

NWChemrate – version 2019
MANUAL

Lu Gem Gao,^{a,b} Jingjing Zheng,^a Mark A. Iron,^a Benjamin A. Ellingson,^a
José C. Corchado,^a Yao-Yuan Chuang,^a Xuefei Xu,^b and Donald G. Truhlar^a

^a *Department of Chemistry, Chemical theory Center, and Supercomputing Institute,
University of Minnesota, Minneapolis, Minnesota 55455-0431*

^b *Center for Combustion Energy, Department of Energy and Power Engineering, and Key
Laboratory for Thermal Science and Power Engineering of Ministry of Education,
Tsinghua University, Beijing 100084, China*

Program version: 2019
Program version in long form: 2019/P17-C-NW6.8
Program version date: August 5, 2019
Manual version date: July 17, 2021

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 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 functional, MP2, or coupler cluster methods in the *NWChem* package. 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 2019 provides an interface between two programs:

Zheng, J.; Bao, J. L.; Meana-Pañeda, R.; Zhang, S.; Lynch, B. J.; Corchado, J. C.; Chuang, Y.-Y.; Fast, P. L.; Hu, W.-P.; Liu, Y.-P.; Lynch, G. C.; Nguyen, K. A.; Jackels, C. F.; Fernandez Ramos, A.; Ellingson, B. A.; Melissas, V. S.; Villà, J. Rossi, I.; Coitiño, E. L.; Pu, J.; Albu, T. V.; Ratkiewicz, A.; Steckler, R.; Garrett, B. C.; Isaacson, A. D.; Truhlar, D. G. *Polylrate*-version 2017-C; University of Minnesota: Minneapolis, 2017.

and

M. Valiev, E.J. Bylaska, N. Govind, K. Kowalski, T.P. Straatsma, H.J.J. van Dam, D. Wang, J. Nieplocha, E. Apra, T.L. Windus, W.A. de Jong, *NWChem: A Comprehensive and Scalable Open-Source Solution for Large Scale Molecular Simulations*, *Comput. Phys. Commun.* **181**, 1477-1489 (2010).

Table of Contents

1. Licensing.....	5
2. Required References.....	6
3. Introduction.....	7
4. Distribution.....	9
5. Installation.....	12
6. Program Description.....	13
6.1. Description of <i>NWChemrate</i> Subprograms.....	14
6.2. Restarting a Calculation.....	17
6.3. Limitations of <i>NWChemrate</i>	20
7. Description of Input Files.....	21
7.1. File Usage.....	22
7.2. Description of fu5 Unit.....	25
RESTART.....	25
ATOMS.....	25
NOSUPERMOL.....	25
EZERO.....	26
POTENTIAL.....	26
OPTTS and OPTMIN.....	26
SECOND.....	27
GEOM.....	28
CONSTANT.....	28
INITGEO and STATUS.....	28
ANHARMONICITY.....	30
SCALEMASS.....	30
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 NWGENERAL Section.....	34
NWRRESTART (NONWRRESTART).....	34
NWPROC.....	34
RSTTOL.....	34
7.4.2. The NWRCOMMON Section.....	35
NWRMETHOD.....	35
NWRTOP.....	35
NWRMID.....	36
NWRMETHOP.....	36
NWRSCFOP.....	37
NWROPTOP.....	37
7.4.3. The NWRREACT1, NWRREACT2, NWRPROD1, NWRPROD2, NWRWELLR, NWRWELLP, NWRSTART, and NWRPATH Sections.....	39
CHARGE, MULTIPLICITY, and SCFTYPE.....	39
7.5. Description of Interpolated VTST by Mapping (IVTST-M) Usage.....	41
8. Testing <i>NWChemrate</i>	43
8.1. Tests Performed.....	43
8.2. Test Suite.....	44
8.2.1. Test Run oh3tr1.....	46

8.2.2. Test Run oh3tr2.....	47
8.2.3. Test Run oh3tr3.....	49
8.2.4. Test Run oh3tr4.....	50
8.2.5. Test Run oh3tr5.....	51
8.2.6. Test Run ch5tr1.....	52
8.2.7. Test Run ch5tr2.....	54
8.2.8. Test Run ch5tr3.....	56
8.2.9. Test Run ch4ohtr1	57
8.2.10. Test Run ch4ohtr2	58
8.2.11. Test Run ch4ohtr3	60
8.2.12. Test Run ch4ohtr4	61
8.2.13. Test Run ch4ohtr5	63
8.2.14. Test Run ch4ohtr6	65
8.2.15. Test Run nh3ohtr1	67
8.2.16. Test Run R6.....	69
8.3. Test Run Timings	70
8.4. Comparison between <i>NWChemrate</i> and <i>Gaussrate</i>	71
9. Computers, Operating Systems, and <i>NWChem</i> Versions on which the Code has been Developed and Tested.....	73
10. Version Information.....	74
11.1. <i>NWChemrate</i> Version Numbers.....	74
10.2. <i>NWChemrate</i> Revision History.....	75
11. Acknowledgments.....	76

1. LICENSING

NWChemrate – version 2019 is licensed under the [Apache License, Version 2.0](#).
The manual of *NWChemrate* – version 2019 is licensed under [CC-BY-4.0](#).

Publications of results obtained with the *NWChemrate* – version 2019 software should cite the program and/or the article describing the program.

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

The use of the *NWChemrate* – version 2019 implies acceptance of the terms of the 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. L. G. Gao, J. Zheng, M. A. Iron, B. A. Ellingson, J. C. Corchado, Y.-Y. Chuang, E. L. Coitiño, X. Xu, and D. G. Truhlar, *NWChemrate* –version 2019, University of Minnesota, Minneapolis, MN, 2019.
2. J. Zheng, J. L. Bao, R. Meana-Pañeda, S. Zhang, B. J. Lynch, J. C. Corchado, Y.-Y. Chuang, P. L. Fast, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, A. Fernandez Ramos, B. A. Ellingson, V. S. Melissas, J. Villà, I. Rossi, E. L. Coitiño, J. Pu, T. V. Albu, A. Ratkiewicz, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar, *Polyrate* -version 2017-C, University of Minnesota, Minneapolis, MN, 2017.
3. E. Aprà, E. J. Bylaska, W. A. de Jong, N. Govind, K. Kowalski, T. P. Straatsma, M. Valiev, H. J. J. van Dam, Y. Alexeev, J. Anchell, V. Anisimov, F. Aquino, R. Atta-Fynn, J. Autschbach, N. Baumann, D. Bernholdt, K. Bhaskaran-Nair, S. Bogatko, P. Borowski, J. Brabec, E. Cauwet, Y. Chen, G. N. Chuev, C. J. Cramer, J. Daily, M. Deegan, T. H. Dunning Jr., M. Dupuis, K. G. Dyall, G. I. Fann, S. A. Fischer, A. Fonari, H. Früchtl, L. Gagliardi, J. Garza, N. Gawande, K. Glaesemann, J. Hammond, V. Helms, E. Hermes, S. Hirata, M. Jacquelin, L. Jensen, B. G. Johnson, H. Jonsson, R. A. Kendall, M. Klemm, R. Kobayashi, S. Krishnamoorthy, M. Krishnan, Z. Lin, R. D. Lins, A. Logsdail, K. Lopata, W. Ma, J. Martin del Campo, D. Mejia-Rodriguez, J. E. Moore, J. M. Mullin, J. A. Nichols, P. Nichols, J. Nieplocha, A. Otero de la Roza, B. Palmer, A. Panyala, T. Pirojsirikul, B. Peng, R. Peverati, J. Pittner, L. Pollack, P. Sadayappan, D. Silverstein, D. M. A. Smith, T. A. Soares, D. Song, M. Swart, H. L. Taylor, G. Thomas, V. Tipparaju, D. G. Truhlar, K. Tsemekhman, T. Van Voorhis, A. Vazquez-Mayagoitia, P. Verma, O. Villa, A. Vishnu, K. D. Vogiatzis, D. Wang, J. H. Weare, M. J. Williamson, T. L. Windus, K. Wolinski, A. Wong, Q. Wu, C. Yang, Q. Yu, M. Zacharias, Z. Zhang, Y. Zhao, and R. J. Harrison, *NWChem: Past, Present, and Future. Journal of Chemical Physics* **152**, article no. 184102. doi.org/10.1063/5.0004997

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, including 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, 2019/P17-C-NW6.8, interfaces *Polyrate* –version 17-C, which is a program for dynamical rate calculations, with the electronic structure program *NWChem* [any version, although version 6.8 (released in 2017) or later is recommended].

Using *NWChemrate* presupposes that the user has both *Polyrate 17-C* and *NWChem* (version 6.8 preferred, but any version should suffice) on the same computer working properly before attempting to run *NWChemrate*. No part of *Polyrate 17-C* 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_mdep.F`, which are specialized versions of the routines in the `hooks.f` file in `polyrate 17`, 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. There is no license fee. 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 `nwrate2019.tar.Z` or `nwrate2019.tar.gz`. The distributed file (either one of the above files) should be uncompressed and untarred in the `Polyrate 17` directory. To uncompress, type:

```
uncompress nwrate2019.tar.Z
```

or

```
gunzip nwrate2019.tar.gz
```

The uncompressed file will be named `nwrate2019.tar`. After extracting the files from the tar file, which can be done with the command:

```
tar -xvf nwrate2019.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:

```

          Polyrate 17
          |
          -----
    |   |   |   |   |   |   |   |   |   |
  doc exe nwrate obj poten script src testo testrun util
          |
          -----
    |   |   |           |           |
  doc script source      testo      testrun
                                   |
                                   -----
                                   |   |   |   |   |
                                   ch4oh ch5 oh3 nh3oh c6h6ch3
                                           |
                                           SS-QRRK
  
```

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:` 5 files comprising the source code for *NWChemrate*:
`alloc_nwr.F90`, `head_nwr.F`, `hooks_nwr.F`,
`module_nwr.F90`, and `hooks_mdepnw.F`

`testo:` This directory contains selected output files of the test runs:

-18 files: ch4ohtr1.fu6, ch4ohtr2.fu6,
ch4ohtr3.fu15, ch4ohtr4.fu15,
ch4ohtr5.fu15, ch4ohtr6.fu15,
ch5tr1.fu15, ch5tr2.fu15,
ch5tr3.fu15, oh3tr1.fu15,
oh3tr2.fu15, oh3tr3.fu15,
oh3tr4.fu15, oh3tr5.fu15,
nh3ohtr1.fu15, R6.fu15,
falloff.out, QRRK_kE.txt

testrun: 2 files: checknwr and run_all.jc, and 5 subdirectories:
ch4oh, ch5, oh3, nh3oh, and c6h6ch3 with the input files for
the test calculations.

SS-QRRK: 2 files: This directory contains the input file and executable file
for an SS-QRRK follow-up calculation: falloff.inp,
FALLOFF.exe

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 `Polyrate 17`.

The installation of the *NWChemrate* program is carried out by running a C shell script, `configure`. The script will compile the `Polyrate 17` 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 `Polyrate 17` 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 calculations¹ by variational transition state theory with multidimensional tunneling contributions (VTST/MT). Full details of the VTST/MT options are given in the *Polyrate* manual, and full details of the methods are given in a review chapter² and review article.³ The potential energy surface is obtained directly by electronic structure calculations employing – for example – density functional theory (DFT), coupled cluster methods (CC) or Møller-Plesset second order perturbation theory (MP2).

The *NWChemrate* program follows the *Polyrate* hooks protocol, which is explained in the *Polyrate* manual. Under the hooks protocol, 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 hooks routines therefore serve as conduits for passing information and control back and forth between *Polyrate* and *NWChem*. (The user may read more about hooks in the *Polyrate* manual.)

¹ Direct dynamics was introduced in “*Ab Initio* Reaction Paths and Direct Dynamics Calculations,” K. K. Baldrige, M. S. Gordon, R. Steckler, and D. G. Truhlar, *Journal of Physical Chemistry* **93**, 5107-5119 (1989). It has been defined [in “Direct Dynamics Calculation of the Kinetic Isotope Effect for an Organic Hydrogen-Transfer Reaction, Including Corner-Cutting Tunneling in 21 Dimensions,” Y.-P. Liu, D.-h. Lu, A. González-Lafont, D. G. Truhlar, and B. C. Garrett, *Journal of the American Chemical Society* **115**, 7806-7817 (1993)] as follows: “In ... direct dynamics, instead of using a pre-defined PEF [potential energy function], all required energies and forces for each geometry that is important for evaluating dynamical properties are obtained directly from electronic structure calculations.”

² "Variational Transition State Theory with Multidimensional Tunneling," A. Fernandez-Ramos, B. A. Ellingson, B. C. Garrett, and D. G. Truhlar, in *Reviews in Computational Chemistry*, Vol. 23, edited by K. B. Lipkowitz and T. R. Cundari (Wiley-VCH, Hoboken, NJ, 2007), pp. 125-232. doi.org/10.1002/9780470116449.ch3

³ “Variational Transition State Theory: Theoretical Framework and Recent Developments,” J. L. Bao and D. G. Truhlar, *Chemical Society Reviews* **46**, 7548-7596 (2017). (This article is part of the themed collection: Chemical Reaction Dynamics.) doi.org/10.1039/C7CS00602K

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.

<i>ehook</i>	Equivalent to the <i>ehook</i> subroutine in the <i>Polyrate</i> hooks. For a given set of coordinates, it calculates the energy using <i>NWChem</i> .
<i>enatin</i>	Calculates the energy for atomic systems.
<i>escener</i>	Carries out <i>NWChem</i> energy calculations using <i>NWChem</i> .
<i>escfirst</i>	Carries out energy and first derivative calculations using <i>NWChem</i> .
<i>escsec</i>	Carries out Hessian calculations using <i>NWChem</i> .
<i>geoin</i>	Carries out geometry optimizations using <i>NWChem</i> .
<i>ghook</i>	Equivalent to the <i>ghook</i> subroutine in the <i>Polyrate</i> hooks. For a given geometry, calculates the energy and first derivatives.
<i>nwinpfl</i>	Creates the <i>NWChem</i> input file for all calculations.
<i>rdnwout</i>	Reads the output file generated by <i>NWChem</i> and extracts the geometry, energy, and first and second derivatives of the energy with respect to atomic coordinates.
<i>grest</i>	Reads and writes the information in the restart files of a <i>NWChemrate</i> restart calculation.
<i>headr</i>	Prints out the program header.

<i>hhook</i>	Equivalent to the <i>hhook</i> subroutine in the <i>Polyrate</i> hooks. For a given geometry, calculates the Hessian matrix using either <i>NWChem</i> or <i>Polyrate</i> methods.
<i>ohook</i>	Equivalent to the <i>ohook</i> subroutine in the <i>Polyrate</i> hooks. Optimizes a given initial geometry, using either <i>Polyrate</i> or <i>NWChem</i> methods.
<i>prep</i>	Prepares the calculation and, depending on the selected options, carries out certain geometry optimizations and Hessian calculations.
<i>prepj</i>	Initialize variables dependent on each stationary point.
<i>rchmul</i>	Reads the charge and multiplicity of a system from the <i>NWChem</i> output file.
<i>rcodnw</i>	Reads the input option for the <i>NWChem</i> runs from file <code>esp.fu70</code> .
<i>redinf</i>	Reads the information about reactants, products and saddle point calculated in the <i>prep</i> subroutine.
<i>rinpnw</i>	Reads the <code>esp.fu70</code> <i>NWChemrate</i> input file.
<i>rlingr</i>	Utility program for reading in a line of the <code>esp.fu70</code> input file and parsing off comments and extra spaces.
<i>rstatnw</i>	Reads the input options given in the <code>esp.fu70</code> input file for the stationary points.
<i>rdnwtop</i>	Reads the <i>NWChem</i> input file section that precedes the geometry.
<i>rdnwmid</i>	Reads the <i>NWChem</i> input file section that after the geometry but before the <i>NWChem</i> task statement(s).

<i>rdnwmethop</i>	Reads the <i>NWChem</i> options for the method (dft, scf, ccsc, etc.) section of the <i>NWChem</i> input file.
<i>rdnwoptop</i>	Reads the <i>NWChem</i> options for the geometry optimization section of the <i>NWChem</i> input file.
<i>rdnwscfop</i>	Reads the <i>NWChem</i> options for the scf section of the <i>NWChem</i> input file for post-Hartree-Fock methods.
<i>rgennw</i>	Reads the general <i>NWChemrate</i> options.
<i>secin</i>	Carries out <i>NWChem</i> Hessian calculations for stationary points.
<i>setup</i>	Initializes some variables and calls the subroutines for reading in the <code>esp.fu70</code> input file.
<i>store</i>	Stores some information that could be useful in future calculations in order to avoid duplicate <i>NWChem</i> 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 poses 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 *NWChemrate*, some of which are 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 6.8 of *NWChem*; 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, coupled cluster methods, and MP2.
- In *NWChem*, a number of methods, *inter alia* meta-GGA functionals, do not have analytical second derivatives (Hessians). 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 2017-C* and *NWChem* and is familiar with their keywords. Therefore, in describing the common input files, we will only discuss the differences between running *Polyrate 2017-C* and running *NWChemrate 2019*.

7.1. FILE USAGE

In addition to the *Polyrate 2017-C* 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.

<code>esp.fu70</code>	General <i>NWChemrate</i> input data.
<code>esp.fu71</code>	<i>NWChem</i> input file for geometry optimization of reactant 1, if such optimization is going to be carried out.
<code>esp.fu72</code>	<i>NWChem</i> 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 <i>NWChem</i> program.
<code>esp.fu73</code>	<i>NWChem</i> input file for geometry optimization of product 1, if such optimization is going to be carried out.
<code>esp.fu74</code>	<i>NWChem</i> 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 <i>NWChem</i> program.
<code>esp.fu75</code>	<i>NWChem</i> input file for geometry optimization of the saddle point or constrained optimization of a reaction path starting geometry, if such optimization is desired.
<code>esp.fu76</code>	reserved
<code>esp.fu77</code>	<i>NWChem</i> 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 <i>NWChem</i> .
<code>esp.fu78</code>	<i>NWChem</i> 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 <i>NWChem</i> .

- `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 19 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 17-C* manual. Here we clarify some points about preparing the input file for unit `fu5`:

RESTART

If a *Polyrate 17-C* 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 17-C* 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 17-C*, 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 17-C* 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 17-C* 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 12.A.6 of the *Polyrate 17-C* 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 17-C* 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 OPTIMIZATION section have to be set to *OHOOKS*, INITGEO has to be set to *HOOKS* and STATUS set to *0* in the corresponding REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and/or START sections. The initial geometries should be provided in the files *f_u71* through *f_u78*.
- 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 *f_u5* input file. The optimized geometry will be read from the *f_u5* 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 *f_u5* input file. The initial geometries are given in the appropriate sections of the *f_u5* 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 *f_u5* 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 *f_u5* 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 *f_u5* 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 17-C*, 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.

<u>Section name</u>	<u>Description</u>
<code>NWRGENERAL</code>	General options
<code>NWRCOMMON</code>	<i>NWChem</i> options common to all the stationary points
<code>NWRREACT1</code>	<i>NWChem</i> options specific to the first reactant
<code>NWRREACT2</code>	<i>NWChem</i> options specific to the second reactant, if any
<code>NWRPROD1</code>	<i>NWChem</i> options specific to the first product
<code>NWRPROD2</code>	<i>NWChem</i> options specific to the second product, if any
<code>NWRWELLR</code>	<i>NWChem</i> options specific to the reactant well, if any
<code>NWRWELLP</code>	<i>NWChem</i> options specific to the product well, if any
<code>NWRSTART</code>	<i>NWChem</i> options specific to the starting (or saddle) point
<code>NWRPATH</code>	<i>NWChem</i> options common to all the nonstationary points along the reaction path
<code>NWRNWREND</code>	signals last line of the <code>esp.fu70</code> 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 NWGENERAL 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 advised 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 B3LYP
  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 a post-HF method is used) from the *NWChem* optimization output. Nevertheless, if no optimization is selected or if *Polyrate 17-C* 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 $V_{\text{MEP}}(s)$, 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 , $V_{\text{MEP}}(s)$, and the gradients on the fine grid, and s , $V_{\text{MEP}}(s)$, $\omega(s)$, $B_{mF}(s)$, and $|\mathbf{I}(s)|$ are stored on the save grid. (The grids are defined in Sect. 7.A of the *Polyrate* manual) Both grids extend from $s = \text{SLM}$ on the reactant side to $s = \text{SLP}$ 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 $\text{EXNSTEP} * \text{EXSTEP}$ points on each side of the grids. Furthermore SLM is replaced by $\text{SLM} - \text{EXNSTEP} * \text{EXSTEP}$, and SLP is replaced by $\text{SLP} + \text{EXNSTEP} * \text{EXSTEP}$.

8. TESTING *NWCHEMRATE*

8.1. TESTS PERFORMED

We tested version 2019 of *NWChemrate* using version 17-C of Polyrate and the indicated versions of *NWChem* on the following machine (the operating system is indicated in parentheses):

HP Haswell Linux Cluster (CentOS) with Intel Haswell E5-2680v3 processors

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* options for direct dynamics. Therefore, the number of tests included is only sixteen, 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.

The test suite is located in the `testrun` directory, which contains five 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 HP Haswell Linux Cluster.) 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
$\text{OH} + \text{H}_2 \rightarrow \text{H}_2\text{O} + \text{H}$	oh3	oh3tr1, oh3tr2, oh3tr3, oh3tr4, oh3tr5
$\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$	ch5	ch5tr1, ch5tr2, ch5tr3
$\text{CH}_4 + \text{OH} \rightarrow \text{CH}_3 + \text{H}_2\text{O}$	ch4oh	ch4ohtr1, ch4ohtr2, ch4ohtr3, ch4ohtr4, ch4ohtr5, ch4ohtr6.
$\text{NH}_3 + \text{OH} \rightarrow \text{NH}_2 + \text{H}_2\text{O}$	nh3oh	nh3ohtr1
$\text{C}_6\text{H}_6\text{CH}_3 \rightarrow \text{C}_6\text{H}_5\text{CH}_3 + \text{H}$	c6h6ch3	R6

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 (B3LYP/3-21G)
Hindered rotation: none
IVTST/M: off

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 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.

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

8.2.2. Test Run `oh3tr2`

Tunneling: none
Number of levels: single level (B3LYP/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).

<i>NWChemrate</i> I/O files	<i>NWChemrate</i> file names	
<code>oh3tr2.70</code>	<code>esp.fu70</code>	Input data for <i>NWChemrate</i>
<code>oh3tr2.75</code>	<code>esp.fu75</code>	Input data for the saddle point
<code>oh3tr2.dat</code>	<code>poly.fu5</code>	Input data for <i>Polyrate</i>
<code>oh3tr2.61</code>	<code>esp.fu61</code>	<i>Polyrate</i> output file containing information about the stationary points
<code>oh3tr2.fu6</code>	<code>poly.fu6</code>	Long output file

oh3tr2.fu15	poly.fu15	Summary output file
oh3tr2.rst.old	esp.fu83	Restart file read by <i>NWChemrate</i>
oh3tr2.rst	esp.fu84	Restart file written by <i>NWChemrate</i>

8.2.3. Test Run oh3tr3

Tunneling: none
Number of levels: single level (B3LYP/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.

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

8.2.4. Test Run oh3tr4

Tunneling: SCT
Number of levels: single level (UMP2/6-31G*)
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 UHF level although MP2 method is specified in .70 file.

<i>NWChemrate</i> I/O files	<i>NWChemrate</i> file names	
oh3tr4.70	esp.fu70	Input data for <i>NWChemrate</i>
oh3tr4.dat	poly.fu5	Input data for <i>Polyrate</i>
oh3tr4.61	esp.fu61	<i>Polyrate</i> 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 <i>NWChemrate</i>

8.2.5. Test Run oh3tr5

Tunneling: SCT
Number of levels: single level (CCSD(T)/cc-pVDZ)
Hindered rotation: none
IVTST/M: off

This is a test run that uses the coupled cluster 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 UHF level although couple cluster method is specified in .70 file.

<i>NWChemrate</i> I/O files	<i>NWChemrate</i> file names	
oh3tr4.70	esp.fu70	Input data for <i>NWChemrate</i>
oh3tr4.dat	poly.fu5	Input data for <i>Polyrate</i>
oh3tr4.61	esp.fu61	<i>Polyrate</i> 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 <i>NWChemrate</i>

8.2.6. Test Run `ch5tr1`

Tunneling: SCT
Number of levels: single level (B3LYP/6-31G)
Hindered rotation: none
IVTST/M: off

This sample run for the reaction $\text{CH}_3 + \text{H}_2 \rightarrow \text{CH}_4 + \text{H}$ is based on the test run `ch5fu30tr5` included in the *Polyrate 17-C* test suite. It performs calculations of the TST and CVT rate constants, and it is based on following the reaction path using the Page-McIver 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`).

<i>NWChemrate</i> I/O files	<i>NWChemrate</i> file names	
<code>ch5tr1.70</code>	<code>esp.fu70</code>	Input data for <i>NWChemrate</i>
<code>ch5tr1.71</code>	<code>esp.fu71</code>	Input data for reactant 1 optimization
<code>ch5tr1.72</code>	<code>esp.fu72</code>	Input data for reactant 2 optimization
<code>ch5tr1.73</code>	<code>esp.fu73</code>	Input data for product 1 optimization
<code>ch5tr1.74</code>	<code>esp.fu74</code>	Input data for product 2 optimization
<code>ch5tr1.dat</code>	<code>poly.fu5</code>	Input data for <i>Polyrate</i>

ch5tr1.61	esp.fu61	<i>Polyr</i> ate 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 <i>NWChemrate</i>

8.2.7. 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.

<i>NWChemrate</i> I/O files		<i>NWChemrate</i> file names	
<code>ch5tr2.51</code>	<code>poly.fu51</code>		Input data for ISPE information
<code>ch5tr2.70</code>	<code>esp.fu70</code>		Input data for <i>NWChemrate</i>
<code>ch5tr2.71</code>	<code>esp.fu71</code>		Input data for reactant 1 optimization
<code>ch5tr2.72</code>	<code>esp.fu72</code>		Input data for reactant 2 optimization
<code>ch5tr2.73</code>	<code>esp.fu73</code>		Input data for product 1 optimization
<code>ch5tr2.74</code>	<code>esp.fu74</code>		Input data for product 2 optimization
<code>ch5tr2.dat</code>	<code>poly.fu5</code>		Input data for <i>Polyrate</i>
<code>ch5tr2.61</code>	<code>esp.fu61</code>		<i>Polyrate</i> output file containing information about the stationary points
<code>ch5tr2.fu6</code>	<code>poly.fu6</code>		Long output file
<code>ch5tr2.fu15</code>	<code>poly.fu15</code>		Summary output file

ch5tr2.rst.old	esp.fu83	Restart file read by <i>NWChemrate</i>
ch5tr2.rst	esp.fu84	Restart file written by <i>NWChemrate</i>

8.2.8. Test Run `ch5tr3`

Tunneling: SCT
Number of levels: single level (B3LYP/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.

<i>NWChemrate</i> I/O files	<i>NWChemrate</i> file names	
<code>ch5tr3.dat</code>	<code>poly.fu5</code>	Input data for <i>Polyrate</i>
<code>ch5tr3.61</code>	<code>esp.fu61</code>	<i>Polyrate</i> output file containing information about the stationary points
<code>ch5tr3.fu6</code>	<code>poly.fu6</code>	Long output file
<code>ch5tr3.fu15</code>	<code>poly.fu15</code>	Summary output file
<code>ch5tr3.fu31</code>	<code>poly.fu31</code>	Input data for IVTST-M algorithm

8.2.9. Test Run `ch4ohtr1`

Tunneling: none
Number of levels: single level (B3LYP/6-31+G**)
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 $\text{CH}_4 + \text{OH} \rightarrow \text{CH}_3 + \text{H}_2\text{O}$ reaction. The sequence actually has five stages. The first four stages may be thought of a complete single-level calculation or as runs 1-4 of a five-run dual-level calculation. The first step we will take is the calculation of the first reactant, CH_4 . In the `fu5` input file we only include the `REACT1` section, and only the additional input files `fu70` and `fu71` are required. The *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.

<i>NWChemrate</i> I/O files	<i>NWChemrate</i> file names	
<code>ch4ohtr1.70</code>	<code>esp.fu70</code>	Input data for <i>NWChemrate</i>
<code>ch4ohtr1.71</code>	<code>esp.fu71</code>	Input data for reactant 1 optimization
<code>ch4ohtr1.dat</code>	<code>poly.fu5</code>	Input data for <i>Polyrate</i>
<code>ch4ohtr1.61</code>	<code>esp.fu61</code>	<i>Polyrate</i> output file containing information about reactant 1
<code>ch4ohtr1.fu6</code>	<code>poly.fu6</code>	Long output file

8.2.10. Test Run `ch4ohtr2`

Tunneling: none
Number of levels: single level (B3LYP/6-31+G**)
Hindered rotation: none
IVTST/M: off

This is the second step 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_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.

<i>NWChemrate</i> I/O files	<i>NWChemrate</i> file names	
<code>ch4ohtr2.70</code>	<code>esp.fu70</code>	Input data for <i>NWChemrate</i>
<code>ch4ohtr2.72</code>	<code>esp.fu72</code>	Input data for reactant 2 optimization
<code>ch4ohtr2.73</code>	<code>esp.fu73</code>	Input data for product 1 optimization
<code>ch4ohtr2.74</code>	<code>esp.fu74</code>	Input data for product 2 optimization
<code>ch4ohtr2.dat</code>	<code>poly.fu5</code>	Input data for <i>Polyrate</i>

ch4ohtr2.61	esp.fu61	<i>Polyrate</i> 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 (B3LYP/6-31+G**)
Hindered rotation: none
IVTST/M: off

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

<i>NWChemrate</i> I/O files	<i>NWChemrate</i> file names	
<code>ch4ohtr3.70</code>	<code>esp.fu70</code>	Input data for <i>NWChemrate</i>
<code>ch4ohtr3.75</code>	<code>esp.fu75</code>	Input data for the saddle point optimization
<code>ch4ohtr3.dat</code>	<code>poly.fu5</code>	Input data for <i>Polyrate</i>
<code>ch4ohtr3.61</code>	<code>esp.fu61</code>	<i>Polyrate</i> output file containing information about the stationary points
<code>ch4ohtr3.fu6</code>	<code>poly.fu6</code>	Long output file

8.2.12. Test Run `ch4ohtr4`

Tunneling: SCT
Number of levels: single level (B3LYP/6-31+G**)
Hindered rotation: mode 14
IVTST/M: off

This is the fourth and final step 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 Page-McIver 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).

<i>NWChemrate</i> I/O files	<i>NWChemrate</i> file names	
<code>ch4ohtr4.70</code>	<code>esp.fu70</code>	Input data for <i>NWChemrate</i>
<code>ch4ohtr4.dat</code>	<code>poly.fu5</code>	Input data for <i>Polyrate</i>
<code>ch4ohtr4.61</code>	<code>esp.fu61</code>	<i>Polyrate</i> output file containing information about the stationary points
<code>ch4ohtr4.fu6</code>	<code>poly.fu6</code>	Long output file
<code>ch4ohtr4.fu15</code>	<code>poly.fu15</code>	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 $\text{CH}_4 + \text{OH} \rightarrow \text{CH}_3 + \text{H}_2\text{O}$ 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 (μOMT), 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.

<i>NWChemrate</i> I/O files	<i>NWChemrate</i> file names	
<code>ch4ohtr5.70</code>	<code>esp.fu70</code>	Input data for <i>NWChemrate</i>
<code>ch4ohtr4.rst</code>	<code>esp.fu83</code>	Restart file read by <i>NWChemrate</i>
<code>ch4ohtr5.dat</code>	<code>poly.fu5</code>	Input data for <i>Polyrate</i>
<code>ch4ohtr5.50</code>	<code>poly.fu50</code>	Additional input data for the <i>Polyrate</i> VTST-IC calculation

ch4ohtr5.61	esp.fu61	<i>Polyrate</i> 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 <i>NWChemrate</i>

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 $\text{CH}_4 + \text{OH} \rightarrow \text{CH}_3 + \text{H}_2\text{O}$ reaction. The lower-level information is read from the restart file from the `ch4ohtr4` run. The higher-level data is included by means of the `fu50` file. Tunneling is calculated by the microcanonical optimized multidimensional tunneling method (μOMT), 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.

<i>NWChemrate</i> I/O files	<i>NWChemrate</i> file names	
<code>ch4ohtr6.70</code>	<code>esp.fu70</code>	Input data for <i>NWChemrate</i>
<code>ch4ohtr4.rst</code>	<code>esp.fu83</code>	Restart file read by <i>NWChemrate</i>
<code>ch4ohtr6.dat</code>	<code>poly.fu5</code>	Input data for <i>Polyrate</i>
<code>ch4ohtr6.50</code>	<code>poly.fu50</code>	Additional input data for the <i>Polyrate</i> VTST-IC calculation
<code>ch4ohtr6.61</code>	<code>esp.fu61</code>	<i>Polyrate</i> 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 <i>NWChemrate</i>

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 $\text{NH}_3 + \text{OH} \rightarrow \text{NH}_2 + \text{H}_2\text{O}$ 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.

<i>NWChemrate</i> I/O files	<i>NWChemrate</i> file names	
<code>nh3ohtr1.70</code>	<code>esp.fu70</code>	Input data for <i>NWChemrate</i>
<code>nh3ohtr1.71</code>	<code>esp.fu71</code>	Input data for reactant 1 optimization
<code>nh3ohtr1.72</code>	<code>esp.fu72</code>	Input data for reactant 2 optimization
<code>nh3ohtr1.73</code>	<code>esp.fu73</code>	Input data for product 1 optimization
<code>nh3ohtr1.74</code>	<code>esp.fu74</code>	Input data for product 2 optimization
<code>nh3ohtr1.75</code>	<code>esp.fu75</code>	Input data for the saddle point optimization
<code>nh3ohtr1.77</code>	<code>esp.fu77</code>	Input data for the reactant well optimization
<code>nh3ohtr1.78</code>	<code>esp.fu78</code>	Input data for the product well optimization
<code>nh3ohtr1.dat</code>	<code>poly.fu5</code>	Input data for <i>Polyrate</i>

nh3ohtr1.61	esp.fu61	<i>Polyr</i> ate output file containing information about the stationary points
nh3ohtr1.fu6	poly.fu6	Long output file
nh3ohtr1.fu15	poly.fu15	Summary output file

8.2.16. Test Run R6

Tunneling: none
Number of levels: single level (B3LYP/6-31G*)
Hindered rotation: none
IVTST/M: off

This sample run is for the reaction $C_6H_6CH_3 \rightarrow C_6H_5CH_3 + H$, where $C_6H_6CH_3$ is methylcyclohexadienyl radical (the adduct of H and toluene). A scale factor of 0.977 for vibrational frequencies is used in the computation. This run computes the high-pressure-limit rate constants; pressure-dependent rate constant can then be computed by the SS-QRRK utility program of *Polyrate* (please see the SS-QRRK manual in the *Polyrate* distribution); the input file for the following SS-QRRK calculation is put in the subdirectory /SS-QRRK together with the executable file for the SS-QRRK calculation.

<i>NWChemrate</i> I/O files		<i>NWChemrate</i> file names
R6.70	esp.fu70	Input data for <i>NWChemrate</i>
R6.71	esp.fu71	Input data for reactant 1 optimization
R6.73	esp.fu73	Input data for product 1 optimization
R6.74	esp.fu74	Input data for product 2 optimization
R6.75	esp.fu75	Input data for the saddle point optimization
R6.dat	poly.fu5	Input data for <i>Polyrate</i>
R6.61	esp.fu61	<i>Polyrate</i> output file containing information about the stationary points
R6.fu6	poly.fu6	Long output file
R6.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 user time on eight processors.

The following timings apply to version 2019/P17-C/-NW6.8. These calculations are all run on eight processors.

HP Haswell Linux Cluster (CentOS) with Intel Haswell E5-2680v3 processors

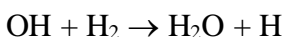
Test run ch5tr1	5036.9
Test run ch5tr2	0.2
Test run ch5tr3	1036.1
Test run oh3tr1	222.0
Test run oh3tr2	962.2
Test run oh3tr3	0.0
Test run oh3tr4	351.9
Test run oh3tr5	78590.2
Test run ch4ohtr1	100.5
Test run ch4ohtr2	450.6
Test run ch4ohtr3	182.6
Test run ch4ohtr4	5433.3
Test run ch4ohtr5	6066.2
Test run ch4ohtr6	7512.5
Test run nh3ohtr1	370.4
Test run R6	107780.9

8.4. COMPARISON BETWEEN *NWChemrate* AND *Gaussrate*

Except for test run oh3tr5, all the other test runs have been compared with the results obtained by performing the same calculation using *Gaussrate*, and they agreed well with each other. Here are two examples to show the comparison of the results.

Example 1: test run oh3tr4

The reaction is



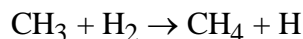
The forward rate constants were calculated at the MP2/6-31G* level with 0.015 Å for the step size of s (scaled to 1 amu) and with an s range from -0.60 – 0.60 Å. The forward rate constants are shown in the following table.

Forward rate constants ($\text{cm}^3 \text{ molecule}^{-1} \text{ s}^{-1}$)

T(K)	TST	CVT	CVT/ZCT	CVT/SCT
<i>Gaussrate 17-B</i>				
200	3.91E-25	1.16E-25	2.22E-19	1.21E-18
300	1.43E-20	7.39E-21	6.16E-19	2.14E-18
600	5.57E-16	4.83E-16	1.01E-15	1.27E-15
1000	5.07E-14	5.02E-14	5.58E-14	6.05E-14
<i>NWChemrate 2019</i>				
200	3.90E-25	1.16E-25	2.22E-19	1.21E-18
300	1.43E-20	7.39E-21	6.18E-19	2.14E-18
600	5.57E-16	4.82E-16	1.01E-15	1.27E-15
1000	5.06E-14	5.02E-14	5.57E-14	6.04E-14

Example 2: test run ch5tr1

The reaction is



The forward rate constants were calculated at the B3LYP/6-31G level with 0.01 Å for the step size of s (scaled to 1 amu) and with an s range from -1.00 – 1.00 Å. The forward rate constants are shown in the following table.

Forward rate constants ($\text{cm}^3 \text{ molecule}^{-1} \text{ s}^{-1}$)

T(K)	TST	CVT	CVT/ZCT	CVT/SCT
<i>Gaussrate 17-B</i>				
298	8.20E-19	7.23E-19	2.01E-18	2.55E-18
400	3.54E-17	3.28E-17	5.56E-17	6.21E-17
667	3.01E-15	2.96E-15	3.48E-15	3.60E-15
1000	3.39E-14	3.39E-14	3.54E-14	3.60E-14
1550	2.68E-13	2.67E-13	2.66E-13	2.68E-13
<i>NWChemrate 2019</i>				
298	8.17E-19	7.22E-19	2.00E-18	2.54E-18
400	3.53E-17	3.28E-17	5.55E-17	6.20E-17
667	3.01E-15	2.96E-15	3.48E-15	3.60E-15
1000	3.39E-14	3.39E-14	3.55E-14	3.60E-14
1550	2.68E-13	2.68E-13	2.67E-13	2.68E-13

The numbers in the tables show that two programs give essentially 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**

Machine (Operating system)	<i>NWChem</i> Version
2007/P9.7-NW5.0	
IBM Regatta, Power4 processors (AIX 5.3)	5.0
IBM BladeCenter AMD Opteron processors (SuSE Linux)	5.0
SGI-Altix, Itanium 2 processors (SuSE Linux)	5.0
2019/P17-C-NW6.8	
HP Haswell Linux Cluster (CentOS) with Intel Haswell E5-2680v3 processors (CentOS Linux)	6.8

10. VERSION INFORMATION

11.1. *NWChemrate* VERSION NUMBERS

NWChemrate version numbers have the long form 20zw/Pv'-NWv'' where 20zw is the *NWChemrate* version number, z' is version of *Polyrate* to which it is linked, and v'' is the number of the version of *NWChem* being used. 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

Version 2007 (long version number: 2007/P9.7-NW5.0)

This is the initial release version of *NWChemrate*.

Version 2019 (long version number: 2019/P17-C-NW6.8)

This version is based on new versions of *Polyrate* and *NWChem*, and it includes additional test runs.

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 supported most of the research on VTST/MT) and the Office of Naval Research (which supported the some of the development of integrated software tools).