEMRATE AND GAUSSRATE

We performed the same calculations using NWCHEMRATE and GAUSSRATE. The results of forward rate constants are shown in the following table. The reaction is

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

The forward rate constants were calculated at the PBE0/6-31G level. PBE0 are chosen because PBE0 (also called PBE1PBE or PBEh in some places in the literatures) is a popular functional defined uniformly in most electronic structure packages. Here we used 0.01 Å for step size of *s* and *s* range from -0.30 – 0.30 Å. These values should be enough for purpose of testing program.

Summary of forward rate constants (cm^3/molecule^{-1} s^{-1})

T(K)	TST	CVT	CVT/ZCT	CVT/SCT
GUASSRATE				
298.0	4.90E-17	4.79E-17	4.82E-15	5.90E-14
400.0	7.46E-16	7.33E-16	1.01E-14	5.86E-14
667.0	1.88E-14	1.86E-14	4.83E-14	1.03E-13
1000.0	1.15E-13	1.15E-13	1.76E-13	2.50E-13
1550.0	5.98E-13	5.95E-13	7.11E-13	8.25E-13
NWCHEMRATE				
298.0	4.88E-17	4.78E-17	4.81E-15	5.87E-14
400.0	7.43E-16	7.31E-16	1.01E-14	5.84E-14
667.0	1.87E-14	1.85E-14	4.83E-14	1.03E-13
1000.0	1.15E-13	1.14E-13	1.75E-13	2.49E-13
1550.0	5.97E-13	5.93E-13	7.09E-13	8.23E-13

The numbers in the tables show that two programs give the identical results using the same electronic structure method and basis set.

9. COMPUTERS, OPERATING SYSTEMS, AND NWCHEM VERSIONS ON WHICH THE CODE HAS BEEN DEVELOPED AND TESTED

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

NWCHEMRATE VERSION		2007/P9.7-NW5.0
Machine (OS)	**Compilers**	**NWCHEM Version**
IBM Regatta, Power4 procs (AIX 5.3)	XLF (v10.1)	5.0
IBM BladeCenter AMD Opteron procs (SuSE Linux)	g77 from gcc v3.3.3	5.0
SGI-Altix, Itanium 2 procs (SuSE Linux)	g77 from gcc v3.3.3	5.0

10. VERSION INFORMATION

11.1. NWCHEMRATE VERSION NUMBERS

NWCHEMRATE version numbers have the general form 200x/Px'.y'.z'-NWx".y" where 200x is the NWCHEMRATE version number, x'.y'.z' is version of POLYRATE to which it is linked, and x".y" is the number of the version of NWCHEM being used. In general, the lowest level revisions (y or z') are bug fixes, while the higher level revisions (x, x', or y') are significant feature enhancements. The part of the version number before the solidus(/) is unique, that is, it changes if the version of POLYRATE changes or the version of NWCHEM changes or the interface changes. Thus the part after the solidus is redundant and may be omitted for brevity.

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

10.2. NWCHEMRATE REVISION HISTORY

This section contains the revision history as of version 2007 (December 2007), the first version of the code.

Version 2007

1. This is the initial release version of NWCHEMRATE.

11. ACKNOWLEDGMENTS

The development of NWCHEMRATE has been supported in part by grants from the U.S. Department of Energy, Office of Basic Energy Sciences (which supports research on VTST/MT) and the Office of Naval Research (which supports the development of integrated software tools).