Manual
MC-TINKERATE – version 2010/P9.1–M2010

Titus V. Albu, José C. Corchado, Yongho Kim, Jordi Villà,
Jianhua Xing, Hai Lin, Osanna Tishchenko, Masahiro Higashi, and Donald G. Truhlar
Department of Chemistry and Supercomputer Institute
University of Minnesota, Minneapolis, Minnesota 55455

Program version: 2010/P9.1–M2010
Program version date: October 4, 2010
Manual version date: July 31, 2021

Abstract:

MC-TINKERATE is a computer program for carrying out calculations of chemical reaction rates of polyatomic species by using single-configuration molecular mechanics (SCMM) methods or multi-configuration molecular mechanics (MCMM) methods available in MC-TINKER to calculate the potential energy surface and by using POLYRATE for the dynamics. The interface between POLYRATE and MC-TINKER is based on the POLYRATE hooks protocol. The dynamical methods used are variational or conventional transition state theory and multidimensional semiclassical approximations for tunneling and nonclassical reflection. Rate constants may be calculated by any of the methods available in the POLYRATE program for canonical or microcanonical ensembles or for specific vibrational states of selected modes with translational, rotational, and other vibrational modes treated thermally. MC-TINKER is based on TINKER by J. W. Ponder. In MC-TINKER both SCMM and MCMM calculations are available. In SCMM calculations, potential energies, gradients, and Hessians can be calculated by any of the well-established methods in TINKER; however, such methods are not useful for investigating processes involving bond breaking, bond making, or bond rearrangement. In MCMM calculations, the diagonal elements of the MCMM configuration interaction matrix are calculated on the fly by MC-TINKER (using TINKER routines), and the off-diagonal elements of the MCMM configuration interaction matrix are fitted to reproduce electronic structure theory data that are read from an external file. In this respect, MCMM serves as an advanced form of direct dynamics.

LICENSING

Publications of results obtained with the *MC-TINKERATE* - version 2010 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 *MC-TINKERATE* - version 2010 implies acceptance of the terms of the licenses.

CONTENTS

1. REQUIRED REFERENCES

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

1 T. V. Albu, O. Tishchenko, J. C. Corchado, Y. Kim, J. Villà, J. Xing, H. Lin, M. Higashi and D. G. Truhlar, MC-TINKERATE–version 2008, University of Minnesota, Minneapolis, MN, 2008.

2 J. C. Corchado, Y.-Y. Chuang, P. L. Fast, J. Villà, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, V. S. Melissas, B. J. Lynch, I. Rossi, E. L. Coitiño, A. Fernandez-Ramos, J. Pu, T. V. Albu, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar, POLYRATE–version 9.1, University of Minnesota, Minneapolis, MN, 2002.

3 O. Tishchenko, M. Higashi, T. V. Albu, J. C. Corchado, Y. Kim, J. Villà, J. Xing, H. Lin, and D. G. Truhlar, MC-TINKER–version 2008, University of Minnesota, Minneapolis, MN, 2008.

4 J. W. Ponder, TINKER–version 3.5, Washington University, St. Louis, MO, 1997.

For example: "The calculations were carried out using the MC-TINKERATE[1] multi-configuration molecular mechanics computer program, which interfaces the POLYRATE[2] and MC-TINKER[3] programs. MC-TINKER[3] is built on single-configuration molecular mechanics subroutines from the TINKER[4] computer program."

The references are required; the wording is optional.

2. INTRODUCTION

The MC-TINKERATE program interfaces the POLYRATE program, which is a program for dynamics calculations based on variational transition state theory with semiclassical multidimensional tunneling contributions, and the MC-TINKER program, which provides the potential energy surface used in the dynamics calculations. In order to calculate forward and reverse reaction rates, POLYRATE requires potential energy surface information for reactants, for products, and in the reaction swath, which is a broad region between reactants and products. The reaction swath region includes the saddle point, the reaction-path valley in the vicinity of the variational transition state, and the region containing significant large-curvature tunneling paths. In the case of small-curvature tunneling, the tunneling path region is in the zero-point-motion part of the valley around the minimum-energy reaction path, but in the more general case that allows for large-curvature tunneling, it is broader.

The MC-TINKER program is a composite of the computer program TINKER, which is program for single-configuration molecular mechanics calculations, and the MCT module, which is a module for multi-configuration molecular mechanics calculations. The MC-TINKER program can carry out two types of calculations: calculations based on single-configuration molecular mechanics (MM or, for emphasis, SCMM), as in TINKER itself, or calculations based on multi-configuration molecular mechanics (MCMM). (Throughout this manual, we use MM and SCMM interchangeably as synonyms.)

The most flexible method in MC-TINKER is the MCMM method [1-3]. This technique is mainly based on the combination of four computational techniques: the semiempirical valence bond method [4-14], Chang, Minichino, and Miller's method of estimating V_{12} for semiempirical valence bond calculations [15, 16], the use of redundant internal coordinates [17, 18], and the Shepard interpolation method [19, 20]. The method of Coulson and Danielsson [7] and the empirical valence bond (EVB) method, which is an approach to semiempirical valence bond theory that is widely used by Warshel and coworkers [12-14], may also be considered to be molecular-mechanics-based ways to parameterize valence bond theory, as alternatives to the quantum mechanical based approaches of Eyring, Polanyi, Sato, and Ellison [5, 8-10]. Another relevant reference is the work of Downing and Michl [21], who pointed out the possibility that the matrix elements of a nondiagonal Hamiltonian representation of the potential energy surfaces might be more amicable to approximation by simple functional forms than are the eigenvalues (which are the adiabatic potential energy surfaces). A detailed description of the MCMM method is given in the original papers [1–3], and therefore we will only present a brief overview. We note, however, that the overview in Section 2.1 employs a more general perspective than was used in Refs. [1] and [3].

Since MC-TINKERATE is focused on the calculation of reaction rates with POLYRATE, we do not emphasize the use of nuclear permutation symmetry in this manual, but this is discussed elsewhere [2].

A recent development in using MCMM for reaction rate calculations is the use of partial Hessians for the supplementary points. This is discussed in a recent publication [29].

2.1. Short description of multi-configuration molecular mechanics method

In MCMM, we represent a nonstationary point on a potential energy surface by using two configurations. For a chemical reaction, these two configurations correspond to the reactant and product configurations of that reaction. The energy for any point on the potential energy surface is approximated by the lowest root V of the equation

$$\begin{vmatrix} V_{11} - V & V_{12} \\ V_{12} & V_{22} - V \end{vmatrix} = 0 \qquad (2.1)$$

where V_{11} and V_{22} are the energy functions of the two configurations, and V_{12} is the resonance integral, also called the resonance energy function or the resonance interaction.

The V_{11} and V_{22} functions are approximated by standard molecular mechanics methods. The approximation of V_{12} is less obvious. Several approaches of determining V_{12} are available in MC-TINKER; the most advanced of these approaches uses a multi-point extension [1–3] of a variant of the method described in Refs. [14, 15]. The key element in the method of Refs. [14, 15] is that V_{12} is chosen locally to reproduce a Taylor series approximation to V at a point. We apply this at a set of points, called Shepard points, and we estimate V_{12} at other points in the interaction region by using Shepard interpolation to join several such local approximations to provide a semiglobal approximation.

For the Shepard interpolation of V_{12}, the MCMM method uses a number of electronic structure theory Shepard points and two single-configuration molecular mechanics Shepard points, the latter two points being called the SCMM point for configuration 1 and the SCMM point for configuration 2. While there are no restrictions about the location of the electronic structure theory Shepard points, it is expected that these points will be placed in regions where neither of the molecular mechanics potentials are accurate. At the two SCMM points we define V_{12} (as well as its first and second derivatives with respect to coordinates) to be zero so the potential V reduces to V_{11} or V_{22}, whichever is lower in energy. These points should be therefore chosen in regions where the configuration 1 and configuration 2 molecular mechanics potential functions, respectively, are accurate.

The Taylor series expansion of V_{12} is used to interpolate between the electronic structure theory points to calculate V_{12} for any geometry of interest on the potential energy surface. To make the approach applicable for any number of electronic structure theory points, we use a Shepard interpolation method that is a modification of the scheme described in Ref. [20], which is itself a revised version of the method in Ref. [19]. This Shepard scheme ensures that the potential energy surface is a continuous function (with continuous first and second derivatives) that exactly reproduces the energies, gradients, and Hessians at the electronic structure theory Shepard points. The procedure allows the results to be improved by adding more electronic structure theory points, eventually converging (in principle and with sufficient care) to the potential function that would be obtained from a dense set of electronic structure theory results. In order to make the method rotationally and translationally invariant, the interpolation procedure is carried out in internal coordinates, in particular in redundant or nonredundant internal coordinates. Note that internal coordinates appear in the MCMM method in three different contexts:

(a) the set **q** used to evaluate MM energies and derivatives,
(b) the set **r** used for the Shepard interpolation step, and
(c) the set **s** used to evaluate the weight function.

The distinction between these coordinates was not explicitly mentioned in Refs. [1, 3], but it is stressed in Ref. [2] and in the Appendix to the MC-TINKER–version 2008 manual. In Refs. [1, 3], the notation **q** was used for all three sets. While the composition of the set **q** is completely determined by the molecular mechanics method, sets **r** and **s** are introduced in MCMM. Both sets **r** and **s** should be specified by the user in the input esp.fu85 file. The current version of the code only supports the following cases:

(i) sets **r** and **s** are the same
(ii) set **s** is a subset of **r**

Other cases require additional considerations.

Given the geometry **x** of a point in Cartesians, for which we want to obtain the MCMM energy, gradient, and Hessian, the algorithm carries out the following steps:

A. Transformation of the geometries, gradients, and Hessians of the Shepard points from Cartesian to redundant or nonredundant internal coordinates **r** as are used in step A.

B. Calculation of V_{12} and its first and second derivatives in internal coordinates at all geometries for which electronic structure theory gradient and Hessian data are available.

C. Transformation of an input geometry $\mathbf{x}$ in Cartesians to the same set of redundant or nonredundant internal coordinates $\mathbf{r}(\mathbf{x})$.

D. Estimation of V_{12} and its first and second derivatives with respect to internal coordinates at the geometry $\mathbf{r}(\mathbf{x})$ by means of Shepard interpolation, which involves a weight function written in terms of coordinates $\mathbf{s}(\mathbf{x})$.

E. Transformation of the internal-coordinate derivatives of V_{12} at $\mathbf{r}(\mathbf{x})$ to Cartesian coordinates by the inverse of the transformation in step C (note that using the inverse transformation preserves the correct orientation in Cartesian coordinates).

F. Calculation of V_{11} and V_{22} and their derivatives at the geometry $\mathbf{x}$ in Cartesian coordinates (TINKER evaluates the derivatives in internal coordinates $\mathbf{q}$ and then transforms them to Cartesian coordinates).

G. Calculation of $V(\mathbf{x})$ and its first and second derivatives in Cartesian coordinates.

2.2. Zero of energy in multi-configuration molecular mechanics method

Special attention should be paid to the zero of energy in an MCMM calculation. The molecular mechanics (MM) force field and the electronic structure (ES) calculation each have their own origin for the zero of energy. The MM energy of a molecule is the sum of several terms. For some small molecules like OH, CH_4, H_2O, and CH_3, the MM energies for the optimized structures are zero. Thus, for example, the TINKER energy of reaction for the OH + $CH_4 \rightarrow H_2O + CH_3$ reaction is zero, because both the TINKER reactant state energy of OH + CH_4 and the TINKER product state energy of H_2O + CH_3 are zero. This does not mean that the actual energy of reaction is zero, though, because one cannot compare the MM energies for different molecules that have different atom types and connectivities, since they are from different MM potential functions. For more complicated molecules, such as propane, the MM energy is not necessarily zero even in the optimized structure.

To connect the MM energies for the two configurations with the ES energies, a unique scale of energies should be defined. This energy scale is defined in the MCMM calculations based on the MM and ES energies at some states (or geometries) that we call energy reference states. The program runs if the energy reference states are chosen at any arbitrary state or geometry. We alert the user though that it is more physical to choose the energy reference states where the molecular mechanics potentials are relatively accurate, i.e., close to equilibrium structures. (Note that the energy reference states do not need to be the same as the SCMM points, which are the single-configuration molecular mechanics points used in Shepard interpolation.)

The MM evaluations of the V_{11} and V_{22} matrix elements at these states (obtained using TINKER) have the following values:

$$V_{11}(\text{configuration 1 energy reference state}) = \mathtt{vz1}^{\text{MM}} \tag{2.2}$$

$$V_{22}(\text{configuration 2 energy reference state}) = \mathtt{vz2}^{\text{MM}} \tag{2.3}$$

For emphasis we used superscript MM to denote that the value are obtained through molecular mechanics calculations. As stated above, $\mathtt{vz1}^{\text{MM}}$ and $\mathtt{vz2}^{\text{MM}}$ may be both zero for some simple cases if the energy reference states are the optimized structures but in the general case may have any value.

The electronic structure theory energies, which are the high-level energies, from GAUSSIAN or other electronic structure packages satisfy:

$$V(\text{configuration 1 energy reference state}) = \mathtt{Vz1}^{\mathrm{ES}} \tag{2.4}$$

$$V(\text{configuration 2 energy reference state}) = \mathtt{Vz2}^{\mathrm{ES}} \tag{2.5}$$

We need to have this energy scale consistent with the one obtained from MM calculations. We defined the zero of energy to be the electronic structure theory energy at the configuration 1 energy reference state. Therefore:

$$V(\text{configuration 1 energy reference state}) = \mathtt{0} \tag{2.6}$$

$$V(\text{configuration 2 energy reference state}) = \mathtt{ediff0} \tag{2.7}$$

where *ediff0* is given by:

$$\mathtt{ediff0} = \mathtt{Vz2}^{\mathrm{ES}} - \mathtt{Vz1}^{\mathrm{ES}} \tag{2.8}$$

and it is obtained only from electronic structure calculations. Then, whenever MC-TINKER determines a MM energy, V_{11} or V_{22}, these energies will be adjusted as:

$$V_{11}(\text{MC-TINKER}) = V_{11}(\text{TINKER}) - \mathtt{vz1}^{\mathrm{MM}} \tag{2.9}$$

$$V_{22}(\text{MC-TINKER}) = V_{22}(\text{TINKER}) - \mathtt{vz2}^{\mathrm{MM}} + \mathtt{ediff0} \tag{2.10}$$

Furthermore, when an electronic structure energy V is used in an MCMM calculation, we use the adjusted energy value given by:

$$V(\text{MC-TINKER}) = V^{\mathrm{ES}} - \mathtt{Vz1}^{\mathrm{ES}} \tag{2.11}$$

Later in this manual, we sometimes use another notation, in particular, *vz1* is called ZERO1, *vz2* is called ZERO2, and *ediff0* is called EDIFF (ZERO1, ZERO2, and EDIFF are the keywords whose arguments are *vz1*, *vz2*, and *ediff0*).

2.3. Rate calculations with MCMM and VTST/MT

Rate constants are calculated by variational transition state theory (VTST) with multidimensional tunneling (MT) [18, 22-26]. Such calculations require potential energy surface (PES) information for generalized transition states (GTSs), points in the tunneling swath, and reactants (also products if reverse rate constants are desired). There are several possibilities:

		PES information for	
Reaction type	Option	reactants	GTSs and tunneling
unimolecular	1.1	E.S.	MCMM
	1.2	SCMM	MCMM
	1.3	MCMM	MCMM
bimolecular	2.1	E.S.	MCMM
	2.2	SCMM	MCMM
	2.3	MCMM	MCMM

where E.S. denotes electronic structure data.

The present code supports all these options. However, some clarifications should be made. The POLYRATE program allows various options for entering the PES information for the reactants (and/or products). One of these options is to enter the information required for a rate calculation (i.e., equilibrium geometry, vibrational frequencies, etc) in a POLYRATE input file, `poly.fu5`. For this information one could use values obtained (in calculations that should precede the rate calculation) by either E.S. or MCMM or SCMM methods, or one could use experimental values. If this option of obtaining reactant information is used, than all of the tabulated options are available. Another option of obtaining reactant information is to provide the program with an initial geometry that will be first optimized, then the program will calculates the properties of this equilibrium structure. For this option of obtaining the reactant information only the tabulated option 1.3 is available. Indirectly, options 1.1 and 1.2 can be also obtained if the equilibrium geometry of the reactant is used as a Shepard point (either molecular mechanics or electronic structure theory Shepard point). In this later case, the MCMM results will reproduce, at that geometry, either the molecular mechanics or the electronic structure results.

3. DISTRIBUTION

MC-TINKERATE is a set of subroutines for interfacing POLYRATE and MC-TINKER. The user should obtain three items of code:

- POLYRATE The POLYRATE code is a licensed code available form the University of Minnesota. For further information see: `http://comp.chem.umn.edu/polyrate`

- MC-TINKER The MC-TINKER code is a licensed code available form the University of Minnesota. For further information see: `http://comp.chem.umn.edu/mc-tinker`

- MC-TINKERATE The MC-TINKERATE code is a licensed code available form the University of Minnesota. For further information see: `http://comp.chem.umn.edu/mc-tinkerate`

The prospective user does not need to obtain TINKER because a complete version of TINKER is included in MC-TINKER. The prospective user of MC-TINKERATE must obtain and install POLYRATE and MC-TINKER before proceeding with MC-TINKERATE.

The MC-TINKERATE program is distributed as a compressed tar file named `mc-tinkerate2010.tar.gz`. To uncompress, enter:

```
gunzip mc-tinkerate2010.tar.gz
```

The uncompressed file will be named `mc-tinkerate2010.tar`. After extracting the files from the tar file, which can be done with the command:

```
tar -xvf mc-tinkerate2010.tar
```

a new directory, `mc-tinkerate2010`, is created. This directory contains all the files included in the distribution package, which are located in subdirectories according to the following tree structure:

```
                       mc-tinkerate9.1
                               |
-------------------------------------------------
 |        |         |        |        |
docmcr  exemcr  scriptmcr  srcmcr  testmcr
                             |        |
                             -      ------------------------
                             |     |     |    |    |       |
                          dattim test1 test2 ... testo testo2010
```

`docmcr`: 1 file: `mc-tinkerate2010.pdf`. This file is the MC-TINKERATE manual.

`exemcr`: no files. (After the compilation, the executable will be moved to this directory.)

`scriptmcr`: 4 files containing compilation scripts for MC-TINKERATE: `compile.compaq`, `compile.ibm`, `compile.ifort`, and `compile.sgi`

`srcmcr`: 6 files and 1 subdirectory.

(i) The 6 files comprising the source code: `headr.f`, `hooks.f`, `mcrate.inc`, `mrsrc1.f`, `mrsrc2.f`, `param.inc`. The `mcrate.inc`, `mrsrc1.f`, and `mrsrc2.f` files comprises the MC-TINKERATE source code, and the `headr.f`, `hooks.f` and `param.inc`. are modified POLYRATE files that will be used in compiling the program instead of the files distributed in the POLYRATE source code.

(ii) Directory `dattim` collects the MC-TINKERATE machine-dependent subroutine files: `dattim.linux3_tr`, `mcdatm.compaq_tr`, `mcdatm.cray_tr`, `mcdatm.dec_tr`, `mcdatm.hp_tr`, `mcdatm.ibm_tr`, `mcdatm.iris_tr`, `mcdatm.linux_tr`, `mcdatm.linux2_tr`, `mcdatm.linux3_tr`, and `mcdatm.sun_tr`.

`testmcr`: 4 files: `run_all`, `run_all_FC`, `getnew`, and `compnew`, and 20 subdirectories: (i) 18 directories (`test1`, `test2`, ..., and `test18`) with the input files for the test calculations, (ii) one directory (`testo`) with the distributed output files of the test runs, and (iii) one directory (`testonew`) without files. Script `run_all` does tests 1~17 while script `run_all_FC` does tests 1~18. Running script `getnew` will collect the most recent outputs for all test calculations into directory `testonew`, and subsequently running script `compnew` will generate files for comparisons between these newly obtained and those distributed outputs for all test runs.

4. INSTALLATION

First the user should obtain licensed copies and install the POLYRATE and MC-TINKER packages. Next the user should obtain a licensed copy of the MC-TINKERATE package. After downloading, uncompressing, and untarring the file (as discussed in Section 3 of this manual), the MC-TINKERATE package should appear as a directory, as described in Section 3.

There are a few variables that have to be set large enough to accommodate the system or systems to be studied. These variables are set in include files, either on the MC-TINKER or on the POLYRATE side of the interface. The user should consult the MC-TINKER and POLYRATE manuals for additional details; in addition, we note that these variables are explained in comment lines in the include files themselves. Note that the modified POLYRATE include file, `param.inc`, distributed in the MC-TINKERATE source code is used instead of the file distributed in the POLYRATE source code in order to assure consistency between the sizes of various arrays on both sides of the interface.

The installation of the MC-TINKERATE program is carried out by running a C shell script, in particular, `compile.ibm` for IBM computers, `compile.sgi` for SGI computers, or `compile.compaq` for Compaq computers. To run the script, the user should also give the paths of the `mc-tinker2010` and the `mc-tinkerate2010` directories. For example, if the `mc-tinker2010` path is `/usr/mc-tinker2010`, and the `mc-tinkerate2010` path is `/usr/mc-tinkerate2010` than the command should be

```
compile.ibm /usr/mc-tinker2010 /usr/mc-tinkerate2010
```

on an IBM machine,

```
compile.sgi /usr/mc-tinker2010 /usr/mc-tinkerate2010
```

on an SGI Origin machine, or

```
compile.compaq /usr/mc-tinker2010 /usr/mc-tinkerate2010
```

on a Compaq machine. These scripts will compile the source code and will generate an executable file. In Section 9 we list the computers and operating systems on which the code has been tested. The installation of MC-TINKERATE on a different kind of computer or operating system should be straightforward, requiring only changes in the compiler and loader options in any one of the scripts, `compile.ibm`, `compile.sgi`, or `compile.compaq`, as well as the machine-specific timing subroutine that can be obtained from the MC-TINKER and POLYRATE source codes. After installation, the executable file is put into the `exemcr` subdirectory of the `mc-tinkerate2010` directory.

5. PROGRAM DESCRIPTION

The MC-TINKERATE program follows the POLYRATE hooks protocol. This means that if energy, gradient, and/or Hessian calculations are desired by POLYRATE, the program calls the appropriate hooks subroutine that will return that information. (The user may read more about hooks in the POLYRATE manual, although everything that is required to be known to use MC-TINKERATE is given here.) The MC-TINKERATE program interfaces the POLYRATE hooks routines with the MC-TINKER (more specifically, with the MCT module) hooks subroutines.

5.1. Structure of MC-TINKERATE

When energy, gradient, and/or Hessian information is desired, the POLYRATE program calls the appropriate hooks subroutine (the *ehook*, *ghook*, and/or *hhook* subroutines, respectively) that will return that information. In addition, POLYRATE calls the *ohook* subroutine for a geometry optimization, the *prep* subroutine for some initializations, and the *prepj* subroutine for initializations that are species dependent.

The *ehook*, *ghook*, *hhook*, *prep*, and *prepj* hooks subroutines of POLYRATE call only MC-TINKERATE hooks subroutines (*mrehook*, *mrghook*, *mrhhook*, *mrprep*, and *mrprepj*) and do not directly call MC-TINKER hooks subroutines (*mehook*, *mghook*, and *mhhook*). The *ohook* POLYRATE hooks subroutine calls only POLYRATE subroutines and no MC-TINKERATE or MC-TINKER subroutines because no geometry optimization options are available in the current version of MC-TINKER. We also mention that the variables are transferred between the POLYRATE hooks subroutines and the MC-TINKERATE hooks subroutines only through parameter lists, whereas the transfer of variables between the MC-TINKERATE hooks subroutines and the MC-TINKER subroutines is accomplished both through parameter lists and through common blocks.

The first MC-TINKERATE hooks subroutine that POLYRATE calls is the *mrprep* subroutine. The *mrprep* subroutine makes the initializations on the MC-TINKER side of the interface. The first MC-TINKER subroutine called in *mrprep* is subroutine *mcfope*, which opens the `esp.fu85` input file. This subroutine sets the defaults and reads the various options of the MCMM/SCMM calculations (in subroutine *mcrfgi*) and then closes the `esp.fu85` input file (by calling subroutine *mcfclo*). Next, the program prints out a summary table of all the options in the MCMM/SCMM calculation (in subroutine *mcptab*), and it checks if the variables read on the POLYRATE and MC-TINKER sides of the interface are consistent. The next step is preparing for the calculation, and this is carried out in subroutine *mcscal*. The *mcscal* subroutine is described in more detail in the MC-TINKER manual. Finally, some variables are saved on the MC-TINKER side of the interface, and, if necessary, the interface files are read in. Note that the *mrprep* subroutine is called only once prior to any calculation.

In the current version of the code, the other MC-TINKERATE hooks subroutines (*mrehook*, *mrghook*, *mrhhook*, and *mrprepj*) are very simple, having either only one call to the appropriate MC-TINKER subroutine (as is the case of *mrehook*, *mrghook*, and *mrhhook*) or only some initializations (in the case of *mrprepj*).

6. FILE USAGE

The MC-TINKERATE program uses several files for input and output data. The names of these files are of the form `esp.fu#` or `poly.fu#`, where # is an integer indicating the FORTRAN unit number. The MC-TINKERATE interfaces two computer programs, MC-TINKER and POLYRATE, so the input and output files used in a typical MC-TINKERATE calculation can be divided, based on the part of the code that opens and closes them, in three categories: (i) POLYRATE files, (ii) MC-TINKER files, and (iii) interface files that are read in by the MC-TINKERATE. These files are briefly described in the following sections. For more detailed descriptions, the user is referred to the POLYRATE manual (for the POLYRATE files) or the MC-TINKER manual (for the MC-TINKER files).

6.1. Description of `esp.fu70` input file

The `esp.fu70` input file is an interface file that contains species-dependent input data other that the ones read in the `poly.fu5` POLYRATE input file. The input for FORTRAN unit `esp.fu70` is in the free format keyword style. The `esp.fu70` input consists of up to seven sections. The input file consists of section headers (section headers are always preceded by an asterisk, *) and associated keywords. Please note that every keyword must be used in its proper section, i.e., between the occurrence of its header and the occurrence of the next header (or, for the last section, between the last header and the end of the file).

The MRREACT1, MRREACT2, MRPROD1, MRPROD2, MRWELLR, MRWELLP, and MRSTART sections in the `esp.fu70` input file contain input data that describe the appropriate reaction species as do the REACT1, REACT2, PROD1, PROD2, WELLR, WELLP, and START sections in the `poly.fu5` input file, i.e., reactant 1, reactant 2, product 1, product 2, reactant well, product well, and the saddle (or starting) point.

In the current version of the program, only variable keywords are used in the `esp.fu70` input file. A variable keyword always requires an argument. The argument must appear on the same line as the variable keyword. *All* variable keywords have default values, and these types of keywords need to be given only if the user wishes to override the default. Accordingly, the `esp.fu70` input file should be present only if the user wishes to override the default values of any of the keywords.

Example:

```
*MRREACT1

MRUNITINPUT bohr

*MRWELLP

MRUNITINPUT bohr

*MRSTART

MRUNITINPUT ang
```

6.1.1. The MRREACT1, MRREACT2, MRPROD1, MRPROD2, MRWELLR, MRWELLP, and MRSTART sections

The MRREACT1, MRREACT2, MRPROD1, MRPROD2, MRWELLR, MRWELLP, and MRSTART sections are used to give additional information to the reaction species that are not given in the `poly.fu5` input file. In the current version of the program, each of these sections can have only one keyword, which is explained below.

MRUNITINPUT

MRUNITINPUT is a variable keyword that specifies the units for the geometries given in the appropriate `esp.fu71` to `esp.fu78` input file. The default is *`ang`*.

Option	Description
`bohr`	coordinates are in bohrs
`ang`	coordinates are in angstroms

Example:

```
MRUNITINPUT   bohr
```

6.2. Description of `esp.fu71` to `esp.fu78` input files

The `esp.fu71` to `esp.fu78` input files are interface files that include the geometry information for the reaction species. All the records are read in free format. These files use a format similar to the one used in the `esp.fu84` MC-TINKER input file.

The table below synthesizes the reaction species for which the `esp.fu71` to `esp.fu78` input files are used, and the sections in the `esp.fu70` MC-TINKER input file and `poly.fu5` POLYRATE input file correlated with these species.

Reaction species	Input file	Section in `poly.fu5`	Section in `esp.fu70`
Reactant 1	`esp.fu71`	REACT1	MRREACT1
Reactant 2	`esp.fu72`	REACT2	MRREACT2
Product 1	`esp.fu73`	PROD1	MRPROD1
Product 2	`esp.fu74`	PROD2	MRPROD2
Saddle point	`esp.fu75`	START	MRSTART
Reactant well	`esp.fu77`	WELLR	MRWELLR
Product well	`esp.fu78`	WELLP	MRWELLP

The `esp.fu71` to `esp.fu78` input files are read only if the INITGEO keyword in the `poly.fu5` POLYRATE input file is set to *`hooks`*. The only consideration to be taken into account is the restriction on the ordering of atoms, that is the order used in these files should be the same as the order used for the GEOM keyword in the appropriate section of the `poly.fu5` POLYRATE input file.

The first two lines of the `esp.fu71` to `esp.fu78` input files should contain:

First line: The keyword POINTS followed by *`mcnpts`*, which is an integer indicating the number of geometries given in the file. In the current version of the code this number should be *`1`*.

Second line: The keyword FORMAT followed by *`mcifrm`*, which is an integer indicating the format used to read the data in the file.

Example:

```
POINTS        1
FORMAT        2
```

At this time, the remaining information of the file is read depending of the values of *mcifrm*.

For *mcifrm* = 1, the file contains the following records:

Record 1: An integer indicating the number of atoms in the species.

Record 2: Comment line.

Record 3: A number of lines equal to the number of atoms in the reaction species containing the index of the atom in the molecule (or the chemical symbol) and the Cartesian coordinates for that atom.

Example:

```
points 1
format 1
   7
Comment: First geometry of saddle point
1   0.000000  0.000000  0.000000
2   1.105391  0.000000  0.000000
3   1.452921  1.160817  0.000000
4   1.536666 -0.459271 -0.908287
5   1.540129 -0.398133  0.937112
6   1.386584  2.344293 -0.681525
7   0.850429  2.088654 -1.487727
```

For *mcifrm* = 2, the file contains the following records:

Record 1: Comment line.

Record 2: A number of lines equal to the number of atoms in the reaction species containing the index of the atom in the molecule (or the chemical symbol) and the Cartesian coordinates for that atom.

Example:

```
Points 1
Format 2
Comment: First geometry of reactant 1
1   0.000000  0.000000  0.000000
5   1.540129 -0.398133  0.937112
6   1.386584  2.344293 -0.681525
7   0.850429  2.088654 -1.487727
```

For *mcifrm* = 3, the file contains the following records:

Record 1: Comment line.

Record 2: A number of lines equal to the number of atoms in the reaction species; these lines contain the Cartesian coordinates for the atoms.

Example:

```
POINTS 1
FORMAT 3
Comment: First geometry of reactant 2
 1.105391  0.000000  0.000000
 1.452921  1.160817  0.000000
 1.536666 -0.459271 -0.908287
```

Note that, independent of the format used, everything that is written on a line following the third Cartesian coordinate is ignored.

The formats used for geometries are the same as the formats used by TINKER (*mcifrm* = 1), by the fu27 file of POLYRATE (*mcifrm* = 2), and by the fu5 file of POLYRATE (*mcifrm* = 3).

6.3. Short description of `esp.fu81` and `esp.fu82` input files

The `esp.fu81` and `esp.fu82` input files are MC-TINKER input files that contain the geometries of the single-configuration molecular mechanics Shepard points (the SCMM point for configuration 1 and SCMM point for configuration 2, respectively), the atom types, and the connectivities for the configurations 1 and 2, respectively. These input files are written in the same fashion as any Cartesian coordinate input file for TINKER. The only consideration to be taken into account is the restriction on the ordering of atoms, that is, the order used in these files should be the same as the order used for the MCATOMS keyword in MCGENERAL section of the general MC-TINKER input file.

Example:

```
7  HOH---CH3 well
1  H   -2.528735  0.393818 -0.630578   5  2
2  C*  -2.225690 -0.619356 -0.324313  29  1  4  5
3  H     .457798  1.026333  0.821507  21  6
4  H   -1.822563 -1.322295 -1.069648   5  2
5  H   -2.320781 -0.927796  0.728296   5  2
6  O    1.068112  0.590172  0.243503   6  3  7
7  H    1.013893 -0.326197  0.476171  21  6
```

6.4. Short description of `esp.fu83` input file

The `esp.fu83` input file is a MC-TINKER input file that contains the input data (geometry, energy, gradient, Hessian) for the electronic structure theory Shepard points used in Shepard interpolation. The input for FORTRAN unit `esp.fu83` is in the free format keyword style. The input file consists of section headers (section headers are always preceded by an asterisk, *) and associated keywords. The `esp.fu83` input file consists of the MCGEN83 section, which gives general information about the file, and a number of POINT sections, which give the electronic structure data for an electronic structure theory Shepard point. Note that the sections can be entered in any order in the input file. The keywords used in this file are of two types: variable and list.

Example:

```
*MCGEN83
NESTSP 2

*POINT 1
ENERGY 0.0189494
GEOMETRY
 -6.69454329E-02 -1.96574211E+00  0.00000000E+00 -9.15479645E-01  8.85833624E-01
  0.00000000E+00 -1.03439821E-01  3.77243731E+00  0.00000000E+00
END
GRADIENT
 -2.80934625E-08 -7.33177670E-09  2.12772599E-11 -1.41410517E-08  3.67943859E-08
  1.46347707E-11  4.22345150E-08 -2.94626079E-08 -3.59120307E-11
END
HESSIAN
  7.26849015E-03 -2.46048400E-02  9.35414373E-02 -4.14090454E-12  5.92075782E-12
 -1.80332522E-08 -1.58285758E-02  5.21279047E-02 -6.41965167E-12  2.30461818E-02
 -6.01027280E-03  1.56393041E-02 -4.61939022E-11 -2.75845692E-02 -1.39825157E-01
  1.52073611E-11 -3.72014964E-11 -2.99670799E-08  1.54332795E-11  1.38812605E-10
  8.11309802E-08  8.56008545E-03 -2.75230647E-02  1.05605569E-11 -7.21760599E-03
  3.35948420E-02 -3.06406234E-11 -1.34247944E-03  3.06151128E-02 -1.09180741E-01
  4.02730797E-11 -2.45433355E-02  1.24185853E-01 -1.01611061E-10 -6.07177726E-03
 -1.50051120E-02 -1.10664188E-11  3.12807014E-11  4.74362771E-08 -9.01366720E-12
 -9.26186406E-11 -5.11638840E-08  2.00800682E-11  6.13379559E-11  3.77438791E-09
END

*POINT 2
ENERGY 0.0155076
GEOMETRY
  1.88783626E-02 -1.77342847E+00  0.00000000E+00 -9.64608741E-01  7.58182633E-01
  0.00000000E+00 -1.48060031E-02  3.98045268E+00  0.00000000E+00
END
GRADIENT
 -7.04527640E-03  2.08625864E-02  4.20704162E-11  1.23914913E-02  9.85972298E-05
 -1.53427741E-11 -5.34621492E-03 -2.09611837E-02 -2.67276420E-11
END
HESSIAN
  2.73425524E-02 -8.91295912E-02  2.38375655E-01 -1.64800680E-11  3.20686826E-11
 -7.62842906E-03 -3.22699724E-02  9.74643152E-02  1.37702563E-11  3.31359082E-02
  7.49672027E-02 -1.93418930E-01 -3.23699396E-11 -7.95148555E-02  1.64483703E-01
  2.15023069E-11 -2.75062061E-11  7.14722292E-03 -1.28980457E-11  6.75315374E-11
 -1.27930503E-02  4.92742148E-03 -8.33472403E-03  2.70970039E-12 -8.65935818E-04
  4.54765281E-03 -8.60421477E-12 -4.06148564E-03  1.41623885E-02 -4.49567246E-02
  3.01891512E-13 -1.79494597E-02  2.89352272E-02 -4.00256698E-11  3.78707122E-03
  1.60214974E-02 -5.02228204E-12 -4.56263118E-12  4.81206933E-04 -8.72237563E-13
 -3.51617696E-11  5.64582739E-03  5.89458449E-12  3.97241047E-11 -6.12703432E-03
END
```

6.5. Short description of `esp.fu85` input file

The `esp.fu85` input file is a MC-TINKER input file that contains the general input data that controls the MCMM/SCMM part of the calculation. The input for FORTRAN unit `esp.fu85` is in the free format keyword style. The `esp.fu85` input is divided into sections, and it consists of section headers (section headers are always preceded by an asterisk, *) and associated keywords. The keywords are of two types: variable and list. *All* variable keywords have default values, and these types of keywords need to be given only if the user wishes to override the default. Please note that every keyword must be used in its proper section, i.e., between the occurrence of its header and the occurrence of the next header (or, for the last section, between the proper header and the end of the file). Except for the input of the MCTITLE keyword in the MCGENERAL section, all input is case insensitive, and anything after a pound sign (#) on a line is assumed to be a comment (except within the ICSHEPARD and ICDISTANCE list keywords). Blank lines are ignored.

Example:

```
*MCGENERAL
MCTITLE
CH4 + OH -> CH3 + H2O, AM1/SRP calculation
END
MCATOMS
 1 H
 2 C
 3 H
 4 H
 5 H
 6 O
 7 H
END
ORDERCALC mcmm

*MCENERGETICS
EDIFF -0.0211635

*RESONANCE
ICSHEPARD
  1-2 2-4 2-5 2-3 3-6 6-7 2-6
  3-2-1 3-2-4 3-2-5 1-2-4 1-2-5 2-3-6 3-6-7 4-2-5
  2-3-6-7 5-2-3-6 4-2-3-6 1-2-3-6
END
ICDISTANCE
  2-3 2-6 3-6
END
```

6.5.1. Keyword usage in MC-TINKERATE versus MC-TINKER

Certain keywords of MC-TINKER that are used in the `esp.fu85` input file are not used or have constraints when the `esp.fu85` file is used in a MC-TINKERATE calculation. The usage of each keyword in a MC-TINKER calculation is presented in detail in the MC-TINKER manual and will not be repeated here; instead we will present only the keywords that are not used or are used with constraints in MC-TINKERATE. Each such keyword is presented below in alphabetical order.

MCATOMS

In MC-TINKER, MCATOMS is a list keyword that is used to specify all the atoms in the system. In MC-TINKERATE, the order of atoms in the MCATOMS keyword should be the same as the order of atoms in the ATOMS keyword in the `poly.fu5` POLYRATE input file.

ORDERCALC

In MC-TINKER, ORDERCALC is a variable keyword that specifies whether an SCMM or an MCMM calculation is to be carried out. (Recall that SCMM is a synonym for MM.) The default is *mcmm*. In MC-TINKERATE, the keyword is used to specify if the dynamics calculation should be carried out on an SCMM or an MCMM potential energy surface.

SCMASS

In MC-TINKER, SCMASS is a variable keyword that specifies the value (in amu) for the parameter μ used in defining the mass-scaled coordinates in input and output geometries. In the current version of MC-TINKERATE, this keyword is not used. The value of μ used for rate calculations is defined in the SCALEMASS keyword in the `poly.fu5` POLYRATE input file.

TYPECALC

In MC-TINKER, TYPECALC is a variable keyword that specifies whether an energy, gradient, or Hessian is to be calculated. POLYRATE hooks make all these types of calls to MC-TINKER. In the current version of MC-TINKERATE, this keyword is not used.

UNITINPUT

In MC-TINKER, UNITINPUT is a variable keyword that specifies the units for the geometries given in the `esp.fu84` input file. In the current version of MC-TINKERATE, this keyword is not used, because the units for the geometries read in the `poly.fu5` input file are given in the `poly.fu5` input file, and the units for the geometries read in the `esp.fu71` to `esp.fu78` input files are given in the `esp.fu70` input file

UNITOUTPUT

In MC-TINKER, UNITOUTPUT is a variable keyword that specifies the units for the results printed in the `esp.fu86` output file. In the current version of MC-TINKERATE, this keyword is not used, because the POLYRATE side of the interface controls the output options.

EECALC (or other keywords concerned with an EE-MM or EE-MCMM calculation)

In MC-TINKER, EECALC is a variable keyword that specifies whether an EE-MM or EE-MCMM is to be calculated. To carry out an EE-MM or EE-MCMM calculation, the electrostatic potential distribution from the external region is required, and it cannot be calculated by MC-TINKERATE itself. In the current version of MC-TINKERATE, this keyword is not used; an EE-MM or EE-MCMM calculation is unavailable.

6.6. Short description of `poly.fu5` input file

The `poly.fu5` input file is a POLYRATE input file that contains the general input data that controls the dynamics part of the calculation. The `poly.fu5` input file is divided into sections, and it consists of section headers (section headers are always preceded by as asterisk, *) and associated keywords. The keywords are of three types: variable, switch, and list. *All* variable and switch, and some list keywords have default values, and these types of keywords need to be given only if the user wishes to override the default. Please note that every keyword must be used in its proper section, i.e., between the occurrence of its header and the occurrence of the next header (or, for the last section, between the proper header and the end of the file). Except for the input of the TITLE keyword in the GENERAL section, all input is case insensitive, and anything after a pound sign (#) on a line is assumed to be a comment (except within the INTDEF list keyword). Blank lines are ignored.

6.6.1. Keyword usage in MC-TINKERATE versus POLYRATE

Certain keywords used in the `poly.fu5` input file are used with various constraints when the `poly.fu5` file is used in a MC-TINKERATE calculation. The usage of each keyword of POLYRATE is presented in detail in the POLYRATE manual and will not be repeated here; instead we will present only the keywords that have constraints in MC-TINKERATE. Each such keyword is presented below in alphabetical order.

ATOMS

In POLYRATE, ATOMS is a list keyword that is used to specify all the atoms in the system. In MC-TINKERATE, the order of atoms in the ATOMS keyword should be the same as the order of atoms in the MCATOMS keyword in the `esp.fu85` MC-TINKER input file.

OPTMIN

In POLYRATE, OPTMIN is a variable keyword that determines how geometries of minima are optimized. The default is *bfgs*. In MC-TINKERATE, the OPTMIN keyword should *not* be set to *ohook*; this option is not available in the current version of MC-TINKERATE.

OPTTS

In POLYRATE, OPTTS is a variable keyword that determines how geometries of the transition state is optimized. The default is *nr*. In MC-TINKERATE, the OPTTS keyword should *not* be set to *ohook*; this option is not available in the current version of MC-TINKERATE.

POTENTIAL

In POLYRATE, POTENTIAL is a variable keyword that allows the user to specify the source of the potential energy information for a single-level calculation or for the lower level of a dual-level calculation. The default is *hooks*. In the current version of MC-TINKERATE, the POTENTIAL keyword should be set to the default, *hooks*.

STATUS

In POLYRATE, STATUS is a variable keyword that is used to indicate the status of the stationary points (reactant 1, reactant 2, product 1, product 2, reactant well, product well, and starting point). The default is *0*. In the current version of MC-TINKERATE, due to the fact that calculations are allowed only for species that contains all atoms of the system, the STATUS keyword should be set to *6* for all the species that do not contain all atoms that is reactant 1 and reactant 2 for a bimolecular reaction or product 1 and product 2 for a reaction with two products). Note that if STATUS is *6* then ENERGY, FREQ, and EIGENVECTOR (only for the starting point) list keywords must be given. In addition, the GEOM keyword is always required.

SUPERMOL (NOSUPERMOL)

In POLYRATE, SUPERMOL is a switch keyword that indicates that the supermolecule mode is used for reactants and products. In the current version of MC-TINKERATE, due to the fact that calculations are allowed only for species that contains all atoms of the system, this keyword is not used. Current version of MC-TINKERATE only works in SUPERMOL mode, even if the user specifies NOSUPERMOL.

7. SAMPLE TEST RUNS

The test suite includes seventeen test runs. Each of these test runs is described below. To run the complete test suite, one can use the `run_all` or `run_all_FC` scripts located in the `testmcr` directory. Script `run_all` does tests 1~17 while script `run_all_FC` does tests 1~18. Alternatively, the test runs can be run one at a time by running each of the `run_testr#` scripts located in the appropriate subdirectory in the `testmcr` directory.

Two scripts `getnew` and `compnew` are provided for users' convenience. Running script `getnew` will collect the most recent outputs for all test calculations into directory `testonew`, and subsequently running script `compnew` will generate files for comparisons between these newly obtained and those distributed outputs for all test runs

7.0. A note on the mPW and omPW functionals

Several of the test runs use the MPW1K [31] density functional. This is a combination of 42.8% Hartree-Fock exchange, 57.2% mPW exchange, and 100% PW91 exchange, where mPW denotes the modified Perdew-Wang'91 exchange functional developed by Adamo and Barone [32], and PW91 denotes the Perdew-Wang'91 [33] correlation functional. Note that when the mPW exchange functional is used as part of a density functional developed at Minnesota, the name is changed to MPW with a capital M, which still denotes "modified".

Versions of *Gaussian 98* through g98.a11 have an error in the mPW exchange functional; this error was first pointed out in an appendix to a paper [29] using this functional. The bug causes small errors in all energies calculated using hybrid DFT methods that involve the mPW or MPW exchange functional, including MPW1K. The typical error in bond energies introduced by the bug is about 0.1 kcal/mol.

The bug exists in all versions of *Gaussian 98* through revision a.11. The bug was fixed in a local version of *Gaussian 98* revision a.11, and it has been fixed in all released versions of *Gaussian 03*. Ordinarily when a bug is fixed, one would simply switch to using the corrected functional. However, since a certain number of results were published with the incorrect mPW functional, Gaussian03 has a keyword, namely omPW, for running the old incorrect version of the mPW exchange functional with the bug.

Our third paper employing MCMM [3] was run with the bug in Gaussian. To reproduce those results, one must use the incorrect mPW functional by invoking the omPW keyword. The MPW1K functional with omPW instead of mPW will be called oMPW1K in this manual. to facilitate comparison will past results, test runs 11, 12, and 14–17 were run with the oMPW1K density functional.

7.1. Test run 1

Internal rotation of C_2H_6 by SCMM

This is an SCMM calculation for the internal rotation isomerization rate constant of ethane. Although this is a somewhat artificial example (because it is a degenerate rearrangement and because ordinary pressures are too low to stabilize the system in one or another well), it does illustrate how to use the MC-TINKERATE code in SCMM mode. The values for the STATUS keyword for REACT1, WELLR, SADDLE, WELLP, and PROD1 are all *0* (default). The structures for WELLR and WELLP are the same as those of REACT1 and PROD1, respectively. The minimum energy path, MEP, is calculated by the Euler-Stabilization (*es1*) method. The structures, gradients, Hessians, and energies are calculated with the MM3 force field. The rate constants are computed using TST. No tunneling calculation has been performed.

INPUT FILES:

testr1.81	Configuration 1 input file
testr1.85	General MC-TINKER input file
testr1.dat	General POLYRATE input file
param-testr1.prm	Modified TINKER parameter file
run_testr1	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

testr1.fu15	Summary output file

OUTPUT FILES (not distributed):

testr1.fu6	Full output file
testr1.fu25	Potential energies along the MEP
testr1.fu26	Generalized TS frequencies along the MEP
testr1.fu27	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
testr1.fu61	Stationary point information
testr1.time	Computing time

7.2. Test run 2

The abstraction of hydrogen from CH_4 by OH by MCMM(1)

This run is an MCMM calculation of the OH + CH_4 hydrogen abstraction reaction. It is a bimolecular reaction, and the interpolation is carried out between reactant and product wells. Some parameters in the parameter file `param-testr2.prm` are added in order to describe an atom type (O in OH radical) undefined in the original MM3 force field. The electronic structure data for the reference points are taken from AM1-SRP calculations. The SRP parameters [29] for the O atom are U_{ss} = –119.70 eV, and U_{pp} = –79.97 eV. This test run is based on three Shepard points (one electronic structure theory point for the saddle point, and the two molecular mechanics wells computed using the MM3 force field with additional parameters for the atom types that do not exist in the original MM3 force filed). The number in parenthesis in the MCMM(x) notation indicates the total number of electronic structure theory Shepard points used in the interpolation.

The STATUS keyword is set to *6* for REACT1, REACT2, PROD1, and PROD2, and the structures and frequencies obtained by electronic structure calculations are used as input for the POLYRATE input file. These data are used to calculate the partition functions for the reactant and product. The energy of reaction is also obtained by electronic structure calculations, and this value is used for EDIFF in the MC-TINKER input file and for the product energy in the POLYRATE input file. The STATUS keyword is set to *0* for the wells and the transition state.

The reaction path is scaled to a mass of *1* amu. The MEP is calculated by the Page-McIver method with a step size of *0.001* bohrs. The Hessian matrix is calculated and a generalized normal mode analysis in curvilinear internal coordinates is carried out every *0.01* bohrs. The rate constants are computed using TST and CVT. Tunneling contributions are computed using the ZCT, SCT, and LCG4 methods.

This calculation should reproduce the TST, CVT, and CVT/SCT results in Table V of Ref. [1]. Note that Ref. [1] used the obsolete LCG3 method for LCT and μOMT calculations; the current version of POLYRATE uses LCG4.

INPUT FILES:

`testr2.81`	Configuration 1 input file
`testr2.82`	Configuration 2 input file
`testr2.83`	Electronic structure theory data file
`testr2.85`	General MC-TINKER input file
`testr2.dat`	General POLYRATE input file
`param-testr2.prm`	Modified TINKER parameter file
`run_testr2`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr2.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr2.fu6`	Full output file
`testr2.fu25`	Potential energies along the MEP
`testr2.fu26`	Generalized TS frequencies along the MEP
`testr2.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr2.fu61`	Stationary point information
`testr2.time`	Computing time

7.3. Test run 3

The abstraction of hydrogen from CH_4 by OH by MCMM(3)

This test run is the same as test run 2 except that it uses five Shepard points; these consist of three electronic structure theory points (calculated by AM1-SRP, that is Austin Model 1 with specific reaction parameters), and the reactant-valley and product-valley wells, which are calculated by the molecular mechanics force field. The three electronic structure Shepard points are the transition state, a point approximately halfway down in energy from the top of barrier toward the reactant, and another point the same amount down in energy toward the product.

INPUT FILES:

`testr3.81`	Configuration 1 input file
`testr3.82`	Configuration 2 input file
`testr3.83`	Electronic structure theory data file
`testr3.85`	General MC-TINKER input file
`testr3.dat`	General POLYRATE input file
`param-testr3.prm`	Modified TINKER parameter file
`run_testr3`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr3.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr3.fu6`	Full output file
`testr3.fu25`	Potential energies along the MEP
`testr3.fu26`	Generalized TS frequencies along the MEP
`testr3.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr3.fu61`	Stationary point information
`testr3.time`	Computing time

7.4. Test run 4

The abstraction of hydrogen from CH_4 by OH by MCMM(5)

This test run is the same as test run 2 except that it uses seven Shepard points; these consist of five AM1-SRP electronic structure theory points, which are selected by an iterative process (see Ref. [1]), and the reactant-valley and product-valley wells, which are calculated by the molecular mechanics force field. The five electronic structure points are the transition state, a point a halfway down in energy from the top of barrier toward the reactant and another point the same amount down in energy toward the product, a point one-quarter way down from the top of barrier toward the reactant and another point the same amount down toward the product.

INPUT FILES:

`testr4.81`	Configuration 1 input file
`testr4.82`	Configuration 2 input file
`testr4.83`	Electronic structure theory data file
`testr4.85`	General MC-TINKER input file
`testr4.dat`	General POLYRATE input file
`param-testr4.prm`	Modified TINKER parameter file
`run_testr4`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr4.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr4.fu6`	Full output file
`testr4.fu25`	Potential energies along the MEP
`testr4.fu26`	Generalized TS frequencies along the MEP
`testr4.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr4.fu61`	Stationary point information
`testr4.time`	Computing time

7.5. Test run 5

The 1,5-sigmatropic hydrogen shift rearrangement of 1,3-*cis*-pentadiene by MCMM(5)

This is an MCMM calculation for the 1,5-sigmatropic rearrangement of 1,3-*cis*-pentadiene. This MCMM calculation is based on seven Shepard points; these consist of five AM1 electronic structure theory points, which are selected by an iterative process (see Ref. [1]), and the reactant-valley and product-valley wells, which are calculated by the molecular mechanics force field. The five electronic structure points are the transition state, a point a halfway down in energy from the top of barrier toward the reactant and another point the same amount down toward the product, a point one-quarter way down from the top of barrier toward the reactant and another point the same amount down toward the product. This calculation has one reactant (1,3-*cis*-pentadiene) and one product (1,3-*cis*-pentadiene). The structures for the reactant and product have the s-*trans* conformation, and those of reactant-valley and product-valley wells have the s-*cis* conformation. The difference in energy between s-*trans* and s-*cis* conformers is about 3.2 kcal/mol in the MM3 force field.

The ZERO1 and ZERO2 values in the MC-TINKER input file are computed by using TINKER with a VESCF calculation. The structure of the reactant well is also optimized with a VESCF calculation. The parameter file `param-testr5.prm` is a modified version of the standard MM3 force field parameters and includes the VESCF-optimized parameters. A new atom type, the atom type number 20, is defined to use the VESCF-optimized parameters and to avoid the VESCF calculations (see MC-TINKER manual). No further special treatment for the π system is made. The structure, energy, and frequencies for the reactant and product wells are exactly the same as those using the VESCF method with the original set of parameters. The electronic structure data is taken from AM1 calculations.

The STATUS keyword is set to *6* for REACT1 and PROD1, and the structures and frequencies obtained by electronic structure calculations are used as input for the POLYRATE input file. These data are used to calculate the partition functions for the reactant and product. Since this reaction is symmetric, the energy of reaction is zero. The STATUS keyword is set to *0* for the wells and the transition state.

The reaction path is scaled to a mass of *1* amu. The MEP is calculated by the Page-McIver method with a step size of *0.001* bohrs. The Hessian matrix is calculated and a generalized normal mode analysis in curvilinear internal coordinates is carried out every *0.01* bohrs. The rate constants are computed using TST and CVT. Tunneling contributions are computed using the ZCT and SCT methods.

INPUT FILES:

`testr5.81`	Configuration 1 input file
`testr5.82`	Configuration 2 input file
`testr5.83`	Electronic structure theory data file
`testr5.85`	General MC-TINKER input file
`testr5.dat`	General POLYRATE input file
`param-testr5.prm`	Modified TINKER parameter file
`run_testr5`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr5.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr5.fu6`	Full output file
`testr5.fu25`	Potential energies along the MEP
`testr5.fu26`	Generalized TS frequencies along the MEP
`testr5.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr5.fu61`	Stationary point information
`testr5.time`	Computing time

7.6. Test run 6

The 1,5-sigmatropic hydrogen shift rearrangement of 1,3-*cis*-pentadiene by MCMM(3)

This test run is the same as the test run 5 except that it uses the Morse bond type for all bonds in the molecule, and it uses only five Shepard points; these consist of three AM1 electronic structure theory points, which are selected by an iterative process (see Ref. [1]), and the reactant-valley and product-valley wells, which are calculated by the molecular mechanics force field. The three high-level points are the transition state, a point a halfway down in energy from the top of barrier toward the reactant and another point the same amount down in energy toward the product.

The parameter file `param-testr6.prm` is the same as the `param-testr5.prm` in the test run 5 except that it uses the BONDTYPE MORSE keyword.

INPUT FILES:

`testr6.81`	Configuration 1 input file
`testr6.82`	Configuration 2 input file
`testr6.83`	Electronic structure theory data file
`testr6.85`	General MC-TINKER input file
`testr6.dat`	General POLYRATE input file
`param-testr6.prm`	Modified TINKER parameter file
`run_testr6`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr6.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr6.fu6`	Full output file
`testr6.fu25`	Potential energies along the MEP
`testr6.fu26`	Generalized TS frequencies along the MEP
`testr6.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr6.fu61`	Stationary point information
`testr6.time`	Computing time

7.7. Test run 7

The 1,5-sigmatropic hydrogen shift rearrangement of 1,3-*cis*-pentadiene by MCMM(3)

This test run is the same as the test run 6 except that it uses the Morse bond type for C(sp^3)-H bonds and the Taylor bond type for all other bonds in the molecule, and it reads the initial geometries for the wells and the saddle point from interface input files.. The parameter file `param-testr7.prm` is the same as the `param-testr5.prm` in the test run 5 except that it uses 99 kcal/mol for the bond strength of the C(sp^3)-H bonds (see the MC-TINKER manual).

INPUT FILES:

`testr7.70`	General interface input file
`testr7.75`	Saddle point geometry input file
`testr7.77`	Reactant well geometry input file
`testr7.78`	Product well geometry input file
`testr7.81`	Configuration 1 input file
`testr7.82`	Configuration 2 input file
`testr7.83`	Electronic structure theory data file
`testr7.85`	General MC-TINKER input file
`testr7.dat`	General POLYRATE input file
`param-testr7.prm`	Modified TINKER parameter file
`run_testr7`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr7.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr7.fu6`	Full output file
`testr7.fu25`	Potential energies along the MEP
`testr7.fu26`	Generalized TS frequencies along the MEP
`testr7.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr7.fu61`	Stationary point information
`testr7.time`	Computing time

7.8. Test run 8

The 1,5-sigmatropic hydrogen shift rearrangement of 1,3-*cis*-pentadiene by MCMM(3)

This test run is the same as the test run 6 except that it uses the Taylor bond type for all bonds in the molecule, it interpolates directly to the reactant and the product without the reactant and product well structures, and it uses. The structures of the reactant and product have the s-*cis* conformation. The MM energy of the s-*cis* conformation is 8.92 kcal/mol, so the ZERO1 and ZERO2 values in the MC-TINKER input file are also changed. The parameter file `param-testr8.prm` is the same as the `param-testr5.prm` in the test run 5.

INPUT FILES:

`testr8.81`	Configuration 1 input file
`testr8.82`	Configuration 2 input file
`testr8.83`	Electronic structure theory data file
`testr8.85`	General MC-TINKER input file
`testr8.dat`	General POLYRATE input file
`param-testr8.prm`	Modified TINKER parameter file
`run_testr8`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr8.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr8.fu6`	Full output file
`testr8.fu25`	Potential energies along the MEP
`testr8.fu26`	Generalized TS frequencies along the MEP
`testr8.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr8.fu61`	Stationary point information
`testr8.time`	Computing time

7.9. Test run 9

The abstraction of hydrogen from CH_3F by CH_2Cl by MCMM(8)

This test run is an MCMM calculation of the CH_2Cl + CH_3F hydrogen abstraction reaction. It is a bimolecular reaction, and Shepard interpolation is carried out from the reactant well to the product well. Some parameters are added to the `param-testr9.prm` parameter file in order to describe some atom types that are undefined in the original MM3 force field. The electronic structure calculations for this reaction are carried out using PM3 method. This test run is based on ten Shepard points; these consist of eight electronic structure theory points, which are selected by an iterative process (see Ref. [1]), and the reactant-valley and product-valley wells, which are calculated by the MM3 force field with additional parameters for the atom types that do not exist in the original MM3 force field. The eight electronic structure points are the saddle point, a point a halfway down in energy from the top of barrier toward the reactant, point the same amount down in energy toward the product, a point one-quarter way down from the top of barrier toward the reactant, another point the same amount down toward the product, the reactant-valley and the product-valley turning points at the representative tunneling energy, and the critical configuration.

The STATUS keyword is set to *6* for REACT1, REACT2, PROD1, and PROD2, and the structures and frequencies obtained by electronic structure calculations are used as input for the POLYRATE input file. These data are used to calculate the partition functions for the reactant and product. The energy of reaction is also obtained by electronic structure calculations, and this value is used for EDIFF in the MC-TINKER input file and for the product energy in the POLYRATE input file. The STATUS keyword is set to *0* for the wells and the transition state.

The reaction path is scaled to a mass of *1* amu. The MEP is calculated by the Page-McIver method with a step size of *0.001* bohrs. The Hessian matrix is calculated and a generalized normal mode analysis in curvilinear internal coordinates is carried out every *0.01* bohrs. The rate constants are computed using TST and CVT. Tunneling contributions are computed using the ZCT, SCT, and LCG3 methods.

INPUT FILES:

`testr9.81`	Configuration 1 input file
`testr9.82`	Configuration 2 input file
`testr9.83`	Electronic structure theory data file
`testr9.85`	General MC-TINKER input file
`testr9.dat`	General POLYRATE input file
`param-testr9.prm`	Modified TINKER parameter file
`run_testr9`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr9.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr9.fu6`	Full output file
`testr9.fu25`	Potential energies along the MEP
`testr9.fu26`	Generalized TS frequencies along the MEP
`testr9.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr9.fu61`	Stationary point information
`testr9.time`	Computing time

7.10. Test run 10

The abstraction of hydrogen from CH_3F by CH_2Cl by MCMM(8)

This test run is the same as the test run 9 except that it calculates the large-tunneling contributions using the LCG4 method instead of the LCG3 methods. The parameter file `param-testr10.prm` is the same as the `param-testr9.prm` in the test run 9.

INPUT FILES:

`testr10.81`	Configuration 1 input file
`testr10.82`	Configuration 2 input file
`testr10.83`	Electronic structure theory data file
`testr10.85`	General MC-TINKER input file
`testr10.dat`	General POLYRATE input file
`param-testr10.prm`	Modified TINKER parameter file
`run_testr10`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr10.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr10.fu6`	Full output file
`testr10.fu25`	Potential energies along the MEP
`testr10.fu26`	Generalized TS frequencies along the MEP
`testr10.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr10.fu61`	Stationary point information
`testr10.time`	Computing time

7.11. Test run 11

The abstraction of hydrogen from CH_4 by O by MCMM(8)

This run is an MCMM calculation of the O + CH_4 hydrogen abstraction reaction. Because it is a bimolecular reaction, the interpolation is carried out between the reactant and product wells. Some parameters in the parameter file `param-testr11.prm` are added in order to describe two atom types (O in OH radical and O atom) that are undefined in the original MM3 force field. The electronic structure theory data is obtained at the oMPW1K/6-31+G(d,p) level of theory. This test run is based on ten Shepard points: eight electronic structure theory points, selected as described next, and the reactant-valley and product-valley wells, which are calculated with the MM3 force field with the additional parameters for the atom types that do not exist in the original MM3 force field. The eight electronic structure points used are the saddle point, a point one-quarter way down from the top of the barrier toward the products CH_3 + OH (which would be the reactants if the reaction were considered in the exoergic direction) and a point at the same energy toward the reactants CH_4 + O (which would be the products if the reaction were considered in the exoergic direction) as determined in the MCMM–0 run, a point halfway down from the top of the barrier toward the products and a point at the same energy toward the reactants as determined in the MCMM–2 run, the reactant-valley and the product-valley turning points at the LCG4 representative tunneling energy at 500 K as determined in the MCMM–4 run, and another point that is the Cartesian average of these last two points. Two internal coordinates are used for calculating generalized distances between points in Shepard interpolation step, and these are the O–H and the C–H distances where H is the transferring hydrogen atom.

The STATUS keyword is set to *6* for REACT1, REACT2, PROD1, and PROD2, and the structures and frequencies obtained by electronic structure calculations are used as input for the POLYRATE input file. These data are used to calculate the partition functions for the reactant and product. The energy of reaction is also obtained by electronic structure calculations, and this value is used for EDIFF in the MC-TINKER input file and for the product energy in the POLYRATE input file. The STATUS keyword is set to *0* for the wells and the transition state.

The reaction path is scaled to a mass of *1* amu. The MEP is calculated by the Page-McIver method with a step size of *0.001* bohrs. The Hessian matrix is calculated and a generalized normal mode analysis in curvilinear internal coordinates is carried out every *0.01* bohrs. The rate constants are computed using TST and CVT. Tunneling contributions are computed using the ZCT, SCT, and LCG4 methods.

INPUT FILES:

`testr11.81`	Configuration 1 input file
`testr11.82`	Configuration 2 input file
`testr11.83`	Electronic structure theory data file
`testr11.85`	General MC-TINKER input file
`testr11.dat`	General POLYRATE input file
`param-testr11.prm`	Modified TINKER parameter file
`run_testr11`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr11.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr11.fu6`	Full output file
`testr11.fu25`	Potential energies along the MEP
`testr11.fu26`	Generalized TS frequencies along the MEP
`testr11.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr11.fu61`	Stationary point information
`testr11.time`	Computing time

7.12. Test run 12

The abstraction of hydrogen from CH_4 by O by MCMM(8)

This test run is the same as test run 11 except that the sets of internal coordinates **r** and **s** (see Sec. 2.1) are the same. These coordinates are defined under the ICDISTANCE keyword in the general MC-TINKER input file. The parameter file `param-testr12.prm` is the same as the `param-testr11.prm` in the test run 11.

INPUT FILES:

`testr12.81`	Configuration 1 input file
`testr12.82`	Configuration 2 input file
`testr12.83`	Electronic structure theory data file
`testr12.85`	General MC-TINKER input file
`testr12.dat`	General POLYRATE input file
`param-testr12.prm`	Modified TINKER parameter file
`run_testr12`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr12.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr12.fu6`	Full output file
`testr12.fu25`	Potential energies along the MEP
`testr12.fu26`	Generalized TS frequencies along the MEP
`testr12.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr12.fu61`	Stationary point information
`testr12.time`	Computing time

7.13. Test run 13

The abstraction of hydrogen from HBr by Cl by MCMM(9)

This run is an MCMM calculation of the Cl + HBr hydrogen abstraction reaction. Because it is a bimolecular reaction, the interpolation is carried out between the reactant and product wells. Some parameters in the parameter file `param-testr13.prm` are added in order to describe five atom types (Cl in HCl, Br in HBr, H, Cl, and Br atoms) that are undefined in the original MM3 force field. The electronic structure theory data is obtained at the MP2(FC)/6-31G(d) level of theory, where MP2 denotes second-order Møller-Plesset perturbation theory, and FC denotes frozen core. This test run is based on eleven Shepard points: nine electronic structure theory points, selected as described in Ref. [3], and the reactant-valley and product-valley wells, which are calculated with the MM3 force field with the additional parameters for the atom types that do not exist in the original MM3 force field. Three internal coordinates are used for the expansions in the Shepard interpolation step (Cl–H and H–Br distances, and Cl–H–Br angle), and three internal coordinates are used for calculating generalized distances between points in the Shepard interpolation step (Cl–H and H–Br distances and Cl–H–Br angle).

The STATUS keyword is set to *6* for REACT1, REACT2, PROD1, and PROD2, and the structures and frequencies obtained by electronic structure calculations are used as input for the POLYRATE input file. These data are used to calculate the partition functions for the reactant and product. The energy of reaction is also obtained by electronic structure calculations, and this value is used for EDIFF in the MC-TINKER input file and for the product energy in the POLYRATE input file. The STATUS keyword is set to *0* for the wells and the transition state.

The reaction path is scaled to a mass of *1* amu. The MEP is calculated by the Page-McIver method with a step size of *0.001* bohrs. The Hessian matrix is calculated and a generalized normal mode analysis in curvilinear internal coordinates is carried out every *0.01* bohrs. The rate constants are computed using TST and CVT. Tunneling contributions are computed using the ZCT, SCT, and LCG4 methods.

INPUT FILES:

`testr13.81`	Configuration 1 input file
`testr13.82`	Configuration 2 input file
`testr13.83`	Electronic structure theory data file
`testr13.85`	General MC-TINKER input file
`testr13.dat`	General POLYRATE input file
`param-testr13.prm`	Modified TINKER parameter file
`run_testr13`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr13.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr13.fu6`	Full output file
`testr13.fu25`	Potential energies along the MEP
`testr13.fu26`	Generalized TS frequencies along the MEP
`testr13.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr13.fu61`	Stationary point information
`testr13.time`	Computing time

7.14. Test run 14

The abstraction of hydrogen from CH_4 by OH by MCMM(11)

This run is an MCMM calculation of the OH + CH_4 hydrogen abstraction reaction. Because it is a bimolecular reaction, the interpolation is carried out between the reactant and product wells. Some parameters in the parameter file `param-testr14.prm` are added in order to describe one atom type (O in OH radical) that is undefined in the original MM3 force field. The electronic structure theory data is obtained at the oMPW1K/6-31+G(d,p) level of theory. This test run is based on thirteen Shepard points: eleven electronic structure theory points, selected as described in Ref. [3], and the reactant-valley and product-valley wells, which are calculated with the MM3 force field with the additional parameters for the atom type that does not exist in the original MM3 force field. Eighteen internal coordinates are used for the expansions in the Shepard interpolation step, and two of these coordinates are used for calculating generalized distances between points in the Shepard interpolation step.

The STATUS keyword is set to *6* for REACT1, REACT2, PROD1, and PROD2, and the structures and frequencies obtained by electronic structure calculations are used as input for the POLYRATE input file. These data are used to calculate the partition functions for the reactant and product. The energy of reaction is also obtained by electronic structure calculations, and this value is used for EDIFF in the MC-TINKER input file and for the product energy in the POLYRATE input file. The STATUS keyword is set to *0* for the wells and the transition state.

The reaction path is scaled to a mass of *1* amu. The MEP is calculated by the Page-McIver method with a step size of *0.001* bohrs. The Hessian matrix is calculated and a generalized normal mode analysis in curvilinear internal coordinates is carried out every *0.01* bohrs. The rate constants are computed using TST and CVT. Tunneling contributions are computed using the ZCT, SCT, and LCG4 methods. For the LCG4 method, only the tunneling contributions into the ground state are included.

This calculation should reproduce the CVT, CVT/ZCT, CVT/SCT, and CVT/LCT(0) results in Table 9-S of the Supporting Information of Ref. [3].

INPUT FILES:

`testr14.70`	General interface input file
`testr14.71`	Reactant 1 geometry input file
`testr14.72`	Reactant 2 geometry input file
`testr14.75`	Saddle point geometry input file
`testr14.81`	Configuration 1 input file
`testr14.82`	Configuration 2 input file
`testr14.83`	Electronic structure theory data file
`testr14.85`	General MC-TINKER input file
`testr14.dat`	General POLYRATE input file
`param-testr14.prm`	Modified TINKER parameter file
`run_testr14`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr14.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr14.fu6`	Full output file
`testr14.fu25`	Potential energies along the MEP
`testr14.fu26`	Generalized TS frequencies along the MEP
`testr14.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr14.fu61`	Stationary point information
`testr14.time`	Computing time

7.15. Test run 15

The abstraction of hydrogen from CH_4 by NH_2 by MCMM(10)

This run is an MCMM calculation of the NH_2 + CH_4 hydrogen abstraction reaction. Because it is a bimolecular reaction, the interpolation is carried out between the reactant and product wells. Some parameters in the parameter file `param-testr15.prm` are added in order to describe one atom type (N in NH_2 radical) that is undefined in the original MM3 force field. The electronic structure theory data is obtained at the oMPW1K/6-31+G(d,p) level of theory. This test run is based on twelve Shepard points: ten electronic structure theory points, selected as described in Ref. [3], and the reactant-valley and product-valley wells, which are calculated with the MM3 force field with the additional parameters for the atom type that does not exist in the original MM3 force field. Twenty-three internal coordinates are used for the expansions in the Shepard interpolation step, and two of these coordinates are used for calculating generalized distances between points in the Shepard interpolation step.

The STATUS keyword is set to *6* for REACT1, REACT2, PROD1, and PROD2, and the structures and frequencies obtained by electronic structure calculations are used as input for the POLYRATE input file. These data are used to calculate the partition functions for the reactant and product. The energy of reaction is also obtained by electronic structure calculations, and this value is used for EDIFF in the MC-TINKER input file and for the product energy in the POLYRATE input file. The STATUS keyword is set to *0* for the wells and the transition state.

The reaction path is scaled to a mass of *1* amu. The MEP is calculated by the Page-McIver method with a step size of *0.001* bohrs. The Hessian matrix is calculated and a generalized normal mode analysis in curvilinear internal coordinates is carried out every *0.01* bohrs. The rate constants are computed using TST and CVT. Tunneling contributions are computed using the ZCT, SCT, and LCG4 methods.

INPUT FILES:

`testr15.81`	Configuration 1 input file
`testr15.82`	Configuration 2 input file
`testr15.83`	Electronic structure theory data file
`testr15.85`	General MC-TINKER input file
`testr15.dat`	General POLYRATE input file
`param-testr15.prm`	Modified TINKER parameter file
`run_testr15`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr15.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr15.fu6`	Full output file
`testr15.fu25`	Potential energies along the MEP
`testr15.fu26`	Generalized TS frequencies along the MEP
`testr15.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr15.fu61`	Stationary point information
`testr15.time`	Computing time

7.16. Test run 16

The abstraction of hydrogen from CH_3Cl by CH_2F by MCMM(11)

This run is an MCMM calculation of the CH_2F + CH_3Cl hydrogen abstraction reaction. Because it is a bimolecular reaction, the interpolation is carried out between the reactant and product wells. Some parameters in the parameter file `param-testr16.prm` are added in order to describe interactions and one atom type (C–F in CH_2F, C–Cl in CH_2Cl, and migrating H) that are undefined in the original MM3 force field. The electronic structure theory data is obtained at the oMPW1K/6-31+G(d,p) level of theory. This test run is based on thirteen Shepard points: eleven electronic structure theory points, selected as described in Ref. [3], and the reactant-valley and product-valley wells, which are calculated with the MM3 force field with the additional parameters that do not exist in the original MM3 force field. Twenty-seven internal coordinates are used for the expansions in the Shepard interpolation step, and two of these coordinates are used for calculating generalized distances between points in the Shepard interpolation step.

The STATUS keyword is set to *6* for REACT1, REACT2, PROD1, and PROD2, and the structures and frequencies obtained by electronic structure calculations are used as input for the POLYRATE input file. These data are used to calculate the partition functions for the reactant and product. The energy of reaction is also obtained by electronic structure calculations, and this value is used for EDIFF in the MC-TINKER input file and for the product energy in the POLYRATE input file. The STATUS keyword is set to *0* for the wells and the transition state.

The reaction path is scaled to a mass of *1* amu. The MEP is calculated by the Page-McIver method with a step size of *0.001* bohrs. The Hessian matrix is calculated and a generalized normal mode analysis in curvilinear internal coordinates is carried out every *0.01* bohrs. The rate constants are computed using TST and CVT. Tunneling contributions are computed using the ZCT, SCT, and LCG4 methods.

INPUT FILES:

`testr16.81`	Configuration 1 input file
`testr16.82`	Configuration 2 input file
`testr16.83`	Electronic structure theory data file
`testr16.85`	General MC-TINKER input file
`testr16.dat`	General POLYRATE input file
`param-testr16.prm`	Modified TINKER parameter file
`run_testr16`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr16.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr16.fu6`	Full output file
`testr16.fu25`	Potential energies along the MEP
`testr16.fu26`	Generalized TS frequencies along the MEP
`testr16.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr16.fu61`	Stationary point information
`testr16.time`	Computing time

7.17. Test run 17

The abstraction of hydrogen from C_3H_8 (at the secondary C position) by OH by MCMM(9)

This run is an MCMM calculation of the OH + C_3H_8 hydrogen abstraction reaction. Because it is a bimolecular reaction, the interpolation is carried out between the reactant and product wells. Some parameters in the parameter file `param-testr17.prm` are added in order to describe one atom type (O in OH radical) that is undefined in the original MM3 force field. The electronic structure theory data are obtained at the oMPW1K/6-31+G(d,p) level of theory. This test run is based on eleven Shepard points: nine electronic structure theory points, selected as described in Ref. [3], and the reactant-valley and product-valley wells, which are calculated with the MM3 force field with the additional parameters for the atom type that does not exist in the original MM3 force field. Thirty-eight internal coordinates are used for the expansions in the Shepard interpolation step, and two of these coordinates are used for calculating generalized distances between points in the Shepard interpolation step.

The STATUS keyword is set to *6* for REACT1, REACT2, PROD1, and PROD2, and the structures and frequencies obtained by electronic structure calculations are used as input for the POLYRATE input file. These data are used to calculate the partition functions for the reactant and product. The energy of reaction is also obtained by electronic structure calculations, and this value is used for EDIFF in the MC-TINKER input file and for the product energy in the POLYRATE input file. The STATUS keyword is set to *0* for the wells and the transition state.

The reaction path is scaled to a mass of *1* amu. The MEP is calculated by the Page-McIver method with a step size of *0.001* bohrs. The Hessian matrix is calculated, and a generalized normal mode analysis in curvilinear internal coordinates is carried out every *0.01* bohrs. The rate constants are computed using TST and CVT. Tunneling contributions are computed using the ZCT, SCT, LCT, and μOMT methods, where SCT is CD-SCSAG and LCT is LCG4. For the LCT and μOMT methods, only the tunneling contributions into the ground state and the first two excited states are included.

INPUT FILES:

`testr17.81`	Configuration 1 input file
`testr17.82`	Configuration 2 input file
`testr17.83`	Electronic structure theory data file
`testr17.85`	General MC-TINKER input file
`testr17.dat`	General POLYRATE input file
`param-testr17.prm`	Modified TINKER parameter file
`run_testr17`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr17.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr17.fu6`	Full output file
`testr17.fu25`	Potential energies along the MEP
`testr17.fu26`	Generalized TS frequencies along the MEP
`testr17.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr17.fu61`	Stationary point information
`testr17.time`	Computing time

7.18. Test run 18

The abstraction of hydrogen from C_3H_8 (at the secondary C position) by OH using the MCMM(11) PES.

This run is an MCMM calculation of the OH + C_3H_8 hydrogen abstraction reaction where reaction occurs at the secondary C position of C_3H_8. It is the same as test run 17 except for the following differences:

(1) The electronic structure theory data is obtained at the MPW1K/6-31+G(d,p) level of theory rather than oMPW1K/6-31+G(d,p). It employs 10 supplementary (non-stationary) points.

(2) Some changes are made in the general POLYRATE input file:

   a. The MEP range is $-3.00\ a_0 < s < +2.00\ a_0$ instead of $-2.90\ a_0 < s < +1.30\ a_0$.
   b. The input stationary geometries are slightly different due to the update of MPW1K, and are given in a.u. rather than Å.
   c. The normal mode frequencies are in a.u. instead of wavenumbers.
   d. The WELLP and WELLR sections are removed.
   e. The *`ediff0`* is -0.02629139 a.u. rather than -0.0263133 a.u. due to the update from oMPW1K to MPW1K. This change of *`ediff0`* is also present in the general MC-TINKER input file `testr18.85`.

(3) The value of the coefficient *D* (see mc-tinker2007 manual) is 0.01 (cf. *D* = 0.2 in test run 17).

The parameter file `param-testr18.prm` is the same as the `param-testr17.prm` in test run 17. The reactant and product configuration files are the same as those in test run 17.

INPUT FILES:

`testr18.81`	Configuration 1 input file
`testr18.82`	Configuration 2 input file
`testr18.83`	Electronic structure theory data file
`testr18.85`	General MC-TINKER input file
`testr18.dat`	General POLYRATE input file
`param-testr18.prm`	Modified TINKER parameter file
`run_testr18`	Script file to run MC-TINKERATE

OUTPUT FILE (distributed):

`testr18.fu15`	Summary output file

OUTPUT FILES (not distributed):

`testr18.fu6`	Full output file
`testr18.fu25`	Potential energies along the MEP
`testr18.fu26`	Generalized TS frequencies along the MEP
`testr18.fu27`	Generalized TS geometries along the MEP in the XYZ format of the XMOL program
`testr18.fu61`	Stationary point information
`testr18.time`	Computing time

8. BIBLIOGRAPHY

[1] Kim, Y.; Corchado, J. C.; Villa, J.; Xing, J.; Truhlar, D. G. *J. Chem. Phys.* **2000**, *112*, 2718.

[2] Tishchenko, O.; Truhlar D. G. *J. Chem. Theory Comput*. **2007**, *3*, 938.

[3] Albu, T. V.; Corchado, J. C.; Truhlar, D. G. *J. Phys. Chem. A* **2001**, *105*, 8465.

[4] London, F. *Z. Electrochem.* **1929**, *35*, 552.

[5] Eyring, H.; Polanyi, M. *Z. Phys. Chem.* **1931**, *B12*, 279.

[6] Kimball, G. E.; Eyring, H. *J. Am. Chem. Soc.* **1932**, *54*, 3876.

[7] Coulson, C. A.; Danielsson, U. *Arkiv Fysik* **1954**, *8*, 239.

[8] Sato, S. *Bull. Chem. Soc. Japan* **1955**, *28*, 450.

[9] Ellison, F. O. *J. Am. Chem. Soc.* **1963**, *85*, 3540.

[10] Parr, C. A.; Truhlar, D. G. *J. Phys. Chem.* **1971**, *75*, 1844.

[11] Kuntz, P. J. In *Atom-Molecule Collision Theory*; Bernstein, R. B., Ed.; Plenum: New York, 1979; p. 79.

[12] Warshel, A.; Weiss, R. M. *J. Am. Chem. Soc.* **1980**, *102*, 6218.

[13] Warshel, A.; Russell, S.; Weiss, R. M. In *Chemical Approaches to Understanding Enzyme Catalysis: Biomimectic Chemistry and Transition State Analogs*; Green, B. S., Ashani, Y., Chipman, D., Eds.; Elsevier: Amsterdam, 1981; p. 267.

[14] Warshel, A. *Computer Modeling of Chemical Reactions in Enzymes and Solutions*; John Wiley & Sons: New York, 1991.

[15] Chang, Y. T.; Miller, W. H. *J. Phys. Chem.* **1990**, *94*, 5884.

[16] Chang, Y. T.; Minichino, C.; Miller, W. H. *J. Chem. Phys.* **1992**, *96*, 4341.

[17] Pulay, P.; Fogarasi, G. *J. Chem. Phys.* **1992**, *96*, 2856.

[18] Chuang, Y.-Y.; Truhlar, D. G. *J. Phys. Chem. A* **1998**, *102*, 242.

[19] Ischtwan, J.; Collins, M. A. *J. Chem. Phys.* **1994**, *100*, 8080.

[20] Nguyen, K. A.; Rossi, I.; Truhlar, D. G. *J. Chem. Phys.* **1995**, *103*, 5522.

[21] Downing, J. W.; Michl, J. In *Potential Energy Surfaces and Dynamics Calculations*; Truhlar, D. G., Ed.; Plenum: New York, 1981; p. 199.

[22] Truhlar, D. G.; Isaacson, A. D.; Garrett, B. C. In *Theory of Chemical Reaction Dynamics*; Baer, M., Ed.; CRC Press: Boca Raton, FL, 1985; p. 65.

[23] Liu, Y. P.; Lu, D. H.; Gonzalez-Lafont, A.; Truhlar, D. G.; Garrett, B. C. *J. Am. Chem. Soc.* **1993**, *115*, 7806.

[24] Steckler, R.; Hu, W.-P.; Liu, Y.-P.; Lynch, G. C.; Garrett, B. C.; Isaacson, A. D.; Melissas, V. S.; Lu, D.-h.; Truong, T. N.; Rai, S. N.; Hancock, G. C.; Lauderdale, J. G.; Joseph, T.; Truhlar, D. G. *Comput. Phys. Commun.* **1995**, *88*, 341.

[25] Fernandez-Ramos, A.; Truhlar, D. G. *J. Chem. Phys.* **2001**, *114*, 1491.

[26] Fernandez-Ramos, A.; Ellingson, B. A.; Garrett, B. C.; Truhlar, D. G. In *Reviews in Computational Chemistry*; Lipkowitz, K. B. and Cundari, T. R.; Eds.; Wiley-VCH, Hoboken: NJ, 2007; Vol. 23, p. 125.

[27] Hu, W. P.; Liu, Y. P.; Truhlar, D. G. *J. Chem. Soc., Faraday Trans.* **1994**, *90*, 1715.

[28] Lynch, B. J.; Zhao, Y.; Truhlar, D. G. *J. Phys. Chem. A* **2003**, *107*, 1384.

[29] Lin, H.; Pu, J.; Albu, T. V.; Truhlar, D. G. *J. Phys. Chem. A* **2004**, *108*, 4112.

[30] Tishchenko, O.; Truhlar, D. G. *J. Phys. Chem. A* **2006**, *110*, 13530.

[31] Lynch, B. J.; Fast, P. L.; Harris, M.; Truhlar, D. G. *J. Phys. Chem. A* **2000**, *104*, 4811.

[32] Adamo, C.; Barone, V. *J. Chem. Phys.* **1998**, *108*, 664.

[33] Perdew, J. P. In *Electronic Structure of Solids*, Ziesche, P.; Eschrig, H., Eds,; Akademie Verlag: Berlin, 1991; p. 11.

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

The computers, operating systems, and compiler versions on which various versions of MC-TINKERATE have been tested are listed in Tables 9.1 and 9.1. The compiler and loader commands used for testing the code are listed in Tables 9.3 and 9.4. Note that the FORTRAN compiler on the Compaq computer uses a Compaq FORTRAN 90 compiler (which is called `f77`) with some compilation flags.

Table 9.1. Operating systems on the various machines on which various versions of the code have been tested

Version	Machine	Operating system
8.8	IBM SP Power3	AIX 4.3.4.0
	SGI Origin 3800 R14000	IRIX 6.5.12f
	Compaq ES40 Alpha 500	Tru64 4.0f
9.1	IBM SP Power3	AIX 4.3
	IBM Regatta Power4	AIX 5.1
2007	IBM Regatta Power4	AIX 5.2
2008	IBM Regatta Power4	AIX 5.3
2010	BladeCenter Linux cluster	SuSE Linux Enterprise Server 9 (x86 64) – 2.6.5 kernel

Table 9.2. Compiler versions used for various machines

Version	Machine	Compiler version
8.8	IBM SP Power3	XL FORTRAN 6.1.0.3
	SGI Origin 3800 R14000	MIPSpro 7.3.1.2m
	Compaq ES40 Alpha 500	Compaq FORTRAN V5.4-1283-46ABA

9.1	IBM SP Power3	XL FORTRAN 6.1.0.3
	IBM Regatta Power4	XL FORTRAN 7.1.1.2
2007	IBM Regatta Power4	IBM XL Fortran V10.1 for AIX
2008	IBM Regatta Power4	IBM XL Fortran V10.1 for AIX
2010	BladeCenter Linux cluster	Intel Fortran complier version 9.1

Table 9.3. Compiler options used for various machines

Version	**Machine**	**Compiler commands**
8.8	IBM SP Power3	`xlf -c -qdpc -qmaxmem=-1`
	SGI Origin 3800 R14000	`f77 -c -O3 -static -64`
	Compaq ES40 Alpha 500	`f77 -c -fpe3 –O3`
9.1	IBM SP Power3	`xlf -c -qdpc -qmaxmem=-1`
	IBM Regatta Power4	`xlf -c -qdpc -qmaxmem=-1`
2007	IBM Regatta Power4	`xlf -c -qdpc -qmaxmem=-1`
2008	IBM Regatta Power4	`xlf -c -qdpc -qmaxmem=-1`
2010	BladeCenter Linux cluster	`ifort –c –w –O3`

Table 9.4. Loader commands used for various machines

Version	**Machine**	**Loader commands**
8.8	IBM SP Power3	`xlf -o`
	SGI Origin 3800 R14000	`F77 -64 -o`
	Compaq ES40 Alpha 500	`F77 -o`
9.1	IBM SP Power3	`xlf -o`
	IBM Regatta Power3	`xlf -o`
2007	IBM Regatta Power4	`xlf -o`
2008	IBM Regatta Power4	`xlf -o`
2010	BladeCenter Linux cluster	`ifort -o`

10. REVISION HISTORY

10.1. TINKERATE

In this section we will list the changes made in TINKERATE, which (as reviewed in Section 10.2) may be considered to be a precursor version of MC-TINKERATE.

10.1.1. TINKERATE–version 8.4 (December 1999)

TINKERATE–version8.4 was the first official version of the TINKERATE program.

10.1.2. TINKERATE–version 8.4.1 (February 2000)

The TINKERATE distribution and manual were modified in order to update from TINKER–version 3.5mn1 to TINKER–version 3.5mn2.

10.1.3. TINKERATE–version 8.4.2 (November 2000)

The TINKERATE distribution and manual were modified in order to update from TINKER–version 3.5mn2 to TINKER–version 3.5mn3.

This version also fixes a bug in the code that was causing the program to stop when a doubly degenerate bend was used in the definition of internal coordinates.

10.1.4. TINKERATE–version 8.5 (November 2000)

This version allows the use, in calculating generalized distances between points in the Shepard interpolation, of internal coordinates that are not present in the set of internal coordinates used for the generalized normal mode analysis. This new feature required changes in the *rdk*, *h12evbi*, and *h12egh* subprograms and the addition of two new subprograms: *ytrincar* and *yreadint*.

The TINKERATE distribution and manual were also modified in order to interface TINKERATE with version 8.5.1 of the POLYRATE program.

10.2. RELATION OF MC-TINKERATE TO TINKERATE

Two computer programs, MC-TINKERATE and TINKERATE, have been developed for using the MCMM algorithm for the interpolation of potential energy surfaces and using the interpolated potential energy surfaces for modeling chemical dynamics and calculating reaction rates in conjunction with POLYRATE computer program. Both programs use TINKER subroutines to calculate the molecular mechanics terms needed in the MCMM algorithm.

The TINKERATE program was the first of the two programs. This program provided an interface between the POLYRATE dynamics program and TINKER, which is a molecular mechanics program.

Another computer program for carrying out MCMM calculations, developed subsequent to TINKERATE, is MC-TINKER. It is more general in that MCMM calculations can be carried out at any given geometry without using POLYRATE. Thus MC-TINKER may be used in stand-alone mode or in conjunction with other molecular dynamics programs, as well as with POLYRATE. In addition, MC-TINKER has some additional features not present in TINKERATE that are presented in more detail in the MC-TINKER manual.

The MC-TINKERATE program provides an interface between the POLYRATE dynamics program and MC-TINKER, and, in that respect, it is a different program than TINKERATE. However, due to the fact that performs same type of calculations as in TINKERATE, MC-TINKERATE can be also seen as a totally re-organized version of TINKERATE. Accordingly, and to make the relation to POLYRATE versions more clear, we numbered the new MC-TINKERATE program versions with numbers larger than those used for the TINKERATE program.

With the release of MC-TINKERATE, the other program, TINKERATE, is obsolete and is no longer distributed.

10.3. MC-TINKERATE

MC-TINKERATE version numbers have the general form 200x/Px'.y'.z'–M200x'' where 200x is the overall version number, x'.y'.z' is the number of the version of POLYRATE on which it is based, and 200x'' is the number of the version of MC-TINKER on which it is based. (In older versions, 200x may be replaced by x.y, and 200x'' may be replaced by x''.y'' or x''.y''.z''.) The part before the solidus, i.e., 200x is unique because it changes if the interface changes, the POLYRATE version changes, or the MC-TINKER version changes. Thus, the part after the solidus is redundant information, and it may be omitted for simplicity if desired.

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.

10.3.1. MC-TINKERATE–version 8.8/P8.7.2–M1.0 (March 2002)

MC-TINKERATE–version 8.8 is the first official version of MC-TINKERATE. See Section 10.2 for discussion of the predecessor of MC-TINKERATE.

10.3.2. MC-TINKERATE–version 9.1/P9.1–M1.0.1 (December 2003)

MC-TINKERATE–version 9.1 is updated from MC-TINKERATE–version 8.8. This version interfaces POLYRATE 9.1, and MC-TINKER 1.0.1, which is a bug-fixed version for MC-TINKER 1.0.

The *dat* subroutines are duplicated in POLYRATE 9.1 and MC-TINKER 1.0.1. This does not cause problems for most platforms. However, in order to avoid possible conflicts in the future, the MC-TINKER 1.0.1 machine-dependent subroutine files are not used in MC-TINKERATE 9.1. Instead, the subdirectory `srcmcr/dattim` of MC-TINKERATE 9.1 provides new subroutine files that were modified from MC-TINKER 1.0.1. These new subroutines call the *dat* subroutine in POLYRATE 9.1 exclusively.

10.3.3. MC-TINKERATE–version 9.1/P9.1–M1.1 (April 2005)

MC-TINKERATE–version 9.1 is updated to interface POLYRATE 9.1, and MC-TINKER 1.1, which is an enhanced version for MC-TINKER 1.0.1. (See the MC-TINKER manual for details.)

10.3.4. MC-TINKERATE–version 2007/P9.1–M2007 (July 2007)

MC-TINKERATE–version 2007 is updated to interface POLYRATE 9.1 and MC-TINKER 2007, which is a corrected and enhanced version of MC-TINKER 1.1. (See the MC-TINKER manual for details.)

Test runs were modified so that internal coordinates used in these tests now satisfy conditions (i) and (ii) stated in Section 2.1. Some test runs were modified to adopt more appropriate van der Waals parameters in the molecular mechanics energy expression because it has been recently shown [30] that significant performance enhancement can be obtained in this way. *Note*: because of these changes, and because of the corrections made to the MCMM gradients (MC-TINKER versions prior to 1.1.1 had incorrect gradients, see the MC-TINKER manual), the locations of Shepard points in test runs do not precisely correspond to the prescription given in Ref. [3]. To follow this prescription, one would have to iteratively reselect new locations for all Shepard points (except those at reaction saddle points) on the new MCMM potential energy surfaces that incorporate these changes. Such ″inconsistency″ should not be a problem when considering that these test runs are included for testing installations of the codes for correctness on different computer platforms. Furthermore, as discussed in Ref. [30], calculated reaction rate constants should not be overly sensitive to the location of Shepard points relative to the minimum energy paths.

10.3.5. MC-TINKERATE–version 2008/P9.1–M2008 (XXX 2008)

MC-TINKERATE–version 2008 is updated to interface POLYRATE 9.1 and MC-TINKER 2008.
Authors: Titus V. Albu, Oksana Tishchenko, José C. Corchado, Yongho Kim, Jordi Villà, Jianhua Xing, Hai Lin, Masahiro Higashi, and Donald G. Truhlar
Updated by: Masahiro Higashi and Donald G. Truhlar

The ISHMM keyword in the RESONANCE section and the GAMESS option of the FORMHESS keyword, which has been added in the MC-TINKER 2008, are available in this version. Note that an EE-MM or EE-MCMM calculation is unavailable in the current version of MC-TINKERATE program (See Sec. 6.5.1.).

10.3.6. MC-TINKERATE–version 2010/P9.1–M2010 (October 2010)

MC-TINKERATE–version 2010 is updated to interface POLYRATE 9.1 and MCSI 2009-1.
Authors: Titus V. Albu, José C. Corchado, Yongho Kim, Jordi Villà,
Jianhua Xing, Hai Lin, Oksana Tishchenko, Masahiro Higashi, and Donald G. Truhlar
Updated by: Osanna Tishchenko and Donald G. Truhlar

The new capabilities available in the MCSI 2009-1 are now available in this version.