May 4, 2012

MultiWell Program Suite User Manual

(MultiWell-2012)

**John R. Barker, N. F. Ortiz, J. M. Preses,
L. L. Lohr,A. Maranzana, P. J. Stimac,
T. L. Nguyen, and T. J. Dhilip Kumar**

University of Michigan
Ann Arbor, MI 48109-2143

Contact:
jrbarker@umich.edu
(734) 763-6239

http://aoss-research.engin.umich.edu/multiwell/

Preface

MultiWell Literature Citation

Please cite the following papers to acknowledge results obtained using this version of the MultiWell Program Suite:

For citing the basic MultiWell package

In the following, <version> refers to the version number (e.g. 2012.1) and <year> refers to the year of publication (e.g. 2012)

a) MultiWell-<version> Software, ,<year>, designed and maintained by John R. Barker with contributors Nicholas F. Ortiz, Jack M. Preses, Lawrence L. Lohr, Andrea Maranzana, Philip J. Stimac, T. Lam Nguyen, and T. J. Dhilip Kumar; University of Michigan, Ann Arbor, MI; http://aoss.engin.umich.edu/multiwell/.
b) John R. Barker, Int. J. Chem. Kinetics, 33, 232-45 (2001).
c) John R. Barker, Int. J. Chem. Kinetics, 41, 748-763 (2009).

Additional references for **`adensum`**

All preceding references, plus...

d) M. Basire, P. Parneix, and F. Calvo, J. Chem. Phys. **129**, 081101 (2008).
e) F. Wang and D. P. Landau, Phys. Rev. Letters **86**, 2050 (2001).
f) Thanh Lam Nguyen and John R. Barker, J. Phys. Chem. A., 114, 3718–3730 (2010).

Additional references for **`sctst`**

All preceding references, plus...

g) W. H. Miller, J. Chem. Phys. **62**, 1899 (1975).
h) W. H. Miller, Faraday Discuss. Chem. Soc. **62**, 40 (1977).
i) W. H. Miller, R. Hernandez, N. C. Handy, D. Jayatilaka, and A. Willets, Chem. Phys. Letters **172**, 62 (1990).
j) R. Hernandez and W. H. Miller, Chem. Phys. Lett. **214** (2), 129 (1993).
k) T. L. Nguyen, J. F. Stanton, and J. R. Barker, Chem. Phys. Letters **499**, 9 (2010).
l) T. L. Nguyen, J. F. Stanton, and J. R. Barker, J. Phys. Chem. A 115, 5118 (2011).

About the Authors

JOHN BARKER wrote most of the code, originally based on his 1983 paper on stochastic methods and subsequent developments.[1-6] NICK ORTIZ (as an undergraduate student) wrote most of the code for MomInert. JACK PRESES (Brookhaven Nat'l Lab.) added several helpful features and helps to maintain the Windows versions. LARRY LOHR has contributed to the development of hindered rotor subroutines. ANDREA MARANZANA (University of Turin) contributed and maintains the codes for automatically generating input files for the MultiWell Suite from output files produced in quantum chemical calculations. PHIL STIMAC implemented

1-dimensional quantum tunneling *via* an unsymmetrical Eckart barrier. LAM NGUYEN installed a method for using quantum eigenvalues for hindered internal rotations, instead of using approximate values (as was done in versions earlier than v.2009.0), wrote a code for computing the effective mass of large amplitude motions, and helped in developing codes for non-separable vibrations[6] and semi-classical transition state theory.[7] DHILIP KUMAR wrote several scripts for automatically using output files from electronic structure programs to build input/output files for MultiWell.

Comments and Bug Reports

Please send suggestions and bug reports (!) to John R. Barker (jrbarker@umich.edu). It is through bug reports that most errors are found.

Acknowledgements

Thanks go to the following people for particularly helpful suggestions, discussions, debugging, or other assistance:

Amity Andersen
Hans-Heinrich Carstensen
Benj FitzPatrick
Michael Frenklach
David M. Golden
Keith D. King
Robert G. ('Glen') MacDonald
Nigel W. Moriarty
Colleen Shovelin
Gregory P. Smith
A. Bencsura
Theodore S. Dibble
Erin Greenwald
John Herbon
George Lendvay
David M. Matheu
William F. Schneider
Robert M. Shroll
Ralph E. Weston, Jr.
David Edwards
Gabriel Robert Da Silva

Some sections of these computer codes were developed as part of research funded by NSF (Atmospheric Chemistry Division), NASA (Upper Atmosphere Research Program), and NASA (Planetary Atmospheres).

Disclaimer: *This material is based in part upon work supported by the National Science Foundation under Grant No. 0344102. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.*

CONTENTS

1. Getting Started

1.1 Software Tools in the MultiWell Suite

MultiWell

Calculates time-dependent concentrations, yields, vibrational distributions, and rate constants as functions of temperature and pressure for unimolecular reaction systems that consist of multiple stable species, multiple reaction channels interconnecting them, and multiple dissociation channels from each stable species. Reactions can be reversible or irreversible. Can include tunneling and/or the effects of slow intramolecular vibrational energy redistribution (IVR).[8,9] NOTE: k(E)'s can be calculated by other programs and read in (see Input file description below).

DenSum

Carries out exact counts for sums and densities of states via the Stein-Rabinovitch extension[10] of the Beyer-Swinehart algorithm.[11] Optionally, the Whitten-Rabinovitch approximation can be used. The following types of separable modes are accepted:

a) vibrations (harmonic and anharmonic)
b) free rotations (classical and quantized)
c) hindered rotations, symmetrical and unsymmetrical (quantized eigenvalues)
d) particle in a box
e) translation

Thermo

Calculates entropy, heat capacity, and H(T)-H(0) for individual species, based on vibrational frequencies, moments of inertia, internal rotation barriers, and electronic state properties. It includes all of the types of modes listed for DenSum. It calculates equilibrium constants, which are useful for obtaining recombination rate constants from the corresponding unimolecular decomposition rate constants. When provided with parameters for reactants and the transition state, it uses canonical transition state theory to calculate rate constants (including tunneling corrections based on the 1-D unsymmetrical Eckart barrier). By using input files generated by Programs 'adensum' and 'sctst' (see below), Thermo can include the effects of fully coupled anharmonic vibrations and/or compute thermal rate constants using the fully-coupled anharmonic semi-classical transition state theory formulated by W. H. Miller and coworkers.

MomInert

Calculates principal moments of inertia for chemical species and approximate reduced moments of inertia for internal rotors. Requires the Cartesian coordinates for the atoms in the molecule, as obtained from many software packages in common use (e.g., HyperChem, Chem3D, MOPAC, GAUSSIAN, C-FOUR, etc.)

Gauss2Multi

Reads output files from the Gaussian electronic structure program and generates data files for the other four programs in the MultiWell Suite. (Parameters not calculated using Gaussian must be added by hand.)

lamm

Uses Cartesian coordinates along a path (obtained from electronic structure calculations) to compute the effective mass for large amplitude motions, such as internal hindered rotations, inversion vibrations, and multi-well potentials. For this purpose, `lamm` is a better choice than `MomInert`. A script, `gauss2lamm`, is provided to read Gaussian output files and construct (in part) the data file needed for `lamm`.

doloops, ansum, and adensum

These three codes are available for computing sums and densities of states for non-separable anharmonic vibrations, based on the perturbation theory expansion. Programs `doloops` and `ansum` are appropriate for use with small molecules, while `adensum` is appropriate for most molecules. Both `doloops` and `adensum`, like `DenSum`, produce output files that can be used directly as input files for `MultiWell`.

sctst

This code is for using the semi-classical transition state theory formulated by W. H. Miller and coworkers[12-15] to compute cumulative reaction probabilities (analogous to the sum of states for the transition state). This enables computation for microcanonical rate constants (k(E)s), which are needed for master equation simulations (using program 'MultiWell'). The code also computes the partition function corresponding to the CRP at a set of temperatures from 50 K to 3400 K and generates a data file that can be used by program 'Thermo' to conveniently compute thermal rate constants using the SCTST of Miller and coworkers.

1.2 Directory Structure

The MultiWell directory is organized as follows:

```
/multiwell-<version>        [Main MultiWell Directory]
     /bin                                          [binary executables]
     /doc                                [version history, license, etc.]
     /examples                               [Example input/output files]
          /densum-examples
          /mominert-examples
          /multiwell-examples
          /thermo-examples
          /gauss2multi-examples
     /thermo-database           [thermodynamics data in format for Thermo]
     /scripts              [admin scripts for running tests and examples]
     /src
          /ansum                                   [source code for ansum]
               /test                                          [test files]
          /adensum                               [source code for adensum]
               /test                                          [test files]
          /doloops                               [source code for doloops]
               /test                                          [test files]
          /gauss2multi                       [source code for gauss2multi]
               /test                                          [test files]
          /densum                                 [source code for densum]
               /test                                          [test files]
          /mominert                             [source code for mominert]
               /test                                          [test files]
          /multiwell                           [source code for multiwell]
               /test                                          [test files]
          /thermo                                 [source code for thermo]
               /test                                          [test files]
          /gauss2multi                       [source code for gauss2multi]
               /test                                          [test files]
```

1.3 Starting Up Linux/Unix (and Mac OS X) Version

In this version, binary executables must be compiled. A Makefile script located in the Main Directory sequentially calls makefiles in each source directory. To execute the Makefile and compile the binary executables, type the command “**`make`**” (omit the quotation marks and use all lower case characters) followed by "**`return`**". After compiling, the binary executables are stored in directory **`bin`**.

To test that the compiled codes are operating correctly, run the script **`runtest_all`** by going to the **`scripts`** directory and typing the command “**`./runtest_all`**”. This script will run the newly compiled codes and allow you to compare the output files to “test” outputs stored in the **`test`** directories associated with the source code. The new outputs will differ from the test outputs with regard to date and computation time. If compiled with FORTRAN-77 compilers other than GNU g77, there may be minor numerical differences. If other differences appear, then it is possible that the compiled codes are not working properly.

It is highly recommended that users do ***not*** place user data files, etc., in directory **`/multiwell-<version>`** (see Section 1.2) Instead, users should create individual directories for user models (see Section 1.5) and execute MultiWell from within those directories. This approach makes it very easy to replace the entire directory **`/multiwell-<version>`** with a newer version. Programs in the MultiWell Suite are executed as described in Section 2.5.

All of the programs in the MultiWell Suite can be executed in the manner described in Section 2.5.

1.4 Starting Up Windows Version

Simple Method

In this version, binary executables (application, or .EXE files) have already been compiled and are found in directory **`bin`**. To run any of the programs in the MultiWell Suite, the steps are as follows:

1. Prepare a data file (for instructions, see the User Manual and the Examples directory) and place it in a directory devoted to your Model (see Section 1.5 of the User manual for recommended Model directory structure). For Densum calculations, for example, the data file (densum.dat) may reside in the directory Model/vibs.

2. Make an alias of the executable of interest (e.g. densum.exe) and place the alias in the same directory (folder) as the data file. For Densum calculations, for example, if the data file [(densum.dat) resides in the directory Model/vibs, the alias should also be place in directory Model/vibs.

3. Double-click the alias. The output files (e.g. densum.out, etc.) will be written to the same directory.

Expert Users

Expert Users can set the PATH in Windows and run all of the programs in the MultiWell Suite in the manner described in Section 2.5.

1.5 Directory Structure for Models

Because there are multiple input and output files associated with any reaction model, we have found that it is most convenient to organize them in the following way. First, keep all models in a directory named "**`models`**". Then within that directory, each model should have its own individual directory named for the specific model. In that way, it is easy to replace the multiwell directory when updates are issued.

Recommended Model directory structure:

```
/model          [name of model; e.g. "c2h6", "cloocl", etc.]
     multiwell.dat                                    [MultiWell data file]
     multiwell.out                                  [MultiWell output file]
     multiwell.sum                                  [MultiWell output file]
     multiwell.rate                                 [MultiWell output file]
     multiwell.flux                                 [MultiWell output file]
     multiwell.array                                [MultiWell output file]
     readme.model      [read-me file for description, literature citation, etc.]
     /DensData  [REQUIRED for sums an densities of states used by MultiWell]
     /multidata                             [for stored multiwell data files]
     /vibs      [densum data files (suggested file suffix: ".vibs")]
     /thermodata             [thermo data files (suggested file suffix: ".therm")]
     /momidata  [mominert data files (suggested file suffix: ".mom")]
     /results   [stored output]
```

1.6 Example Models and Files

Several examples are provided for each of the codes: MultiWell, DenSum, MomInert, Thermo, Gauss2Multi, doloops, adensum, and ansum. The DenSum examples include a set of cases discussed in the literature: useful for testing the accuracy of Densum.

The directory `multiwell/thermo-database` contains a file that has been formatted for use by **`thermo`**, the code used for calculating thermodynamic properties. Data for several dozen molecules are included in the file.

2. MultiWell Master Equation Code

Codes, examples, and this manual are available from the MultiWell Program Suite web site: http://aoss.engin.umich.edu/multiwell/

2.1 Brief Description

MultiWell calculates time-dependent concentrations, yields, vibrational distributions, and rate constants as functions of temperature and pressure for unimolecular reaction systems which consist of multiple stable species, multiple reaction channels interconnecting them, and multiple dissociation channels from each stable species. The stochastic method is used to solve the resulting Master Equation. Users may supply unimolecular reaction rates, sums of states and densities of states, or optionally use Forst's Inverse Laplace Transform method[16-18] to calculate k(E). For weak collisions, users can select from among many collision models, or provide user-defined functions.

The code is intended to be relatively easy to use. It is designed so that very complicated and very simple unimolecular reaction systems can be handled via the data file. Restructuring of the code and recompiling are NOT necessary to handle even the most complex systems.

MultiWell is most suitable for time-dependent non-equilibrium systems. The real time needed for a calculation depends mostly upon the number of collisions during a simulated time period and on the number of stochastic trials needed to achieve the desired precision. For slow reaction rates and precise yields of minor reaction products, the code will require a long run time, but it will produce results. For long calculation runs, we often just let it run overnight.

MultiWell is based on the Gillespie Exact Stochastic algorithm,[19-21] as modified and implemented in our laboratory.[1-4] It has been described in considerable detail in a recent publication.[22] An example calculation has also been published.[23]

In the example,[23] chemical activation and shock wave simulations were carried out for a system consisting of six isomers and 49 energy-dependent unimolecular reactions. The isomers were interconnected by reversible isomerization reactions, and each isomer could also decompose, resulting in 14 sets of products. Many of the capabilities of MultiWell are illustrated in that paper.[23]

2.2 Terminology

The following sketch shows the potential energy as a function of reaction coordinate for a typical unimolecular system with multiple wells.

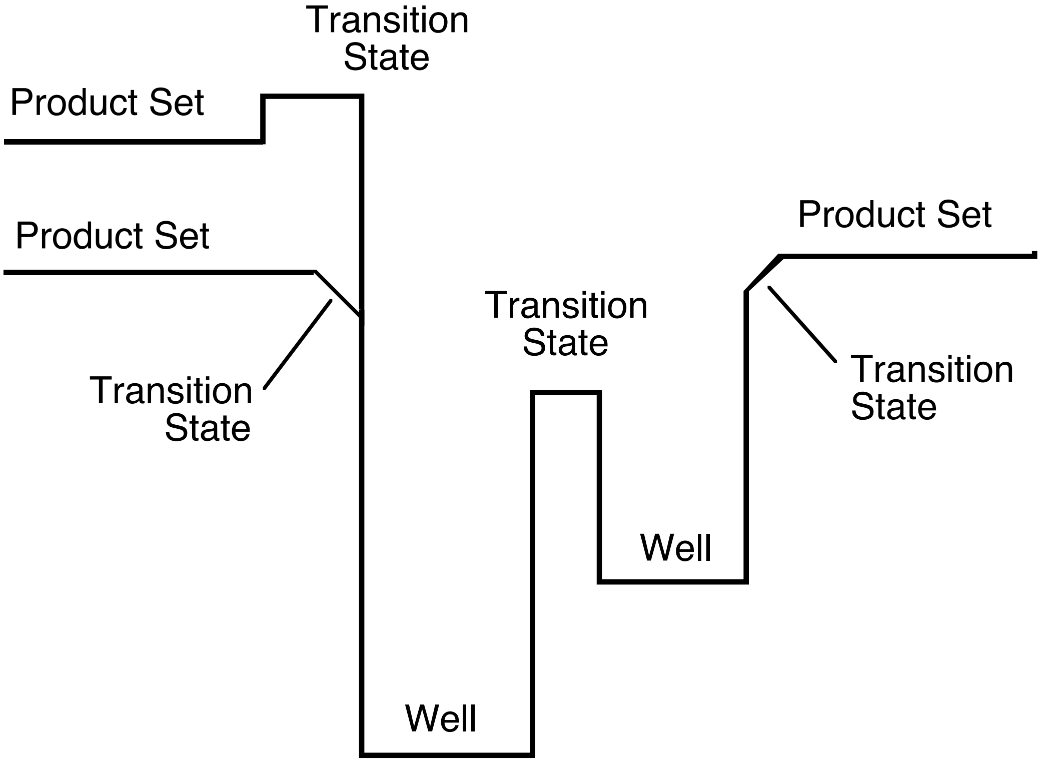
- "Wells" are chemical species corresponding to local minima on the potential energy surface.

- "Transition states" for reaction are defined in the usual way.

- "Product sets" are the fragmentation products corresponding to irreversible reaction *via* a given transition state.

In MultiWell, each Well, each Transition State, and each Product Set is given a name and is designated by an index number, as described below. The names and index numbers are arbitrary, but they provide unique identification.

Collisional activation and deactivation can take place within each well and therefore energy transfer parameters are designated for each. In most cases, energy transfer parameters are poorly known, if known at all, and thus it is usually convenient to assign the same energy transfer parameters to all of the wells.



The energy scheme assumed by MultiWell is shown schematically by the following diagram:

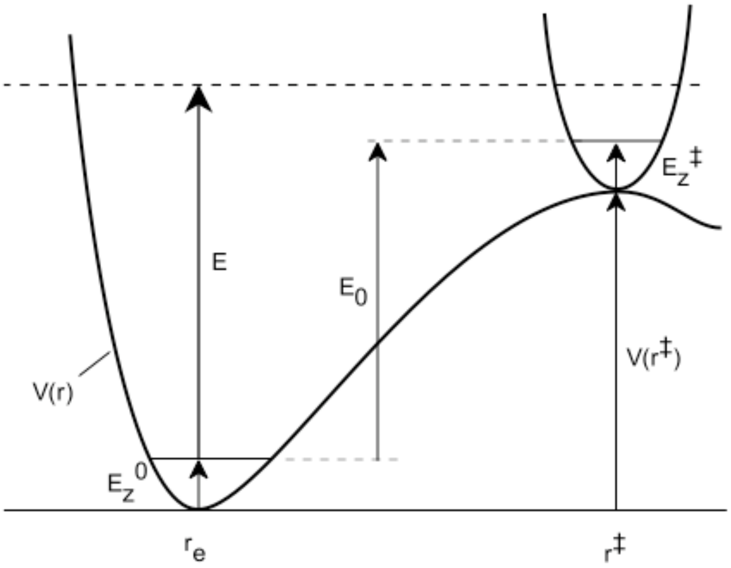


Figure 0. Energy schematic. The active energy *E* is measured from the zero point energy of the reactant well. The critical energy E_0 is the zero point energy difference between the transition state and the reactant well.

2.3 Default Array Dimensions

Array dimensions can be changed by revising the include files: **`Declare1.inc`**, **`Declare2.inc`**, and **`Declare3.inc`**. Currently, the array dimensions are set for use with up to the following defaults. (See the section above on "Terminology".)

Up to 50 Wells.

Up to 25 Reaction channels per well for a total of up to 100 Reactions.

Up to 50 Product Sets.

Double Arrays with user-selected sizes up to **14000** elements (reduced from 20000 in an earlier version).

100 time steps.

200 energy bins x 10 time steps for reporting vibrational distributions.

2.4 Notes on FORTRAN source code and compilation

- There are many explanatory comments embedded in the source code.

- The source code is written for the GNU FORTRAN compiler for LINUX; a makefile is distributed with the code. The code can also be compiled with almost any other FORTRAN compiler.

- A few sections of the source code are platform-dependent (e.g., OPEN statements, date & time calls, and file & path names). These can be found in the following source code segments:

  MultiWell.f [main program]
  DateTime.f
  DensArray.f
  Estart.f
  RateArray.f

- The following compiler options are required, or recommended:
  - **static storage (REQUIRED)**.
  - case-sensitive (RECOMMENDED).
  - variables and constants initialized to zero (RECOMMENDED).
  - double precision transcendentals (RECOMMENDED).
  - promote REAL and COMPLEX to double precision (RECOMMENDED).
  - basic optimizations (RECOMMENDED).

2.5 MultiWell Input Files and Program Execution

The default input data filename is **multiwell.dat** (all lower case). Starting with version 2008.1, it is possible to change the input data file name and run multiple sessions in the same directory at the same time, each with a user-selected **FileName**.

To run MultiWell using the default filename (multiwell.dat):

LINUX/UNIX: in the directory where the input data file and the auxiliary directory DensData reside, type:

```
<PATH>/multiwell-<version>/bin/multiwell <RETURN>
```

where <PATH> designates the PATH to `/multiwell-<version>`. If directory `/multiwell-<version>` resides in the user home directory, type:

```
~/multiwell-<version>/bin/multiwell <RETURN>
```

WINDOWS in a DOS window: in the directory where the input data file and the auxiliary directory DensData reside, type:

```
<PATH>/multiwell-<version>/bin/multiwell <RETURN>
```

To run MultiWell using a user-defined filename (FileName.dat):

Follow the same procedures described above, but type:

```
<PATH>/multiwell-<version>/bin/multiwell FileName <RETURN>
```

2.6 MultiWell Output Files

Output files with an identical FileName erased and written-over for every calculation. To be saved, they must be re-named. Use a word processor/editor capable of wide-open (no truncation of lines) output, because the output can be hundreds of characters in width, depending on the number of species and products. In Linux, Xemacs, Emacs, Nedit and other editors are available for this purpose. For Macintosh OS X, "Tex-Edit Plus" (share-ware available at http://www.nearside.com/trans-tex/) and "TextWrangler" (free-ware available at http://www.barebones.com/) are very convenient word processors for text files, although full-featured word processors can be used as well.

FileName.out

Time-dependent output of concentrations and average energies. Also includes summaries of input parameters. The time-dependent quantities are the instantaneous values at the time indicated: they are not averaged over the time interval. Hence, the averages are only over the number of trials.

FileName.sum

Summary output file intended for convenient calculations of fall-off curves and other pressure-dependent quantities. This file gives all of the header material in the full output file, but instead of the time-dependent results, only the final results of each simulation are given in the form of a summary table.

FileName.rate

Time-dependent output of average unimolecular rate "constants" (which vary with time in non-steady-state systems). Many trials are needed to accumulate good statistics. To improve statistics, the binned results correspond to the number of visits to the bin (which can be many times larger than the number of trials) and are averaged over the time-bin.

FileName.dist

Time-dependent vibrational distributions in Wells (not initial or final products). Only the non-zero array elements are listed. Many trials are needed to accumulate good statistics. Note that the distributions are normalized according to the number of stochastic trials. Therefore, the sum of the array elements for a chemical species (Well) at a given time is equal to the fractional population of that species at that time. Thus the distributions report not only the relative populations as functions of energy and time, but also the growth and decay of species concentrations.

FileName.array

Tabulations of all energy-dependent input data. Includes tables of densities of states, specific rate constants, collision probabilities and normalization factors, and initial energy distributions.

FileName.flux

Tabulates chemical flux via each of the unimolecular reactions. Useful for identifying quasi-equilibrium situations and for tracing chemical pathways.

2.7 MultiWell Input Data File (FileName.dat)

This datafile uses free input format.

NOTES ON FREE INPUT FORMAT: Fields separated by delimiters.
- Standard delimiters on most platforms: commas and spaces.
- Additional delimiters acceptable on some platforms: tabs.
- CHARACTER constants enclosed in apostrophes (') are accepted on most platforms. Some platforms will accept CHARACTER constants without their being enclosed in apostrophes, but then they cannot contain any of the delimiter characters.

MULTIWELL MAJOR INPUT OPTIONS
1. Densities of states are read from an external file created by DenSum, or other code.
2. Specific Rate constants: `k`(E)
    a) RRKM theory via sums of states read from an external file (created by DenSum, or other code).
    b) `k`(E) values read from external file.
    c) Reversible and/or irreversible reactions.
3. Initial energy distributions:
    a) thermal (with an optional energy offset), calculated internally.
    b) chemical activation, calculated internally.
    c) delta function
    d) distribution can be read from an external file.
4. Separate initial vibrational temperature and translational temperature.
5. Can incorporate the effects of slow intramolecular vibrational energy redistribution (IVR).
6. Can include tunneling via an unsymmetrical Eckart barrier.

MULTIWELL INPUT DATA FILE FORMAT

Note: Starting with version 2.0, the data file format is no longer compatible with previous versions.

SECTION A: PHYSICAL PARAMETERS

Line 1

`TITLE` (up to 100 characters)

Line 2

`Egrain1, imax1, Isize , Emax2, IDUM`

`Egrain1` energy grain size of first segment in "double arrays", see Note (units: cm^{-1})

`imax1` size of first segment of double array; selected so that sums or densities of states is a smooth function of energy (less than ~1% relative fluctuations). Note that `imax1` must be less than `Isize`.

`Isize` user-selected size of double array. The Default array size starting in version 2.08 is set for a maximum of 14000 elements in the INCLUDE file "**declare1.inc**". (The array size is defined by `Imax`=14000 in declare1.inc.) This large maximum array size allows users to select any value of `Isize` ≤ 14000 elements without having to recompile the code. If array sizes greater than 14000 elements are needed, the Imax can be changed in the Linux/Unix version by deleting old object files (by typing 'make clean' in /multiwell/src/multiwell) and then recompiling (by typing 'make').

`Emax2` maximum energy of 2nd segment of double arrays (units: cm^{-1})

`IDUM` random number seed (integer); EXAMPLE: "2113989025"

***** **NOTE: "Double arrays"** have two sections: segment 1 consists of `imax1` equally spaced (`Egrain1`) data ranging from E=0; segment 2 consists of equally spaced values from E=0 to `Emax2`; the size of the second segment is (`Isize - imax1`); the energy grain of the second segment is `Emax2`/(`Isize - imax1` - 1).

(Section A, continued...)

Line 3

`Punits`, `Eunits`, `Rotatunits`
[***It is required that the three keywords be entered in this exact order!***]

`Punits` one of the following pressure units keywords:
'`BAR`', `'ATM'`, or `'MCC'`[for molecules/cc] (Note that `'TOR'` is no longer accepted.)

`Eunits` one of the following energy units keywords:
`'CM-1'`, `'KCAL'`, or `'KJOU'` for cm^{-1}, kcal/mole or kJ/mole

`Rotatunits` one of the following keywords for rotational information:
`'AMUA'`, `'GMCM'`, '`CM-1`', `'MHZ'`, `'GHZ'`(for moments of inertia in units of amu.$Å^2$ or g.cm^2, and rotational constants in units of cm^{-1}. MHz, or GHz)
(some combinations of upper/lower case are also accepted)

Line 4

`Temp ,Tvib`

`Temp` translational temperature (units: Kelvin)

`Tvib` initial vibrational temperature (units: Kelvin)

For shock-tube simulations, `Temp` is set equal to the shock (translational) temperature and `Tvib` is set equal to the vibrational temperature prior to the shock (usually room temperature).

Line 5

`Np` number of pressures

Line 6

`PP(1), PP(2), ..., PP(Np)`

List of `Np` pressures

SECTION B: PARAMETERS FOR WELLS AND FOR PRODUCT SETS

Line 7

NWells , NProds

NWells	number of "wells" (includes irreversible product sets)
NProds	number of entrance/exit channels; each channel has a product set associated with it.

Line 8

IMol , MolName , HMol , MolMom , Molsym , Molele , Molopt

(REPEAT NWells times: once for each well.)

IMol	index number for well (1 ... NWells)
MolName	name of well (≤10 characters)
HMol	enthalpy of formation at 0 K (units defined by keyword)
MolMom	moment of inertia for 2-dimensional external rotation (units defined by keyword on Line 3)
Molsym	external symmetry number for well (see Section 9.5 for a discussion)
Molele	electronic partition function for well (REAL number)
Molopt	number of optical isomers for well (see Section 9.5 for a discussion)

See Section 9.4 for a discussion of proper input for External Molecular Rotations.

Line 9

IMol , MolName , Hmol

(REPEAT NProds times: once for each entrance/exit channel, i.e. for each product set.)

IMol	index for channel (NWells+1...NWells+NProds)
MolName	name of Product set (max 10 characters)
Hmol	enthalpy of formation at 0 K (units defined by keyword on Line 3) [ignored unless tunneling is used]

***** NOTE: the numbering of entrance/exit channels starts with NWells+1.

Line 10

SigM, EpsM, AmuM, Amu

SigM	Lennard-Jones σ (Å) for collider
EpsM	Lennard-Jones ε/k_B (Kelvins) for collider

`AmuM` Molecular weight (g/mole) of collider
`Amu` Molecular weight (g/mole) of reactant

Line 11

`Mol`, `Sig`, `Eps`, `ITYPE`, `DC`(1), `DC`(2), ... , `DC`(8)

(REPEAT Lines 11 and 12 `NWells` times: once for each well.)

`Mol` index number of Well
`Sig` Lennard-Jones σ (Å) for this well
`Eps` Lennard-Jones ε/k_B (Kelvins) for this well
`ITYPE` selects model type in Subroutine PDOWN (see below for description of collision models). Model types and explanations are given below.
`DC`(8) eight (8) coefficients for energy transfer model

Line 12

`LJQM` keyword for type of collision rate constant:
'`LJ`' for Lennard-Jones collision rate constant.
'`QM`' for quantum mechanical total collision rate constant[24]

(REPEAT Lines 11 and 12 `NWells` times: once for each well.)

SECTION C: PARAMETERS FOR TRANSITION STATES AND REACTIONS

Line 13

NForward number of forward unimolecular (not recombination) reactions to be input.

Line 14

Mol, ito, TS, RR, j, k, l, AA, EE, KEYWORD, KEYWORD, KEYWORD, KEYWORD, KEYWORD

(REPEAT NForward times: once for each forward reaction.)

Mol index of reactant well

ito index of entrance/exit channel or well

TS Name of transition state (up to 10 characters)

RR 2-D external moment of inertia (units defined by keyword on Line 3) ; **See Section 9.4 for a discussion of proper input for External Molecular Rotations.**

j external symmetry number for TS (see Section 9.5 for a discussion)

Qel electronic partition function for TS (REAL number)

l number of optical isomers for TS (see Section 9.5 for a discussion)

AA A-factor for reaction (units: s^{-1}); only used for ILT method, but ALWAYS read in

EE reaction critical energy (E_0), relative to ZPE of reactant (Mol) (See Section 9.9 for the appropriate E_0 when using SCTST)

KEYWORDS ALWAYS READ FIVE KEYWORDS, IN ANY ORDER. Select one from each of the Five Groups below. **See Section 2.10 (FATAL INPUT ERRORS) for a list of incompatible choices.**

Group 1

'NOREV' for neglecting the reverse reaction

'REV' for calculating reverse reaction rate (automatically treated as NOREV for dissociation reactions).

Group 2

'FAST' for neglecting limitations due to IVR

'SLOW' for including IVR limitations; line 14b contains parameters (see below).

Group 3

'NOTUN' for neglecting tunneling

'TUN' for including tunneling via unsymmetrical Eckart barrier; line 14a contains parameters (see below).

Group4

'NOCENT' for no centrifugal correction

'CENT1' for quasi-diatomic centrifugal correction with 1 adiabatic external rotations

'CENT2' for quasi-diatomic centrifugal correction with 2 adiabatic external rotations

`'CENTX'` for legacy centrifugal correction with 2 adiabatic external rotations (not recommended)
[**Note:** the calculated k_∞ is numerically the same for all options.]

Group 5

`'SUM'` File containing sums of states (usually generated by Densum)
`'ILT'` Inverse laplace transform method for k(E).
`'RKE'` External file containing k(E): <TS filename>+'`.rke`' (e.g. '`TS-1.rke`')

NOTE: `k(E)`'s can be calculated by other programs and read in as an external file (KEYWORD = '`RKE`').

Line 14: Supplementary Lines

The following supplementary lines provide additional information corresponding to some of the Keywords in Line 14. ***The supplementary line immediately follows the line invoking the Keyword.*** (On the rare occasion when more than one supplementary line is required, they must be entered in the order given here.)

Supplementary Line 14a

`'TUN', vimag(Mol,i)`

This line appears only if `KEYWORD 'TUN'` was used in Line 14. It gives the imaginary frequency (cm^{-1}) for the specified reaction. It can only be used when '`NOCENT`' is invoked. Cannot be used simultaneously with '`ILT`' or '`RKE`'.

Supplementary Line 14b

`'SLOW', vivr(Mol,i), vave(Mol,i), kcivr(Mol,i), tivr(Mol,i), civr(Mol,i,1), civr(Mol,i,2), civr(Mol,i,3)`

This line appears only if `KEYWORD 'SLOW'` was used in Line 14. It gives parameters for the IVR transmission coefficient for this reaction:

$$Transmission\ Coefficient = \frac{k_{IVR}(E) + k_{IVR}^{c}[M]}{k_{IVR}(E) + k_{IVR}^{c}[M] + \nu_{ivr}}$$

where ν_{IVR} is the characteristic reaction frequency (as in RRK unimolecular reaction rate theory). At energies above the IVR threshold energy (i.e. $E \geq E_{IVR}^{0}$), the IVR rate constant $k_{IVR}(E)$ is:

$$k_{IVR}(E - E_{0r}) = civr(Mol,i,1) + civr(Mol,i,2) \times (E - E_{0r}) + civr(Mol,i,3) \times (E - E_{0r})^{2}$$

where E (expressed in cm^{-1}) is the energy relative to the reactant zero point energy and E_{0r} is the reaction critical energy (which may include the centrifugal correction).

`vivr(Mol,i)` Characteristic frequency (cm^{-1}) for the reaction; ν_{IVR}/s^{-1} = `vivr`*2.9979×10^{10}.

`vave(Mol,i)` Average frequency (cm^{-1}) of the reactant; used to define an upper limit to k_{IVR}, the IVR rate constant: $k_{IVR} \leq 2$*`vave`*2.9979×10^{10}.

`kcivr` Bimolecular rate constant k^c_{IVR} [cm^3 $molecule^{-1}$ sec^{-1}] for collision-induced IVR. In the absence of other information, k^c_{IVR} may be estimated as approximately equal to the quantum mechanical total collision frequency bimolecular rate constant, as obtained from the MultiWell output (see Line #12, above).

`tivr(Mol,i)` IVR threshold energy (cm^{-1}), measured from the reaction critical energy (i.e. E^0_{IVR}-E_{0r}).

`civr(Mol,i,..)` Three (3) coefficients for second order polynomial fit of k_{ivr} (s^{-1}) as a function of E-E_{0r} (cm^{-1}; energy measured from the reaction critical energy).

SECTION D: CALCULATION SPECIFICATIONS

Line 15

`Ntrials`, `Tspec`, `Tread`, `KEYTEMP`, `Molinit`, `IR`, `Einit`

`Ntrials` number of trials

`Tspec` a `KEYWORD` that specifies meaning of `Tread` (CHARACTER*4)
'TIME' indicates `Tread` = max time simulated (Tlim)
'COLL' indicates `Tread` = max time simulated is calculated from the specified maximum number of collisions experienced by initial well number (`Molinit`).

`Tread` maximum simulated **time** or maximum number of **collisions** (see `Tspec`, above).

`KEYTEMP` a `KEYWORD` that specifies the type of initial energy distribution
`'DELTA'`: Monoenergetic at energy `Einit`
`'THERMAL'`: Thermal (`Tvib`) with energy offset `Einit`
`'CHEMACT'`: Chemical activation (`Tvib`) from "product" #`IR`
`'EXTERNAL'`: Read cumulative energy distribution from external file "`multiwell.pstart`" placed in directory "DensData"

`Molinit` index of initial well

`IR` index number of the "product set" which reacts to produce `Molinit` via chemical activation; neglected if `'CHEMACT'` is not specified.

`Einit` initial energy (relative to ZPE of `Molinit`); neglected if `'CHEMACT'` is specified; same units as `Eunits`.

Line 16

BLANK LINE TO INSURE THAT THE LAST LINE IS FOLLOWED BY A CARRIAGE RETURN (needed for all READ statements). THE CARRIAGE RETURN IS EASILY OVERLOOKED!

2.8 COLLISION MODELS

(see **Line 11** in multiwell data file described above)

This selection of collision models includes most of the empirical models discussed in the literature. Function subroutine "Pdown.f" can be revised to include additional models.

For general guidance in selecting models and parameters, see Barker et al.[25]

For the **EXPONENTIAL MODEL**, use ITYPE=1 with coefficient C(4) set equal to zero so that the second exponential term is equal to zero; Model Types 12 or 13 can also be used.

ITYPE	
1	Biexponential Model
2	Density-weighted Biexponential Model
3	Off-set Gaussian with constant offset and E-dependent width
4	Biexponential Model with energy-dependent fraction
5	Generalized Gaussian with energy-dependent exponent
6	Generalized Gaussian plus Exponential term
7	Weibull Model
8	Lorentzian Step-Ladder Model
9	Exponential+Elastic Model
10	Klaus Luther's empirical function
11	Radiationless transition empirical function
12	Exponential Model with alpha(E)=linear + exponential
13	Exponential Model with alpha(E) switching function
14	Boltzmann-weighted exponential

FUNCTION AND COEFFICIENT DESCRIPTIONS:

ITYPE = 1 for Biexponential Model

Pdown = (1-C(4))*EXP(-(E-EE)/Alpha1) + C(4)*EXP(-(E-EE)/Alpha2)

Alpha1 = C(1) + [E*C(2) + E*E*C(3)]*T**C(8)
Alpha2 = C(5) + [E*C(6) + E*E*C(7)]*T**C(8)

ITYPE = 2 for Density-of-States-weighted Biexponential Model

Pdown = ρ(E)*((1-C(4))*EXP(-(E-EE)/Alpha1) + C(4)*EXP(-(E-EE)/Alpha2))

Alpha1 = C(1) + [E*C(2) + E*E*C(3)]*T**C(8)
Alpha2 = C(5) + [E*C(6) + E*E*C(7)]*T**C(8)

ITYPE = 3 for Off-set Gaussian with constant offset and E-dependent width

Pdown = EXP(-(0.5*(E-EE-C(4))/Alpha1)**2)

C(4) = constant off-set, Alpha1 is the std. dev.
Alpha1 = C(1) + [E*C(2) + E*E*C(3)]*T**C(8)

ITYPE = 5 for Generalized Gaussian with energy-dependent exponent

Pdown = EXP(-[(E-EE)/Alpha]**Exponent)

Alpha = C(1) + [E*C(2) + E*E*C(3)]
exponent = C(5) + [E*C(6) + E*E*C(7)]

ITYPE = 6 for Generalized Gaussian plus Exponential term

Pdown =(1-C(6))*EXP(-[(E-EE)/Alpha1]**Exponent) + C(6)*EXP(-(E-EE)/Alpha2)

Alpha1 = C(1) + [E*C(2) + E*E*C(3)]
Alpha2 = C(7) + E*C(8)
Exponent = C(4) + E*C(5)

ITYPE = 7 for Exponential Model with Switching function

Pdown = EXP(-(E-EE)/Alp)

Alpha1 = C(1) + E*C(2)
Alpha2 = C(3) + E*C(4)
Alp = Alpha1 + 0.5*(Alpha2 - Alpha1)*(1. - TANH((C(5) - E)/C(6)))

ITYPE = 8 for Lorentzian Step-Ladder Model

Pdown = 1 / [(E-EE-Alpha)^2 + Width^2]

Alpha = C(1) + [E*C(2) + E*E*C(3)]
Width = C(5) + [E*C(6) + E*E*C(7)]

ITYPE = 9 for Exponential+Elastic Model

Pdown = [F/(F+C(4))]*EXP(-(E-EE)/Alpha) + elastic

Alpha = C(1) + [E*C(2) + E*E*C(3)]*T**C(8)
F = C(5) + [(E/C(6))**C(7)]*T**C(8); when E=EE

ITYPE = 10 for Klaus Luther's empirical function

Pdown = EXP [-[(E-EE)/Alpha]^Beta]

Alpha = C(1) + E*C(2)
Beta = C(3)

ITYPE = 11 for radiationless transition empirical Function

Pdown = EXP [-(E-EE)/Alpha]'

Alpha = C(1)*[1-exp[-(E/C(2))^C(3)] + C(4)]

ITYPE = 12 for Exponential Model with alpha(E)= Linear+exponential

Pdown = EXP(-(E-EE)/Alpha1)

Alpha1 = C(1) + E*C(2) + C(3)*exp(-E/C(4))

ITYPE = 13 for Exponential Model with Alpha(E) Switching function

Pdown = EXP(-(E-EE)/Alp)

Alp = Alpha1 + 0.5*(Alpha2 - Alpha1)*(1. - TANH((C(5) - E)/C(6)))
Alpha1 = C(1) + E*C(2)
Alpha2 = C(3) + E*C(4)

ITYPE = 14 for Boltzmann-Weighted Exponential
(motivated by recent trajectory calculations[26])

Pdown = B(T;EE,E)*EXP(-(E-EE)/Alp)

B(T;EE,E) = SQRT(rho(EE)*exp(-(EE-E)/RT))
where rho(EE) = density of states
Alp = C(1)

2.9 FORMAT OF EXTERNAL DATA FILES

Location: EXTERNAL DATA FILES MUST BE STORED IN FOLDER (DIRECTORY) "DensData", which is in the folder (directory) from which the MultiWell application is executed.

Densities, Sums and k(E)'s

Creation: Typically, the external files for densities of states and sums of states are generated using the program DenSum, which calculates sums and densities via exact counts. A similar format is used for specific rate constants, as described here.

FILE NAMES

For sums and densities of states: CHARACTER*10 + '.dens'
Example: 'Ethyl-Cl .dens'

For specific rate constants [k(E)'s]: CHARACTER*10 + '.rke'
Example: 'HCl+C2H4.rke'

FILE FORMAT

Note: Starting with version 2008.1, " ____.dens" files, which contain sums and densities of states, may have a block of comment lines preceding Line 1 (below). The first and last comment lines in the block must have *exactly* the following 46 characters:

```
**************INPUT DATA SUMMARY**************
```

Line 1

Name of species or transition state [CHARACTER*10]: up to 10 characters (skipped when read by MultiWell, and therefore can be left blank)

Line 2

Title/Comment line [CHARACTER*100]: up to 100 characters (skipped when read by MultiWell, and therefore can be left blank)

Line 3

`Egrain1 , imax1 , Emax2 , Isize ,Viblo`
[REAL, INTEGER, REAL, INTEGER, REAL]

Note that this ordering of parameters is ***not*** the same as in multiwell.dat. If `Egrain1`, `imax1`, & `Emax2` do not match those for the MultiWell run (see line #2 of the MultiWell data file), then execution terminates with a message. In density and sums of states files, `Viblo` is the lowest vibrational frequency for a given species; it is used in the collision step routines. In rate constant input files, A real number must be input for `Viblo`, but it is ignored by MultiWell.

Line 4

Column label (skipped when read by MultiWell, and therefore can be left blank)

For sums of states, this line might read:

" No. (cm-1)   Density   Sum"

For a rate constant input file, this line might read:

" No.   Energy Dummy RateConst"

Line 5

(Repeated Isize times)

j, A, B, C

[INTEGER,REAL,REAL,REAL]

FOR FILES CONTAINING SUMS & DENSITIES
j = array element index (a total of Isize elements)
A = energy [starting with E = 0.0] (units: cm-1)
B = density of states (units: states/cm-1)
C = sum of states (dimensionless)

FOR FILES CONTAINING SPECIFIC RATE CONSTANTS [k(E)'s]
j = array element index (a total of Isize elements)
A = energy [starting with E = 0.0] (units: cm-1)
B = dummy real number (ignored by MultiWell)
C = k(E) (units: s-1)

Line 6

BLANK LINE TO INSURE THAT THE LAST LINE IS FOLLOWED BY A CARRIAGE RETURN (needed for all READ statements). THE CARRIAGE RETURN IS EASILY OVERLOOKED.

Initial Energy Distribution Function

File name: "multiwell.pstart"

Location: EXTERNAL DATA FILES MUST BE STORED IN FOLDER (DIRECTORY) "DensData", which is in the folder (directory) from which the MultiWell application is executed.

Line 1

`Jsize`, `Hstart`, `Edel`

`Jsize` user-selected array length (≤14000 elements). `Jsize` must be chosen to be large enough (for a given `Hstart`) so that the entire initial energy distribution is represented.

`Hstart` energy grain (express in cm^{-1}) for initial energy distribution: "`Pstart`". `Hstart` should be ≥`Egrain1` used to generate sums and densities of states.

`Edel` energy origin of array, relative to zero point energy of initial excited well (express in cm^{-1})

Line 2

(REPEAT `Jsize` times)

`DUMMY`, `Pstart`

`DUMMY` Must be input, but ignored by MultiWell (may be equal to E, for example)

`Pstart` NORMALIZED cumulative initial energy distribution function. Thus, `Pstart(i)` ranges monotonically from `Pstart(1)`=0 (approximately), up to `Pstart(Jsize)` = 1.0.

(Note that other columns may be present in the data file, but they will be ignored.)

Line 3

BLANK TO INSURE THAT THE LAST LINE IS FOLLOWED BY A CARRIAGE RETURN (needed for all READ statements). THE CARRIAGE RETURN IS EASILY OVERLOOKED.

2.10 FATAL INPUT ERRORS

A FATAL INPUT ERROR is reported and execution is stopped if any of the following parameters for a well or transition state does not match the corresponding value from Line 2 of multiwell.dat:

```
Egrain1, imax1, Isize , Emax2
```

Incompatible Keywords or Input Data:

ILT and **CENT1**, **CENT2**, or **CENTX**
ILT and **TUN**
RKE and **TUN**
CENT and **TUN**
CENT1, **CENT2**, or **CENTX** with **TSmom(Mol,nchann)** ≤ 0
(moment of inertia or rotational constant of the transition state)

3. DenSum: *Separable* Sums and Densities of States

This computer code carries out exact counts for sums and densities of states for *separable* degrees of freedom via the Stein-Rabinovitch[10] extension of the Beyer-Swinehart algorithm.[11] Optionally, the Whitten-Rabinovitch approximation[27,28] can be used.

Default array dimensions are easily changed in file src/densum/declare.inc

150 degrees of freedom (enough for a 50 atom molecule)

50000 energy grains (enough for 100000 cm^{-1} energy maximum, with 2 cm^{-1} grains)

Output Files

densum.out	File containing easily read complete output.
`<FNAME>`.dens	File formatted for input into MultiWell.

Example data and output files are given in the examples directory: `multiwell/examples/densum-examples`.

3.1 Functional Forms

(An)Harmonic Vibration (vib)

The energy of an anharmonic oscillator is

$$E = \omega_e\left(v+\frac{1}{2}\right)+\omega_e x_e\left(v+\frac{1}{2}\right)^2 - ZPE$$

where ω_e is the harmonic frequency and the zero point energy ($v = 0$) is

$$ZPE = \omega_e\left(\frac{1}{2}\right)+\omega_e x_e\left(\frac{1}{2}\right)^2 .$$

For a Morse oscillator, the anharmonicity is

$$\omega_e x_e = -\frac{\omega_e^{\,2}}{4D_e}$$

where D_e is the Morse oscillator well depth (from the bottom of the potential well). ***Note the sign convention for anharmonicity used here is not the usual convention.***

By an "Observed frequency" (key word "`obs`") for a vibrational mode, we mean the fundamental frequency, which is the 0-1 transition frequency for the specific mode when all other vibrational quantum numbers equal zero. When $\omega_e x_e \neq 0$ for a separable mode, the fundamental frequency ω^f is related to the harmonic frequency according to the following expression:

$$\omega_e = \omega^f - 2\omega_e x_e$$

For an "observed" frequency, Densum and Thermo compute ω_e from this expression and use it to calculate the vibrational energy levels.

Classical Rotation (rot)

DenSum uses the method of Astholz et al.[29] For expressions, see Appendix 5 of Robinson and Holbrook,[30] or Section 4.5 of Holbrook et al.[31] Also see Appendix (Theoretical Basis), Section A.2.2 Note that starting with version 2.04, classical rotations are treated purely classically; previously, a semi-classical approximation was imposed (which amounted to requiring that the sum of states is equal to unity at E = 0). **See Section 9.4 for a discussion of proper input for External Molecular Rotations.**

Quantized Rotation (qro)

When $J = 0$, E = 0 and degeneracy (g) is unity. When $J > 0$, the degeneracy (g) and the energy depend on the dimensionality (d) of the rotor[32]:

$d = 1$	$g = 2$	$E = BJ^2$	$J = 0,1,2,3,...$
$d = 2$	$g = (2J + 1)$	$E = BJ(J+1)$	$J = 0,1,2,3,...$
$d = 3$	$g = (2J + 1)(2J + 1)$	$E = BJ(J+1)$	$J = 0,1,2,3,...$

Prior to version 2.04, the sum of states at E=0 was set equal to unity, regardless of rotational symmetry number (σ). The current version gives the sum of states equal to $1/\sigma$ at E=0.

See Section 9.4 for a discussion of proper input for External Molecular Rotations.

K-rotor Rotation (kro)

This degree of freedom type should **NOT** be selected for normal calculations using MultiWell. It is provided in DenSum only for special purposes. **See Section 9.4 for a discussion of proper input for External Molecular Rotations.**

Particle in a Box (box)

1-D box of length L with particle of mass m:

$$E = \frac{h^2 n^2}{8mL^2} - \frac{h^2}{8mL^2} \qquad \text{for } n = 1,2,3,...$$

where E is the energy in excess of the "zero" point energy (when n=1).

The effects of higher dimensions are additive. For example, consider a 2-D box of dimensions L_x, L_y:

$$E = \frac{h^2}{8m}\left\{\left[\frac{n_x}{L_x}\right]^2 + \left[\frac{n_y}{L_y}\right]^2 - \left[\frac{1}{L_x}\right]^2 + \left[\frac{1}{L_y}\right]^2\right\} \qquad \text{for } n_i = 1,2,3,...$$

Because the energies in multiple dimensions are additive, DenSum and Thermo consider them separately. The "frequency parameter" that is needed by DenSum and Thermo: $h^2/(8mL^2)$. Furthermore, DenSum and Thermo assume $L_x = L_y = ...= L$.

1-D Hindered Rotation, Symmetrical (hra, hrb, hrc)

These types are intended for use with symmetrical internal rotors (e.g. a CH_3 rotor). Relationships among parameters for a hindered rotor:

$$B / cm^{-1} = \frac{16.85763}{I_r / amu\ \mathring{A}^2}$$

$$V = \frac{1}{B}\left[\frac{\omega}{n}\right]^2$$

where B is the rotational constant, I_r is the reduced moment of inertia, V is the barrier to internal rotation, and ω is the small amplitude harmonic frequency (cm^{-1}). For convenience, three methods are provided input of any combination of two independent parameters from the set: V, ω, I_r.

1-D Hindered Rotation, Unsymmetrical (hrd)

This type is intended for unsymmetrical rotors (e.g. the CH_2Cl rotor). For this type, one must provide the torsional potential energy and reduced moment of inertia (or rotational constant) as functions of the dihedral angle χ (radians).

For convenience, three forms of the torsional potential energy are accepted (all coefficients in units of cm^{-1}):

Type **`Vhrd1`**

$$V(\chi) = \sum_{n=1}^{N} \frac{V_n}{2}\left[1 - \cos\left(n\sigma_V\left(\chi + \varphi_V\right)\right)\right]$$

Type **`Vhrd2`**

$$V(\chi) = V_0 + \sum_{n=1}^{N} V_n \cos\left(n\sigma_V\left(\chi + \varphi_V\right)\right)$$

Type **Vhrd3** $$V(\chi)=V_0+\sum_{n=1}^{N}V_n^c\cos\left(n\sigma_V\left(\chi+\varphi_V\right)\right)+\sum_{n=1}^{N}V_n^s\sin\left(n\sigma_V\left(\chi+\varphi_V\right)\right)$$

where χ is the dihedral angle (radians), σ_V is the symmetry number for the potential energy, φ_V is a phase angle for the potential (radians).

Also for convenience, either the rotational constant or the moment of inertia, which are functions of the dihedral angle, can be entered (all coefficients in units of cm^{-1}). ***It is VERY IMPORTANT that the angles are defined in the same way both for the potential and for the mass factor.***

Type **Bhrd1** (all coefficients in units of cm^{-1})

$$B(\chi)=B_0+\sum_{n=1}^{N}B_n\cos\left(n\sigma_B\left(\chi+\varphi_B\right)\right)$$

Type **Ihrd1** (all coefficients in units of amu.Å^2)

$$I(\chi)=I_0+\sum_{n=1}^{N}I_n\cos\left(n\sigma_I\left(\chi+\varphi_I\right)\right)$$

where χ is the dihedral angle (radians), σ_B and σ_I are symmetry numbers and φ_B and φ_I are phase angles. ***Repeat: It is VERY IMPORTANT that the same phase angle be used both for the potential and for the mass factor: $\varphi_V = \varphi_B$ or φ_I.***

Translation (trn)

DenSum uses an adaptation of the method of Astholz et al.,[29] but applies it to relative translations (standard state corresponding to 1 molecule/cc). According to this method, the number of translational states (G(E)) in a single energy grain (δE) is given by

$$G(E)=F\left[E^{3/2}-\left(E-\delta E\right)^{3/2}\right]$$

where F is a constant that depends on μ, the reduced mass (gram atomic mass units). When classical rotors are convolved with the 3-D translation, the following expression is used:

$$G(E)=F_r\left[E^{(r+3)/2}-\left(E-\delta E\right)^{(r+3)/2}\right]$$

where F_r depends on μ, the number of rotor degrees of freedom (r), and the moments of inertia. For more details, see the Appendix (Theoretical Basis). Note that G(0) = 0, since the nominal energy corresponds to the top energy in each bin.

3.2 Data File Format

Note: Starting with version 2009.0, the data file format is no longer compatible with previous versions.

Line 1

TITLE (CHARACTER*100): up to 180 characters

Line 2

FNAME

FNAME (CHARACTER*10): name of file to be created with ".dens" appended (up to 10 characters). Example: "Hexadiene" produces file named "Hexadiene.dens".

Line 3

N, IWR, VHAR, VROT

N no. of DoF's, IWR (flag)

IWR 0: Uses exact state counts (energy grain = Egrain1)
1: Uses Whitten-Rabinovitch state densities

VHAR KEYWORD for vibrations
'HAR': for vibrational frequencies input as harmonic frequencies.
'OBS': for vibrational frequencies input as 0-1 fundamental frequencies.

VROT KEYWORD for molecular internal and external rotations, except for hindered rotors of type **hrd** (see special instructions below for type hrd)
'AMUA': for moments of inertia input with units of amu Å^2.
'GMCM': for moments of inertia input with units of g cm^2.
'CM-1': for rotational constant input with units of cm^{-1}.
'MHZ': for rotational constant input with MHz.
'GHZ': for rotational constant input with GHz.
(some combinations of upper/lower case are also accepted)

Notes: (a) All of the rotational information in a DenSum data file must be given in the form specified by VROT; (b) VHAR and VROT can be stated in either order on Line 3.

Line 4

Egrain1, Imax1, Isize, Emax2

Egrain1 energy grain in units of cm^{-1}
Imax1 number of array elements in first segment of double array*
Isize total size of double array (number of elements)
Emax2 maximum energy (cm^{-1}) for calculation

* Starting with MultiWell Version 2.08, densum.out provides information about the number of elements (Imax1) needed to achieve fluctuations of less than 5% in the density of states. If Egrain1 and Imax1 are chosen so that fluctuations in the density of states is greater than 5%, a warning is printed to the screen with a suggestion as to what Imax1 should be increased to. Note that a user can choose to disregard this warning and proceed to run MultiWell.

Line 5..N+3

MODE(I), IDOF(I), WE(I), ANH(I), NG(I)

MODE index number for degree of freedom
IDOF KEY WORD for type of degree of freedom
'vib' (vibration)
WE = vibration frequency (cm-1) [see VHAR, line 3]
ANH = vibration anharmonicity (cm-1)
NG = vibration degeneracy
'box' (particle-in-a-box vibration)
WE = vibration frequency parameter (cm^{-1})
ANH = (not used; but a dummy placeholder value must be included)
NG = vibration degeneracy
'rot' (classical rotation)
WE = rotation moment of inertia [units specified by Vrot, Line 3]
ANH = rotation symmetry number
NG = rotation dimension
'qro' (quantized rotation)
WE = rotation moment of inertia [units specified by Vrot, Line 3]
ANH = rotation symmetry number
NG = rotation dimension
'kro' (K-rotor (1-dimensional); quantized rotation)
WE = rotation moment of inertia [units specified by Vrot, Line 3]
ANH = rotation symmetry number
NG = J (quantum number for total angular momentum)
'hra' (1-D *symmetrical* hindered rotor)
WE = vibration frequency (cm^{-1})
ANH = reduced moment of inertia [units specified by Vrot, Line 3]
NG = symmetry of Potential Energy (number of minima per 2π)

[For an unsymmetrical hindered rotor, type 'hrd' is preferred. However, for a symmetrical potential energy and unsymmetrical mass distribution, the present type can be used for approximate results by giving the potential energy symmetry number NG as a negative value (i.e. –NG) and a new line inserted containing NSIG = symmetry number for the mass distribution]

'hrb' (1-D *symmetrical* hindered rotor)

WE = vibration frequency (cm^{-1})

ANH = barrier (cm^{-1})

NG = symmetry of Pot. Energy (number of minima per 2π))

[For an unsymmetrical hindered rotor, type 'hrd' is preferred. However, for a symmetrical potential energy and unsymmetrical mass distribution, the present type can be used for approximate results by giving the potential energy symmetry number NG as a negative value (i.e. –NG) and a new line inserted containing NSIG = symmetry number for the mass distribution]

'hrc' (1-D *symmetrical* hindered rotor)

WE = reduced moment of inertia [units specified by Vrot, Line 3]

ANH = barrier (cm^{-1})

NG = symmetry of Pot. Energy (number of minima per 2π))

[For an unsymmetrical hindered rotor, type 'hrd' is preferred. However, for a symmetrical potential energy and unsymmetrical mass distribution, the present type can be used for approximate results by giving the potential energy symmetry number NG as a negative value (i.e. –NG) and a new line inserted containing NSIG = symmetry number for the mass distribution]

'hrd' (1-D *unsymmetrical* hindered rotor)

Note: care must be taken to treat optical isomers and unsymmetrical hindered internal rotors in a mutually consistent fashion (see Section 9.5).

WE = total number of coefficients for potential energy function.

ANH = total number of coefficients for rotational constant or moment of inertia function.

NG = symmetry number of internal hindered rotor (equal to unity for an asymmetric rotor)

INSERT 2 ADDITIONAL LINES:

LINE1: VTYPE, SYMMV, PHASEV, COEFF1, COEFF2, ...(in order)

VTYPE = "Vhrd1", "Vhrd2", or "Vhrd3"

SYMMV = symmetry number for the potential (not necessarily equal to NG)

PHASEV = phase angle (radians) for potential

COEFF1 = coefficients for potential, in order on the same line (units of cm^{-1})

LINE2: MTYPE, SYMMM, PHASEM, COEFF1, COEFF2, ...(in order)

MTYPE = "Bhrd1" (rotational constant) or "Ihrd1" (moment of inertia)

SYMMM = symmetry number for Bhrd1 or Ihrd1 (not necessarily equal to NG)

PHASEM = phase angle (radians) for Bhrd1 or Ihrd1

COEFF1 = coefficients for Bhrd1 (units of cm^{-1}) or Ihrd1 (amu•$Å^2$ units), in order, on the same line `'trn'` (3-dimensional relative translation)

`WE` = mass of A (amu)
`ANH` = mass of B (amu)
`NG` = ignored, but convenient to set it equal to "3"

Line N+4

BLANK LINE TO INSURE THAT THE LAST LINE IS FOLLOWED BY A CARRIAGE RETURN (needed for all READ statements). THE CARRIAGE RETURN IS EASILY OVERLOOKED.

3.3 DenSum in Batch Mode

In order to run densum with the 'batch' option, one must first prepare a file: densum.batch. This file lists the names of the DenSum data files (".vibs files") to be processed and the energy grain and double array boundaries. The batch file format is given below.

To use this option, the DenSum data files must all reside in one directory (e.g. directory /vibs) and each must be named "<name>.vibs", where <name> does not contain any blank spaces. When listed in densum.batch, each file name must be listed on a separate line, including the extension ".vibs".

densum.batch Batch File Format

Line 1 `Egrain1, Imax1, Isize,  Emax2`

Line 2 `<name1>.vibs`

Line 3 `<name2>.vibs`

Line 4 `<name3>.vibs`

.... `etc.`

`<end of file>`

Sample densum.batch for three .vibs files

```
10. , 400 , 500 , 50000.
B1.vibs
B2.vibs
CH3CO.vibs
```

Execution

a) On linux:

From within the directory in which densum.batch and the .vibs data files reside,

type '<PATH>densum -batch' in a terminal shell, where PATH is the full path to the densum executable.

or

type '`densum -batch`' in a terminal shell, if the PATH variable has been set to include the /bin directory that holds the densum executable.

or

type '`./densum -batch`' in a terminal shell, if a copy of the densum executable resides in the same directory with `densum.batch` and the `.vibs` files.

b) On windows

There are two possibilities:

1) type "`densum -batch`" in a DOS shell

or

2) double-click on "`densum-batch.bat`". Note that `densum.exe` and `densum-batch.bat` must be in the same directory as `densum.batch` and the `.vibs` files.

4. `MomInert`: Moments of Inertia

This code was written mostly by Nicholas F. Ortiz under the direction of John R. Barker. From Cartesian coordinates, it calculates the principal moments of inertia and approximate reduced moment of inertia for an internal rotation.

Example data and output files can be down-loaded from the MultiWell web site: http://aoss.engin.umich.edu/multiwell/.

Note that computer program `lamm`*, described in Chapter 8 is designed to assist in computing moments of inertia as a function of dihedral angle. Thus* `lamm` *is more appropriate than* `MomInert` *for flexible internal rotations.*

4.1 Data File Format

Note: Starting with version 2.0, the data file format is no longer compatible with previous versions.

Line 1

TITLE A line, up to 100 characters, describing the data. This title is reproduced in the output file.

Line 2

UNITS Keyword designating units of coordinate values.

ANGS Angstrom units.
BOHR Atomic (Bohr) units (1 Bohr $\approx$ 0.529 Å)

Line 3

NATOMS Total number of atoms in the molecule (up to 100 atoms).

Line 4
(Repeat for every atom type [see Section 9.7])

ATYPE, IA, X, Y, Z

ATYPE case sensitive atomic symbol, e.g. "C", "H", "Br79" (see Sec. 9.7)
IA Index number of the atom (1 to NATOMS).
X,Y,Z Cartesian coordinates (Å or Bohr; see Line 2) of the atom.

To assign an arbitrary mass to an atom:

1. Designate "XX" for ATYPE (instead of e.g. "H" or "C") in Line 3.
2. On the line following the atom type designation, insert a line giving the mass to be assigned to this particular atom.

This must be done each time a special mass is to be assigned. It is not necessary for the mass to always be the same. In other words, atom type "XX" may be designated multiple times in the same molecule, and a different mass may be specified each time.

Line 5

`IAI, IAJ`

Atom indices for the two atoms defining the axis of internal rotation. If the atom indices are set equal to zero, then internal rotor is not calculated and **Line 6** and **Line 7** can be omitted.

Line 6

`NR`

Number of atoms in one of the two rotating moieties.

Line 7

List of the atom indices that comprise the rotating moiety containing `NR` atoms.

Line 8

For additional internal rotors, REPEAT lines 5-7. TO TERMINATE, enter two zeros: "0 , 0".

Line 9

BLANK LINE TO INSURE THAT THE LAST LINE IS FOLLOWED BY A CARRIAGE RETURN (needed for all READ statements). THE CARRIAGE RETURN IS EASILY OVERLOOKED.

4.2 Computational Approach

This code uses the two axis-defining atoms and one other atom to define two perpendicular planes which intersect at the rotation axis. The distance from all atoms in the molecule to each plane is calculated. The Pythagorean Theorem is then used to define the distance of each atom to the axis. The mass is defined for each atom type, and then the mass and distance are used to calculate the moment of inertia for the internal rotations in amu Å^2.

The center-of-mass is then defined. This is used to calculate all external products and moments of inertia. These values are put into the proper matrix, and eigenvalues are found. The eigenvalues are the principle moments of inertia.

For the reduced moment of inertia for internal rotation, MomInert uses the approximation that the reduced moment is calculated for the axis that contains the twisting bond. This approximation is reasonably accurate (errors of less than 5-10%) for many species.[33]

5. `Thermo`: Thermochemistry

THERMO calculates **equilibrium constants**, **thermodynamic parameters**, and **canonical transition state theory rate constants** (including Eckart tunneling through an unsymmetrical barrier) *via* standard statistical mechanics formulae. It can also compute canonical rate constants using W. H. Miller's **Semiclassical Transition State Theory**.

Example data and output files are given in the examples directory

Note that the data file for Thermo (thermo.dat) is intentionally very similar in format to the data file for DenSum (densum.dat). Thus major parts of the data files can be copied and pasted.

5.1 Thermodynamic Database

A small thermodynamic database is provided in the main MultiWell directory. The chemical species in this database are of interest to our group. We add entries from time to time, depending on our current research interests. The NIST WebBook[34] is a good source for input data, as are the NIST-JANAF Thermochemical Tables.[35]

5.2 Thermodynamic Output

When given a collection of molecular properties, thermo uses statistical mechanics formulae to calculate the corresponding thermodynamics parameters. Thermo can also calculate canonical transition state theory rate constants. Much of the output is obvious, but some items are explained here.

Standard State: the standard state must be selected. The conventional standard state for most tabulations is 1 bar, ideal gas. The numerical value of the equilibrium constant for a reaction (`Kequil`) depends on the selection of the standard state. When comparing forward and reverse reaction rate constants, for example, it is often more convenient to choose the standard state of 1 molecule cm^{-3}.

Molar Enthalpy: The enthalpy for formation at 0 K is required input. Thermo output echoes the input and also reports the enthalpy of formation at 298.15 K and the standard free energy of formation (`DelG(298)`).

Equilibrium constant: `Kequil` is reported for every temperature. In addition, it is reported as a function of temperature: `Kequil = A(T)*exp(B(T)/T)`. The parameters `A(T)` and `B(T)`, which are obtained by finite differences, are in general functions of temperature. The accuracy of these parameters is less than the accuracy of `Kequil` itself. Note that the numerical value of the equilibrium constant depends on the standard state.

Canonical Transition State Theory Rate Constant: "`RATE k(T)`" is reported for every temperature. In addition, it is reported as a function of temperature: `k(T) = A(T)*exp(B(T)/T)`. The parameters `A(T)` and `B(T)`, which are obtained by finite differences, are in general functions of temperature. The accuracy of these parameters is less than the accuracy of k(T) itself. Note that the user should select the standard state of 1 molecule cm^{-3} (i.e. "MCC") when calculating rate constants.

Molar Entropy: the numerical value for the entropy depends on the standard state that is selected, as well as on the energy units selected.

Molar Heat Capacity: Cp depends on energy units selected.

Enthalpy Function: [H(T)-H(0)] depends on the energy units selected.

Accuracy:

All accuracies depend on the accuracy of the input data. For a given set of input data, the accuracies achieved by Thermo are relative to a benchmark based on the same input data. The benchmarks most commonly used are taken from the NIST-JANAF Thermochemical Tables.[35] Note that when rotations are treated classically, the entropy, heat capacity, free energy, and equilibrium constant are less accurate at low temperatures.

• molecular weights are accurate to 0.002 g mol^{-1}, or better.

• enthalpies and Gibbs free energy for individual species generally agree with the JANAF tables to within 0.05 kJ mol^{-1}, or better.

• entropies for individual species are generally accurate to 0.1 J K^{-1} mol^{-1} or better.

• electronic partition function is accurate to 0.1% or better.

• enthalpy, entropy, heat capacity, and Gibbs free energy differences for reaction (DelS(rxn), DelH(rxn), DelCp(rxn) and DelG(rxn), respectively) are generally more accurate than the corresponding quantities for the individual species.

5.3 Functional Forms

(An)Harmonic Vibration (vib):

The energy of an anharmonic oscillator is

$$E = \omega_e\left(v+\frac{1}{2}\right)+\omega_e x_e\left(v+\frac{1}{2}\right)^2 - ZPE$$

where ω_e is the harmonic frequency and the zero point energy ($v = 0$) is

$$ZPE = \omega_e\left(\frac{1}{2}\right)+\omega_e x_e\left(\frac{1}{2}\right)^2 .$$

For a Morse oscillator, the anharmonicity is

$$\omega_e x_e = -\frac{\omega_e^{\,2}}{4D_e}$$

where D_e is the Morse oscillator well depth (from the bottom of the potential well). ***Note the sign convention for anharmonicity used here is not the usual convention.***

By an "Observed frequency" (key word "`obs`") for a vibrational mode, we mean the fundamental frequency, which is the 0-1 transition frequency for the specific mode when all other vibrational quantum numbers equal zero. When $\omega_e x_e \neq 0$ for a separable mode, the fundamental frequency ω^f is related to the harmonic frequency according to the following expression:

$$\omega_e = \omega^f - 2\omega_e x_e$$

For an "observed" frequency, Densum and Thermo compute ω_e from this expression and use it to calculate the vibrational energy levels.

Classical Rotation (rot):

DenSum uses the method of Astholz et al.[29] For expressions, see Robinson and Holbrook, Appendix 5.[30]

See Section 7.4 for a discussion of External Molecular Rotations.

Quantized Rotation (qro):

When $J = 0$, E = 0 and degeneracy (*g)* is unity. When $J > 0$, the degeneracy (*g)* and the energy depend on the dimensionality (*d*) of the rotor [see J. L. McHale, Molecular Spectroscopy (Prentice-Hall, 1999), 216f]:

$d = 1$	$g = 2$	$E = BJ^2, \quad J = 0,1,2,3,...$
$d = 2$	$g = (2J + 1)$	$E = J(J+1)$
$d = 3$	$g = (2J + 1)\,(2J + 1)$	$E = J(J+1)$

See Section 9.4 for a discussion of External Molecular Rotations and the proper input.

Particle in a Box (box):

1-D box of length L with particle of mass m:

$$E = \frac{h^2 n^2}{8mL^2} - \frac{h^2}{8mL^2} \qquad \text{for } n = 1,2,3,...$$

where E is the energy in excess of the "zero" point energy (when n=1).

The effects of higher dimensions are additive. For example, consider a 2-D box of dimensions L_x, L_y:

$$E = \frac{h^2}{8m}\left\{\left[\frac{n_x}{L_x}\right]^2 + \left[\frac{n_y}{L_y}\right]^2 - \left[\frac{1}{L_x}\right]^2 + \left[\frac{1}{L_y}\right]^2\right\} \qquad \text{for } n_i = 1,2,3,...$$

Because the energies in multiple dimensions are additive, DenSum and Thermo consider them separably. The "frequency parameter" that is needed by DenSum and Thermo: $\frac{h^2}{8mL^2}$. Furthermore, DenSum and Thermo assume $L_x = L_y = ... = L$.

1-D Hindered Rotation, Symmetrical (hra, hrb, hrc)

These types are intended for use with symmetrical internal rotors (e.g. a CH_3 rotor). Relationships among parameters for a hindered rotor:

$$B / cm^{-1} = \frac{16.85763}{I_r / amu\ \mathring{A}^2}$$

$$V = \frac{1}{B}\left[\frac{\omega}{n}\right]^2$$

where B is the rotational constant, I_r is the reduced moment of inertia, V is the barrier to internal rotation, and ω is the small amplitude harmonic frequency (cm^{-1}). For convenience, three methods are provided input of any combination of two independent parameters from the set: V, ω, I_r.

1-D Hindered Rotation, "Unsymmetrical" (hrd)

This type is intended for unsymmetrical rotors (e.g. the CH_2Cl rotor) and flexible internal rotors. For this type, one must provide the torsional potential energy and reduced moment of inertia (or rotational constant) as functions of the dihedral angle χ (radians). For a discussion, see the Technical Notes.

Note that computer program `lamm`*, described in Chapter 8 is designed to assist in computing moments of inertia as a function of dihedral angle. Thus* `lamm` *is more appropriate than* `MomInert` *for flexible internal rotations.*

For convenience, three forms of the torsional potential energy are accepted (all coefficients in units of cm^{-1}):

Type **`Vhrd1`** $$V(\chi) = \sum_{n=1}^{N} \frac{V_n}{2}\left[1 - \cos\left(n\sigma_V\left(\chi + \varphi_V\right)\right)\right]$$

Type **`Vhrd2`** $$V(\chi) = V_0 + \sum_{n=1}^{N} V_n \cos\left(n\sigma_V\left(\chi + \varphi_V\right)\right)$$

Type **`Vhrd3`** $$V(\chi)=V_0+\sum_{n=1}^{N}V_n^c\cos\left(n\sigma_V\left(\chi+\varphi_V\right)\right)+\sum_{n=1}^{N}V_n^s\sin\left(n\sigma_V\left(\chi+\varphi_V\right)\right)$$

where χ is the dihedral angle (radians), σ_V is the symmetry number for the potential energy, φ_V is a phase angle for the potential (radians).

Also for convenience, either the rotational constant or the moment of inertia, which are functions of the dihedral angle, can be entered (all coefficients in units of cm^{-1}). ***It is VERY IMPORTANT that the angles are defined in the same way both for the potential and for the mass factor.***

Type **`Bhrd1`** (all coefficients in units of cm^{-1})

$$B(\chi)=B_0+\sum_{n=1}^{N}B_n\cos\left(n\sigma_B\left(\chi+\varphi_B\right)\right)$$

Type **`Ihrd1`** (all coefficients in units of $amu \bullet \AA^2$)

$$I(\chi)=I_0+\sum_{n=1}^{N}I_n\cos\left(n\sigma_I\left(\chi+\varphi_I\right)\right)$$

where χ is the dihedral angle (radians), σ_B and σ_I are symmetry numbers and φ_B and φ_I are phase angles (radians). ***Repeat: It is VERY IMPORTANT that the same phase angle be used both for the potential and for the mass factor: $\varphi_V = \varphi_B$ or φ_I.***

Hindered Gorin Model (gor) and Fitting to Experimental Rate Constants (fit)

Thermo includes the capability to automatically find the hindrance parameters for the Hindered Gorin transition state.[36-43] One can choose one or both of the following types (`IDOF`) of degrees of freedom.

1) `IDOF` = `gor` selected for one vibrational stretching mode. For a selected potential energy function, Thermo finds the center of mass distance r_{max} corresponding to maximum of $V_{effective}$ at temperature T, where the rotational energy in the 2-D pseudo-diatomic rotation is assumed to be RT. From the value of r_{max}, Thermo computes the 2-D moment of inertia.

2) `IDOF` = `fit` selected for two (linear molecule) or three (non-linear) rotational dimensions. Thermo finds the hindrance parameters γ (`gamma`) and η (`eta`) that produce a good fit at each temperature to experimental rate constants (one for each temperature) that are entered.

$$\gamma=(1-\eta)^{1/2}$$

If both IDOF = `gor` and IDOF = `fit` are selected, Thermo finds the maximum of $V_{effective}$ and uses it to find the hindrance parameters (`gamma` and `eta`) that produce a good fit to experimental rate constants (one for each of the `Nt` temperatures) that are entered.

The selectable potential energy functions are:

`MORSE` *(Morse Oscillator)*

$$V_{Morse}(r) = D_e\left\{1-\exp\left[-\beta_{Morse}\left(r-r_e\right)\right]\right\}^2 - D_e$$

$$\beta_{Morse} = 2\pi\nu_e\sqrt{\frac{\mu}{2D_e}}$$

`VARSHNI` *(Varshni Oscillator[44])*

$$V_{Varshni}(r) = D_e\left\{1-\left(\frac{r_e}{r}\right)\exp\left[-\beta_{Varshni}\left(r^2-r_e^{\,2}\right)\right]\right\}^2 - D_e$$

$$\beta_{Varshni} = \frac{1}{2r_e}\left[2\pi\nu\sqrt{\frac{\mu}{2D_e}}-\frac{1}{r_e}\right] = \frac{1}{2r_e}\left[\beta_{Morse}-\frac{1}{r_e}\right]$$

`sMORSE` *("Stiff Morse" Oscillator)*

This is a Morse oscillator, but replaces β_{Morse} with $\beta_{Stiff} = \beta_{Morse}\left[1+c_s\left(r-r_e\right)^2\right]$; c_s is an empirical parameter and must be ≥0.

The input is assumed to correspond to the following generic recombination reaction:

$$A + B \rightarrow C^{\ddagger}$$

In the thermo input, A and B are labeled as reactants (`"reac"`) and $C^{\ddagger}$ is labeled as a transition state (`"ctst"`). For a barrier-less recombination reaction, the enthalpy of formation at 0 K (`DelH`) for $C^{\ddagger}$ must be set equal to the sum of the enthalpies of formation for the two reactants. All of the internal and rotational degrees of freedom for A and B are entered as usual. The *internal* degrees of freedom for $C^{\ddagger}$ consist of the *combined* degrees of freedom (*internal* and *external*) of A and B, *plus* the external rotations for $C^{\ddagger}$. Thus the number of degrees of freedom (including degeneracies and multiple dimensions) of $C^{\ddagger}$ is

$$n_C = n_A + n_B + n_C^{rots}$$

where n_A and n_B are the total number of degrees of freedom for A and B, and n_C^{rots} is the number of external rotations of $C^{\ddagger}$.

In order to find the center of mass distance r_{max} corresponding to maximum of $V_{effective}$ at temperature T, the external 2-D rotation for $C^{\ddagger}$ is labeled `"gor"` (see below). This signals that this particular entry is a dummy and will be adjusted in the process of finding the maximum $V_{effective}$. The potential energy function is specified later in the data file (see below).

In order to adjust the "hindrance" to fit specified rate constants, some of the rotations in $C^{\ddagger}$ that were originally associated with A and B must be multiplied by γ (see above). These are labeled "`fit`" (see below). These rotations do *not* include internal rotation about the new bond in $C^{\ddagger}$, but only the rotations about internal axes perpendicular to the new bond. Later in the data file, the (experimental) rate constants to be fitted are entered... one for each temperature.

Set of Fully Coupled Anharmonic Vibrations (qvb)

This type specifies that program `Thermo` must read a supplementary data file named `<name>.qvb`, which must be generated previously by program `adensum` (Section 7.2). This supplementary file must be present in the same directory as data file `thermo.dat`. See Section 7.2 for more details. By using this supplementary data file, Thermo can utilize full intermode vibrational anharmonic coupling when computing thermodynamic data. This coupling is most important at high temperatures. This option is used when computing thermal rate constants with semi-classical transition state theory (SCTST; see Sections 7.4 and 9.9 for more details).

Thermal Rate Constants from SCTST (crp)

This type specifies that program `Thermo` reads a supplementary data file named `<name>.qcrp`, which must be generated previously by program `sctst` (Section 7.4) File `<name>.qcrp` must be present in the same directory as data file `thermo.dat`. See Sections 7.4 and 9.9 for more details. By using this supplementary data file, `Thermo` can utilize the semi-classical transition state theory (SCTST) of W. H. Miller and coworkers. SCTST accounts for fully coupled degrees of freedom, including the reaction coordinate. As a result, the theory incorporates multidimensional quantum tunneling along the curved reaction path, which is very important at low temperatures. The degree of freedom type "`qvb`" is usually also used when computing SCTST rate constants (see the preceding paragraph) in order to obtain good accuracy at both low and high temperatures.

5.4 Data File Format

Note: Starting with version 2009.0, the data file format is no longer compatible with previous versions.

<u>**Line 1**</u>

`EKEY , SSKEY`

`EKEY` Key Word for Energy units: "`KCAL`" or "`KJOU`" (upper case characters). Note that "`CM-1`" is a reserved keyword to be fully implemented later.

`SSKEY` Key Word for Standard State: "`BAR`", "`ATM`", or "`MCC`" (molecule/cc)

<u>**Line 2**</u>

`Nt` number of temperatures

<u>**Line 3**</u>

`T1`, `T2`, `T3`,... List of `Nt` temperatures

<u>**Line 4**</u>

`Ns` number of chemical species

<u>**Line 5**</u> for reactants, products, or "none"; *see below for transition states*

`REPROD`, `MOLNAME`, `DelH`

`REPROD` key word:
"`reac`" = reactant
"`prod`" = product
"`none`" = not included in equilibrium constant

`MOLNAME` Name of chemical species (up to 10 characters)

`DelH` enthalpy of formation at 0 K

<u>**Line 5'**</u> for ***transition states***

`REPROD`, `MOLNAME`, `DelH`, `vimag`, `VVR`

`REPROD` = "`ctst`"

For transition states, only. Assumes 1-D quantum mechanical tunneling via an unsymmetrical Eckart energy barrier. *Tunneling is neglected if `vimag` = 0.0 or `VVR` = 0.0.*

This choice calculates the canonical transition state theory rate constant based on reactant and transition state parameters. It assumes the reaction consists of reactants (each labeled "reac") proceeding to the transition state (labeled "ctst"). Note that for bimolecular and termolecular reactions, the standard state should be set to "MCC" for concentration units.

MOLNAME Name of transition state (up to 10 characters)

DelH enthalpy of formation at 0 K. (Note: when using the SCTST (see section 7.4), DelH is the enthalpy of formation of the transition state at 0 K.)

vimag magnitude of imaginary frequency (units of cm^{-1}). When using the SCTST (see section 7.4), vimag is not read, but must be entered. *Tunneling is neglected if vimag = 0.0.*

VVR Height of barrier for the reverse reaction (energy units specified in Line 1). When using the SCTST (see section 7.4), VVR is not read, but must be entered. *Tunneling is neglected if VVR = 0.0.*

Line 6

FORMULA Empirical formula written in the usual way with atomic symbols. These symbols include "D" for deuterium and "T" for tritium. Special atomic symbols used for other isotopic species (e.g. "Cl35") are summarized in Section 9.7. Whenever an atomic symbol includes numeric characters (e.g. "Cl35"), it must be enclosed in square brackets (e.g. "[Cl35]"). Examples of empirical formulas: C2H6O, CH3CH2OH, (CH3)2OH, (CH3)2(N)O2H, CH[Br79]2(CH3)3, CH3OD, C([C13]H3)4.

Lines 7-9

Three comment lines

Line 10

Sym, Sopt, Nele

Sym	external symmetry number (INTEGER)
Sopt	number of optical isomers (INTEGER)
Nele	number of electronic energy levels (INTEGER)

Line 11

Elev, gele (Repeat line for Nele electronic states)

Elev electronic level energies (REAL).
NOTE: lowest level should be at energy Elev = 0.0.

gele electronic level degeneracies (INTEGER)

Line 12

N , **VHAR** , **VROT** number of vibrations and rotations to be read in

VHAR KEYWORD for vibrations
'HAR': for vibrational frequencies input as harmonic frequencies.
'OBS': for vibrational frequencies input as 0-1 fundamental frequencies.

VROT KEYWORD for molecular internal and external rotations, except for hindered rotors of type **hrd** (see special instructions below for type hrd)
'AMUA': for moments of inertia input with units of amu Å^2.
'GMCM': for moments of inertia input with units of g cm^2.
'CM-1': for rotational constant input with units of cm^{-1}.
'MHZ': for rotational constant input with MHz.
'GHZ': for rotational constant input with GHz.
(some combinations of upper/lower case are also accepted)

Note: All of the rotational information for a given molecular species must be given in the form specified by VROT for that species.

Line 13

(repeat N times for the N vibrations and rotations)

MODE(I), IDOF(I), WE(I), ANH(I), NG(I)

MODE index number for degree of freedom

IDOF KEY WORD for type of degree of freedom

'vib' (vibration)
WE = vibration frequency (cm-1) [see VHAR, line 3]]
ANH = vibration anharmonicity (cm-1)
NG = vibration degeneracy

'box' (particle-in-a-box vibration)
WE = vibration frequency parameter (cm^{-1})
ANH = (not used; but a dummy placeholder value must be included)
NG = vibration degeneracy

'rot' (classical rotation)
WE = rotation moment of inertia [units specified by Vrot, Line 3]
ANH = rotation symmetry number
NG = rotation dimension

'qro' (quantized rotation)
WE = rotation moment of inertia [units specified by Vrot, Line 3]
ANH = rotation symmetry number
NG = rotation dimension

'kro' (K-rotor (1-dimensional); quantized rotation)
WE = rotation moment of inertia [units specified by Vrot, Line 3]
ANH = rotation symmetry number
NG = J (quantum number for total angular momentum)

'hra' (1-D *symmetrical* hindered rotor)

WE = vibration frequency (cm^{-1})

ANH = reduced moment of inertia [units specified by Vrot, Line 3]

NG = symmetry of Potential Energy (number of minima per 2π)

[For an unsymmetrical hindered rotor, type 'hrd' is preferred. However, for a symmetrical potential energy and unsymmetrical mass distribution, the present type can be used for approximate results by giving the potential energy symmetry number NG as a negative value (i.e. –NG) and a new line inserted containing NSIG = symmetry number for the mass distribution]

'hrb' (1-D *symmetrical* hindered rotor)

WE = vibration frequency (cm^{-1})

ANH = barrier (cm^{-1})

NG = symmetry of Pot. Energy (number of minima per 2π))

[For an unsymmetrical hindered rotor, type 'hrd' is preferred. However, for a symmetrical potential energy and unsymmetrical mass distribution, the present type can be used for approximate results by giving the potential energy symmetry number NG as a negative value (i.e. –NG) and a new line inserted containing NSIG = symmetry number for the mass distribution]

'hrc' (1-D *symmetrical* hindered rotor)

WE = reduced moment of inertia [units specified by Vrot, Line 3]

ANH = barrier (cm^{-1})

NG = symmetry of Pot. Energy (number of minima per 2π))

[For an unsymmetrical hindered rotor, type 'hrd' is preferred. However, for a symmetrical potential energy and unsymmetrical mass distribution, the present type can be used for approximate results by giving the potential energy symmetry number NG as a negative value (i.e. –NG) and a new line inserted containing NSIG = symmetry number for the mass distribution]

'hrd' (1-D *unsymmetrical* hindered rotor)

Note: care must be taken to treat optical isomers and unsymmetrical hindered internal rotors in a mutually consistent fashion (see Section 7.5).

WE = total number of coefficients for potential energy function.

ANH = total number of coefficients for rotational constant or moment of inertia function.

NG = symmetry number of internal hindered rotor (equal to unity for an asymmetric rotor)

INSERT 2 ADDITIONAL LINES:

LINE1: VTYPE, SYMMV, PHASEV, COEFF1, COEFF2, ...(in order)

VTYPE = "Vhrd1", "Vhrd2", or "Vhrd3"

SYMMV = symmetry number for the potential

PHASEV = phase angle (radians) for potential

COEFF1 = coefficients for potential, in order on the same line (units of cm^{-1})

LINE2: MTYPE, SYMMM, PHASEV, COEFF1, COEFF2, ...(in order)

MTYPE = "Bhrd1" for rotational constant or "Ihrd1" for moment of inertia

SYMMM = symmetry number for Bhrd1 or Ihrd1

`PHASEM` = phase angle (radians) for `Bhrd1` or `Ihrd1`
`COEFF1` = coefficients for `Bhrd1` (units of cm^{-1}) or `Ihrd1` (amu•Å^2 units), in order on the same line

`'fit'` (change the hindrance parameter of the Hindered Gorin Model to fit an experimental rate constants listed on **LINE 14**)
`WE` = rotational moment of inertia (amu Å^2) for the internal rotor
`ANH` = rotational symmetry number
`NG` = rotational dimension

`'gor'` (Gorin Model. **Requires LINES 15-19**)
`WE` = 2D adiabatic external rotor (amu Å^2); not used but always read.
`ANH` = rotational symmetry number
`NG` = 2; rotational dimension

`'qvb'` (read supplementary data file to account for fully-coupled vibrational anharmonicity)
`WE` = dummy place-holder; not used but always read.
`ANH` = dummy place-holder; not used but always read.
`NG` = dummy place-holder; not used but always read.

`'crp'` (read supplementary data file to compute thermal rate constant using SCTST). NOTE: when using this type, `vimag` and `VVR` on Line 5' are not read, but must be entered.
`WE` = dummy place-holder; not used but always read.
`ANH` = dummy place-holder; not used but always read.
`NG` = dummy place-holder; not used but always read.

RETURN TO **LINE #5** and REPEAT FOR THE NEXT CHEMICAL SPECIES

AFTER ALL CHEMICAL SPECIES HAVE BEEN ENTERED, RESUME HERE:

Line 14 (required if `'fit'` is used)
`kexp(1)`, `kexp(2)`, ... list of `Nt` experimental recombination rate constants (units of cm^3 molecule^{-1} s^{-1}): one for each temperature.

Line 15-19 (required if `'gor'` is used)

Line 15 `readpot` Keyword `'MORSE'`, `'VARSHNI'`, or `'sMORSE'` (Stiff-Morse). This is the type of bonding potential energy function.
Line 16 `freq` Harmonic frequency (cm^{-1}) of the normal mode associated with the forming bond.
Line 17 `De` Dissociation Energy De (units of `Eunit`) of the forming bond.
Line 18 `re` EQUILIBRIUM center-of-mass distance (Å) between the of the two fragments that are forming the new bond.
Line 19 `c` Stiff-Morse parameter (needed only for `'sMORSE'` type of potential).

Line 20 (BLANK LINE)

6. `gauss2multi`: A Tool for Creating Data Files

In many cases, one wants to create data files from calculated properties of molecules and transition states. The program **`gauss2multi`** is a tool designed to assist in this process. The source code is located in the `multiwell/src` directory and the executable is located in the `multiwell/bin` directory. The tool reads vibrational frequencies, coordinates, and energies from the standard `.log` or `.out` output files produced by GAUSSIAN 98[45] and GAUSSIAN 03.[46]

In order to convert GAUSSIAN output files into multiwell input files you need:
1) GAUSSIAN output files with frequency calculation (file extension: `.log` or `.out`)
2) `gauss2multi.cfg` configuration file, which must reside in the same directory as the gaussian output files. An example is found in the `examples` directory

The following files will be created for each GAUSSIAN file that is read:

FILE NAME	DESCRIPTION (see MultiWell User Manual)
`name.coords`	mominert input file
`name.coords.out`	mominert output file
`name.vibs`	densum input file
`name.dens`	densum output file
`name.therm`	thermo input file
`name.therm.out`	thermo output file
`multiwell.dat`	rough drat of multiwell input file

("name" refers to the GAUSSIAN file name, which cannot exceed 10 characters in length)

You can create and edit the configuration file by yourself or use the step-by-step script to create it.

The `gauss2multi.cfg` configuration file contains the following items (an example file is given below):

Line:
1 - Energy units: `KCAL` , `KJOU`, or `CM-1` (use upper case characters)
2 - number of temperatures
3 - list of temperatures separated by one or more blank spaces
4 - Pressure units: `BAR`, `TOR`, `ATM`, or `MCC` (use upper case characters)
5 - number of pressures
6 - list of pressures separated by one or more blank spaces
7 - `Egrain`, `imax1`, `Isize`, `Emax2`
8 - index number (ascending order), name of gaussian output file (`.log` or `.out`), type of structure: `WELL`, `PROD`, or `TS` (use upper case characters)
Repeat Line-8 for additional species.

TO RUN `gauss2multi`

There are two modes of operation:

(a) To set up files for an entire multi-species model.
In this case, the energies (i.e. ΔH_f^0(0 K)) for all of the species will be calculated with respect to the energy of the first well (index number = 1) listed in `gauss2multi.cfg`.

In the same directory as the GAUSSIAN files, type '`gauss2multi`'

or

(b) To generate data files for a single species.
In the same directory as the GAUSSIAN files, type '`gauss2multi <GAUSSIAN _Name_File>`'

IMPORTANT:

The conversion to MultiWell files cannot be completely automated: all of these files may require manual changes! Be careful!

EXAMPLE CONFIGURATION FILE (`gauss2multi.cfg`)

```
KCAL
3
298.15  398  498
TOR
2
760  1200
10.    400     500     50000.
1  B2.log WELL
2  B_Intermed.log  WELL
3  TS-B-B2.log     TS
4  C_product2.log  PROD
5  Product-3.log   PROD
```

7. doloops, ansum, adensum, and sctst: *non-Separable* Vibrations

For a brief introduction to non-separable anharmonic vibrations, see Technical Note #2.

The first three programs, which are for computing sums and densities of states for non-separable vibrations, are described elsewhere.[6,47] All of them include the ability to include quantum and classical rotations in the sums and densities of states. Programs `doloops` and `adensum` also can include hindered internal rotations. Each program is particularly suited for specific applications. In particular, the number of atoms or vibrational modes strongly affects execution times. Note that "exact" refers to exact at the level of second order vibrational perturbation theory (VPT2).

	~Atoms	~Vibrations	Accuracy	Method
`doloops`	3-4	3-6	exact	direct count using nested DO-loops[6]
`adensum`	4- >50	6- >150	approximate	Wang-Landau algorithm[6 ,48,49]
`ansum`	3-6	3-12	approximate	Monte Carlo algorithm[47]

The third program, `sctst`, is for computing rate constants by suing the semi-classical transition state theory.

NOTE. In second order vibrational perturbation theory (VPT2), the zero point energy includes contributions from the harmonic frequencies, the anharmonicities, and an additional term denoted ε_0 (see Eq. 9 in Section 9.9).[50,51] This term, which cancels-out in spectroscopy applications but must be included for thermochemistry and kinetics, is computed by electronic structure programs that employ VPT2 (e.g. GUSSIAN and CFOUR), but it is not computed by MultiWell. For *ab initio* kinetics calculations involving small molecules the term may be significant, but it is very small for molecules with small rotational constants. For high accuracy *ab initio* kinetics calculations (e.g. [7], E_0 can be included manually by modifying the reaction critical energy.

7.1 Program doloops

(For definitions of terms, see Technical Note #2.)

This program computes exact counts of bound and quasi-bound states by using straightforward nested DO-loops, which step through all allowed quantum numbers and compute E_v for all states up to a high energy. The state energies are binned in energy grains specified by user (typically 10 cm^{-1}). Sums and densities of states can then be computed using the combined Beyer-Swinhart[11] and Stein-Rabinovitch[10] algorithms (the BSSR algorithm). The number of nested DO-loops is equal to a number of degrees of freedom (DOF). The algorithm is very efficient for small molecules having 3-4 atoms, but becomes extremely slow for larger species.[6]

Running doloops

The executable (binary) program resides in the directory `multiwell<version>/bin`, where `<version>` is the version number (e.g. "2010.0") To execute the compiled program, enter the directory that contains the `doloops` data file ("`doloops.dat`") and then type `~/multiwell<version>/bin/ doloops`. Directory `multiwell<version>` is assumed to reside in your Home directory. The output file `doloops.out` is automatically placed in the directory that contains `doloops.dat`. A second output file `<name>.dens`, which is designed to be an input file for the MultiWell master equation code, is also generated, where `<name>` is the name of the chemical species (specified in `doloops.dat`).

Data File Format

(For definitions of terms, see Technical Note #2.)

Line 1: `Name` (species name; ≤10 characters)

Line 2: `Title 1` (≤100 characters)

Line 3: `Title 2` (≤100 characters)

Line 4: `ns` (number of vibrational modes), `NY` (number of **Y** matrix elements), `NZ` (number of **Z** matrix elements), `WW` (a keyword to designate frequencies as ω_e or ω^0)

`WW` = `We` for ω

`WW` = `Wf` for fundamental vibrations (0-1 transitions)

`WW` = `W0` for ω^0 [note: `0` is a zero]

(The distinction between ω and ω^0 is explained by eqs. (II,267) through (II,269) of Herzberg [52]. The computer code carries out the conversion between them.)

Lines 5+ (5 to 4+`ns`): `wa(i)` [i = 1 , `ns`] vibrational frequencies (cm^{-1})

Line 6 (5+`ns`): `KEYWORD` for reading the **X** anharmonicity matrix:

= `lower` (for reading lower half of matrix, plus the diagonal)

= `upper` (for reading upper half of matrix, plus the diagonal)

Lines 7+ (6+`ns` to 5+2`ns`): `xa(k,j)` anharmonicities (cm^{-1}) in upper or lower halves (plus the diagonal) of the `ns`×`ns` anharmonicity matrix.

Line 8 (6+2`ns`): `Nsep` (number of separable modes to be convoluted with the `ns` non-separable modes)

Line 9 (7+2`ns`): input data for the separable modes (quantum rotors (1-D or 2-D), and/or particle in a box). Use the same input format as in `DenSum` (see the MultiWell User manual)

Line 10 (8+2`ns`): `Egrain1, imax1, Isize, Emax2, KEYWORD`
`Egrain1` = energy grain (cm^{-1})
`imax1` = number of array elements in lower part of the Double Array (see MultiWell User Manual)
`Isize` = total number elements in the Double Array.
`Emax2` = upper energy limit

Notes

1. Energies are relative to the zero point energy.

NOTE. In second order vibrational perturbation theory (VPT2), the zero point energy includes contributions from the harmonic frequencies, the anharmonicities, and an additional term denoted ε_0 (see Eq. 9 in Section 9.9).[50,51] This term, which cancels-out in spectroscopy applications but must be included for thermochemistry and kinetics, is computed by electronic structure programs that employ VPT2 (e.g. GUSSIAN and CFOUR), but it is not computed by MultiWell. For *ab initio* kinetics calculations involving small molecules the term may be significant, but it is very small for molecules with small rotational constants. For high accuracy *ab initio* kinetics calculations (e.g. [7], E_0 can be included manually by modifying the reaction critical energy.

7.2 Program adensum

(For definitions of terms, see Technical Note #2.)

Wang and Landau[48,53] pioneered an efficient random walk algorithm in energy space to compute densities of states for use in classical statistical models. Their approach is based on the idea that a density-weighted Monte Carlo sampling of the energy states of the system will produce a flat histogram of samples in equally-spaced energy bins if the density of states function is exact, in the limit of an infinite number of trials. The algorithm is initiated with a trial density of states in every energy bin and the density-weighted Monte Carlo sampling begins. A histogram records every visit to an energy bin. The algorithm is designed to modify the relative density of states $G(E_i)$ on every visit to an energy bin (at energy E_i) in such a way so that $G(E_i)$ approaches exact proportionality with the true density of states after a large number of samples.[48,53] The algorithm can be halted by monitoring the "flatness" of the histogram, or by limiting the number of Monte Carlo samples. Basire et al.[49] adapted the Wang-Landau algorithm for computing quantum densities of states for fully coupled anharmonic systems using the perturbation theory expansion for vibrational energy. It is a powerful method. For a full description of the algorithm and notation, see the paper by Basire et al. [49] As one of their demonstrations, they computed the density of states for naphthalene, based on the fully-coupled **X** matrix reported by Cané et al.[54]

We have modified the Basire et al. version[49] of the Wang-Landau algorithm in several minor ways to meet the needs for the density of states in chemical kinetics and master equation applications. In our approach, we compute statistics for the histogram based only on energy bins that contain states. We do not use the "flatness" criterion for halting the algorithm, but instead use the number of Monte Carlo trials per energy bin, since that is the principal factor that controls the relative error (see below). We also apply the tests described above to ensure that selected states are bound or quasi-bound.

We have automated the algorithm by choosing the probability of accepting a move (parameter p; see Basire et al. for notation) according to N, the number of DOF. As discussed by Basire et al., if p is too large, the Monte Carlo selection can sample the entire energy range rapidly, but there may be many wasted samples that fall out of range. On the other hand if p is too small, the entire energy range may not be sampled efficiently, and more trials will be needed. In this work, we adopt p =MIN(1/N, 0.25). This protocol is consistent with the values for p used by Basire et al.[49] As discussed in a later section, our results show little sensitivity to the selection of p.

The most important new modification we have made to the Basire et al. version[49] of the Wang-landau algorithm is to introduce automatic normalization of the density of states. Recall that the Wang-Landau algorithm produces results that are *proportional* to the exact density of states. The un-normalized $G(E_i)$ may be sufficient for many applications, but the absolute density of states $\rho(E_i)$ is needed for calculating rate constants from statistical rate theories, for example. In our approach, the size of the energy bins (i.e. the energy grain size ΔE) is chosen small enough so that only one state, the lowest energy state (i.e. the zero point energy state), falls in the lowest energy bin. Since we know that only one state resides in the lowest energy bin, the density of states in lowest energy bin must be $(\Delta E)^{-1}$, we then obtain the absolute density of states:

$$\rho(E_i) = \frac{G(E_i)}{G(E_i = 0)} \frac{1}{\Delta E}$$

For vibrational states, it is easy to choose ΔE small enough so that only the zero point energy state falls within the lowest energy bin. For master equation and chemical kinetics applications, the typical energy grain is $\Delta E \leq 10$ cm^{-1} and vibrational frequencies are typically much larger. If rotations were to be included with the vibrations, normalization can still be achieved easily by using one of the other methods described above to count the number of rotational states in the lowest energy grain and normalizing accordingly.

Our modified version of the Wang-Landau algorithm[48,53] draws heavily on that of Basire et al., [49] and is summarized elsewhere.[6]

NOTE. In second order vibrational perturbation theory (VPT2), the zero point energy includes contributions from the harmonic frequencies, the anharmonicities, and an additional term denoted ε_0 (see Eq. 9 in Section 9.9).[50,51] This term, which cancels-out in spectroscopy applications but must be included for thermochemistry and kinetics, is computed by electronic structure programs that employ VPT2 (e.g. GUSSIAN and CFOUR), but it is not computed by MultiWell. For *ab initio* kinetics calculations involving small molecules the term may be significant, but it is very small for molecules with small rotational constants. For high accuracy *ab initio* kinetics calculations (e.g. [7], E_0 can be included manually by modifying the reaction critical energy.

Running adensum

The executable (binary) program resides in the directory `multiwell<version>/bin`, where `<version>` is the version number (e.g. "`2010.0`") To execute the compiled program, enter the directory that contains the `adensum` data file ("`adensum.dat`") and then type `~/multiwell<version>/bin/ adensum`. Directory `multiwell<version>` is assumed to reside in your Home directory. The output file `adensum.out` is automatically placed in the directory that contains `adensum.dat`. A second output file `<name>.dens`, which is designed to be an input file for the MultiWell master equation code, is also generated, where `<name>` is the name of the chemical species (specified in `adensum.dat`).

`Adensum` has been designed to run using a "checkpoint file", which is generated after the density of states of the fully coupled vibrations have been calculated, which is the most time-consuming section. The following section convolutes the separable degrees of freedom, if any are specified. The separable degrees of freedom may include free or hindered rotors, harmonic vibrations, etc. In particular, the K-rotor and possibly other external rotational degrees of freedom may be specified. If desired, the densities of states may be recalculated using various combinations of separable degrees of freedom, but retaining the same set of fully coupled vibrations. Repetitive calculations like these can be restarted using the checkpoint file, which saves considerable computer time.

Three output files are generated by adensum: `adensum.out`, `<name>.dens`, and `<name>.qvb`. The first file (`adensum.out`) is a standard output file, which summarizes all input and output. The second file (`<name>.dens`), which takes a chemical species name as part of the file name, gives sums and densities of states, formatted as double arrays; it can be used as input for the MultiWell master equation code. By using this input file, master equation calculations can be carried out with densities of states computed by including the fully-coupled vibrations. The third file (`<name>.qvb`) is formatted as a supplementary input file for program `Thermo`; it tabulates the partition function and other thermodynamic properties of the coupled vibrational modes at a set of fixed temperatures from 50 K to >3000 K. `Thermo` can utilize these partition functions to compute and thermodynamics data and canonical rate constants. See Section 5.3 of this User Manual for a description of how the `<name>.qvb` file is used by Thermo.

Data File Format

(For definitions of terms, see Technical Note #9.)

Line 1: `Name` (species name; ≤10 characters)

Line 2: `Title 1` (≤100 characters)

Line 3: `Title 2` (≤100 characters)

Line 4: `ns` (number of vibrational modes), `NY` (number of **Y** matrix elements), `NZ` (number of **Z** matrix elements), `WW` (a keyword to designate frequencies as ω_e or ω^0)

`WW` = `We` for ω

`WW` = `Wf` for fundamental vibrations (0-1 transitions)

`WW` = `W0` for ω^0 [note: `0` is a zero]

(The distinction between ω and ω^0 is explained by eqs. (II,267) through (II,269) of Herzberg [52]. The computer code carries out the conversion between them.)

Lines 5+ (5 to 4+`ns`): `wa(i)` [i = 1 , `ns`] vibrational frequencies (cm^{-1})

Line 6 (5+`ns`): `KEYWORD` for reading the **X** anharmonicity matrix:

= `lower` (for reading lower half of matrix, plus the diagonal)

= `upper` (for reading upper half of matrix, plus the diagonal)

Lines 7+ (6+`ns` to 5+2`ns`): `xa(k,j)` anharmonicities (cm^{-1}) in upper or lower halves (plus the diagonal) of the `ns`×`ns` anharmonicity matrix.

Line 8: `Nsep`, `Vrot`

`Nsep` = number of separable modes to be convoluted with the `ns` non-separable modes)

`Vrot` = KEYWORD for specifying units for internal and external rotations, except for hindered rotors of type **`hrd`** (see special instructions in the chapter on DenSum for type `hrd`)

`'AMUA'`: for moments of inertia input with units of amu Å^2.

`'GMCM'`: for moments of inertia input with units of g cm^2.

`'CM-1'`: for rotational constant input with units of cm^{-1}.

`'MHZ'`: for rotational constant input with MHz.

`'GHZ'`: for rotational constant input with GHz.
(some combinations of upper/lower case are also accepted)

Line 9: input data for the separable modes (classical or quantum free rotors, hindered rotors of several types, and Morse oscillators). Use the same input format for these degrees of freedom as used in `DenSum` (see the MultiWell User Manual)

Line 10: `Egrain1, imax1, Isize, Emax2, KEYWORD`

`Egrain1` = energy grain (cm^{-1})
`imax1` = number of array elements in lower part of the Double Array (see MultiWell User Manual)
`Isize` = total number elements in the Double Array.
`Emax2` = upper energy limit
`KEYWORD` = controls number of stochastic trials:
`Fair:` low statistical accuracy (10^2 trials per energy grain)
`Good:` good statistical accuracy (10^3 trials per energy grain).
`Better:` better statistical accuracy (10^4 trials per energy grain)
`Best:` best statistical accuracy (10^5 trials per energy grain)
`Extra:` best statistical accuracy (10^6 trials per energy grain, but no more than 2×10^9 total trials for all energy grains)

Line 11: `chekpoint`

If `chekpoint = "chekstart"`, then the new calculation starts by reading the pre-existing checkpoint file. Otherwise, the full calculation if is carried out. Line 10 must be identical to the data file used to create the pre-existing checkpoint file.

Notes

1. Energies are relative to the zero point energy.
2. Be careful when using Gaussian! The Gaussian log file has two sections related to vibrations. The first section is for the usual vibrational analysis to obtain harmonic frequencies. These frequencies are listed from lowest to highest. The second section is for anharmonicities. In the second section, the harmonic frequencies are listed again (along with the fundamental frequencies), but the ordering may not be the same as in the first section. The input for the codes distributed in MultiWell will accept the frequencies in any order, but the ordering of the harmonic frequencies must be consistent with the ordering used for the anharmonicities. For consistency, the Gaussian user must be sure to use the ordering of the harmonic frequencies as listed in association with the anharmonicities in the Gaussian log file.

7.3 Program ansum

(For definitions of terms, see Technical Note #2.)

This program uses an efficient multidimensional Monte Carlo integration method to compute sums of vibrational states for non-separable systems[47]. It uses the full second order (pair-wise) **X** matrix, which can be obtained (in some cases) from spectroscopic data, or from electronic structure calculations (e.g. GAUSSIAN[55] and CFour[56]).

To reduce execution time, $G_{anh}(E)$ is usually calculated at fairly large energy intervals, even though a much smaller energy grain size may be desired. Since the density of states is the derivative of the sum of states ($\rho(E) = \mathrm{d}G/\mathrm{d}E$), it is possible to obtain the density of states by fitting the computed $G_{anh}(E)$ to a suitable function and then taking the derivative. In this implementation, the sum of states is assumed to be described by an exponential function, which is fitted to each pair of adjacent $G_{anh}(E)$ values, and $\rho(E)$ is obtained from the derivative.

Running ansum

The executable (binary) program resides in the directory `multiwell<version>/bin`, where `<version>` is the version number (e.g. "`2010.0`") To execute the compiled program, enter the directory that contains the `ansum` data file ("`ansum.dat`") and then type `~/multiwell<version>/bin/ ansum`. Directory `multiwell<version>` is assumed to reside in your Home directory. The output file `ansum.out` is automatically placed in the directory that contains `ansum.dat`.

Data File Format

(For definitions of terms, see Technical Note #2.)

Line 1: `Title 1` (≤100 characters)

Line 2: `Title 2` (≤100 characters)

Line 3: `ns` (number of vibrational modes), `WW` (a keyword to designate frequencies as ω_e or ω^0)

`WW` = `We` for ω

`WW` = `Wf` for fundamental vibrations (0-1 transitions)

`WW` = `W0` for ω^0 [note: `0` is a zero]

(The distinction between ω and ω^0 is explained by eqs. (II,267) through (II,269) of Herzberg [52]. The computer code carries out the conversion between them.)

Line 4: `wa(i)` [i = 1 , `ns`] vibrational frequencies (cm^{-1})

Line 5: `KEYWORD` for reading the anharmonicity matrix:

= `lower` (for reading lower half of matrix, plus the diagonal)

= `upper` (for reading upper half of matrix, plus the diagonal)

Lines 6 to `ns`+5: `xa(k,j)` anharmonicities (cm^{-1}) in upper or lower halves (plus the diagonal) of the `ns`×`ns` anharmonicity matrix.

Line `ns`+6: `J` , $\mathtt{B}_\mathrm{K}$

J = total angular momentum quantum number (for symmetric top)
B_K = rotational constant (cm^{-1}) for the K-rotor of a symmetric top.
For a symmetric top with rotational constants A, B=C, the rotational constant for the K-rotor is B_K = A-B.

Line ns+7: Estart, Emax, Estep, Beta, TrialMax, Percent
Estart = initial energy (cm^{-1})
Emax = maximum energy (cm^{-1})
Estep = energy difference between consecutive calculations (cm^{-1})
Beta = multiplier (≥1) to establish sampling boundary
TrialMax = maximum number of Monte Carlo samples
Percent = desired relative error in Sum of States

Notes

1. Beta = 1.01 usually works reasonably well; smaller values greater than unity (e.g. 1.001) may be needed to improve numerical efficiency for large molecules. Larger values should also be tested, although the numerical efficiency will be reduced.

2. Calculations are terminated by either achieving Percent, or by reaching TrialMax, depending on which is reached first.

3. To achieve small relative errors (Percent) requires large numbers (TrialMax) of Monte Carlo samples. The actual numbers required depend on the number of vibrational modes (ns) and other factors. Before carrying out a long, time-consuming run, it is recommended that a quick preliminary run be carried out using Percent = 10 (for example).

4. Because the densities of states are calculated by finite differences, the results are more accurate when small energy steps (small Estep) are used. Alternatively, the sums of states can be fitted by least squares to some differentiable function which can be differentiated to obtain the densities of states.

5. Energies are relative to the zero point energy.

6. The K-rotor quantum states are constrained to the range from -J to +J, which is proper for rigid rotors.[18]

NOTE. In second order vibrational perturbation theory (VPT2), the zero point energy includes contributions from the harmonic frequencies, the anharmonicities, and an additional term denoted ε_0 (see Eq. 9 in Section 9.9).[50,51] This term, which cancels-out in spectroscopy applications but must be included for thermochemistry and kinetics, is computed by electronic structure programs that employ VPT2 (e.g. GUSSIAN and CFOUR), but it is not computed by MultiWell. For *ab initio* kinetics calculations involving small molecules the term may be significant, but it is very small for molecules with small rotational constants. For high accuracy *ab initio* kinetics calculations (e.g. Ref. [7], ε_0 can be included manually by modifying the reaction critical energy.

Example Data File

```
CH2Br [A. Baldacci et al., JPCA 113, 6083-6090 (2009)]
Fundamentals from experiments, Xij from CCSD(T)/cc-pVTZ

9 , 'Wf'

3001.044
2243.802
1424.944
1225.21
768.76
595.0
3052.96
1252.55
930.281
upper
-28.4   -2.5  -8.1   -3.6   -0.9  +1.4  -117.6  -10.8  -7.0
-32.0   -3.4  -3.5   -3.3    0.0  -0.2    -9.3   +2.3
 -6.7   -2.5  -2.2   -0.6  -22.1  -2.8    -7.8
 -4.4   -3.6  -1.6  -10.2   -6.2  -0.1
 -1.4   -4.5  -1.3   -4.4   +3.3
 -3.5   +2.2  -1.1   -6.2
-33.0  -12.2  -6.4
 -1.9   -6.7
 -0.6

0 , 3.674

5000.  50000. , 5000. , 1.001 , 1.e+08 , 1.0
```

Example Output File

```
     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
     @@
     @        PROGRAM ANSUM v.1.02   [May 2009]    @
     @       by John R. Barker   @
     @@
     @      Sums of states for non-separable systems        @
     @@
     @  [J.R. Barker, J.Phys.Chem., 91, 3849-3854 (1987)]   @
     @@
     @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@



 Fri May 22 10:13:02 2009
 Elapsed User Time:        0.000 sec
 Total Elapsed Time:       0.000 min

CH2Br [A. Baldacci et al., JPCA 113, 6083-6090 (2009)]
Fundamentals from experiments, Xij from CCSD(T)/cc-pVTZ




  9 Vibrational Frequencies (cm-1)
  We; E = We*(v+1/2) +...; Wo: E = W0*v +...;  0<->1 fund.
  De: BDE for Separable Morse oscillator [0.0 if non-Morse]

  No.     WeWo      0-1_Fund.   De(cm-1)
    1   3132.394   3029.444   3001.044    80788.1
    2   2317.752   2275.802   2243.802    40463.1
    3   1463.094   1431.644   1424.944    76477.8
    4   1249.660   1229.610   1225.210    85905.7
    5    780.010    770.160    768.760   105919.0
    6    607.200    598.500    595.000    25585.9
    7   3202.860   3085.960   3052.960    72145.1
    8   1283.100   1254.450   1252.550   207058.5
    9    945.781    930.881    930.281   361058.1

(Anharmonic) Zero point energy (cm-1) =   7397.076


Anharmonicity Matrix (cm-1)

123456789
  1   -28.400    -2.500    -8.100    -3.600    -0.900     1.400  -117.600   -10.800    -7.000
  2    -2.500   -32.000    -3.400    -3.500    -3.300     0.000    -0.200    -9.300     2.300
  3    -8.100    -3.400    -6.700    -2.500    -2.200    -0.600   -22.100    -2.800    -7.800
  4    -3.600    -3.500    -2.500    -4.400    -3.600    -1.600   -10.200    -6.200    -0.100
  5    -0.900    -3.300    -2.200    -3.600    -1.400    -4.500    -1.300    -4.400     3.300
  6     1.400     0.000    -0.600    -1.600    -4.500    -3.500     2.200    -1.100    -6.200
  7  -117.600    -0.200   -22.100   -10.200    -1.300     2.200   -33.000   -12.200    -6.400
  8   -10.800    -9.300    -2.800    -6.200    -4.400    -1.100   -12.200    -1.900    -6.700
  9    -7.000     2.300    -7.800    -0.100     3.300    -6.200    -6.400    -6.700    -0.600

Angular momentum J =   0
K-rotor rotational constant (cm-1) Bk = 3.674E+00

Beta =  1.0010

Maximum number of Monte Carlo samples = 1.0E+08



OUTPUT (without densities)
 Energy(cm-1)  Ntrials   Efficiency    Sums     %_Error

    5000.0   8.142E+05    9.98E-01   4.610E+02    1.00
   10000.0   3.660E+06    9.93E-01   1.521E+04    1.00
   15000.0   9.107E+06    9.86E-01   1.882E+05    1.00
   20000.0   1.218E+07    9.81E-01   1.360E+06    1.00
   25000.0   1.511E+07    9.70E-01   6.753E+06    1.00
   30000.0   2.139E+07    9.65E-01   2.788E+07    1.00
   35000.0   3.270E+07    9.60E-01   1.018E+08    1.00
   40000.0   3.524E+07    9.53E-01   3.125E+08    1.00
   45000.0   3.330E+07    9.45E-01   8.479E+08    1.00
   50000.0   3.475E+07    9.40E-01   2.229E+09    1.00

 Fri May 22 10:47:46 2009
 Elapsed User Time:     2076.610 sec
 Total Elapsed Time:      34.639 min



OUTPUT (including densities)
 Energy(cm-1)  Ntrials   Efficiency    Sums     %_Error    Densities

    5000.0   8.142E+05    9.98E-01   4.610E+02    1.00    4.439E-01
   10000.0   3.660E+06    9.93E-01   1.521E+04    1.00    9.141E+00
   15000.0   9.107E+06    9.86E-01   1.882E+05    1.00    8.458E+01
   20000.0   1.218E+07    9.81E-01   1.360E+06    1.00    4.869E+02
   25000.0   1.511E+07    9.70E-01   6.753E+06    1.00    2.040E+03
   30000.0   2.139E+07    9.65E-01   2.788E+07    1.00    7.565E+03
   35000.0   3.270E+07    9.60E-01   1.018E+08    1.00    2.461E+04
   40000.0   3.524E+07    9.53E-01   3.125E+08    1.00    6.623E+04
   45000.0   3.330E+07    9.45E-01   8.479E+08    1.00    1.666E+05
   50000.0   3.475E+07    9.40E-01   2.229E+09    1.00    4.307E+05
```

7.4 Program sctst

This program is a new implementation[7] of the semi-classical transition state theory (SCTST) formulated by W. H. Miller and coworkers.[12,13,15,57] Our implementation is based on the Wang and Landau algorithm[48] and the density of states algorithm of Basire et al.,[49] as refined by Nguyen and Barker.[6] In particular, the code is an extension of program adensum (see above). See section 9.9 of this User Manual for an outline of the method. The literature references should be consulted for further details about the theory, its usefulness, and applications. The literature should be cited as described at the Preface of this User Manual.

The SCTST is a powerful method, since it can be used for computing both thermal (i.e. canonical k(T)) and microcanonical (i.e. k(E)) rate constants, which are particularly useful for master equation calculations. Unlike ordinary transition state theory, the SCTST is fully non-separable: all of the degrees of freedom may be coupled and it is not necessary to assume that the reaction coordinate is separable from the others. It also accounts naturally for zero point energy and for quantum mechanical tunneling along the curved reaction path in hyperdimensional space. In test cases, it is highly accurate.

The input is similar to that for program adensum, but it also includes the imaginary frequency and the anharmonicity coefficients associated with the reaction coordinate. The imaginary frequency and the off-diagonal anharmonicity coefficients associated with the reaction coordinate are pure imaginary numbers; the diagonal anharmonicity for the reaction coordinate is a pure real number. These facts affect how the parameters are specified in the input data file sctst.dat.

The output files are analogous to the output files from programs adensum and densum. The general output file sctst.out summarizes the input and gives "sums" and "densities" of states for the full energy range specified. These "sums" and "densities" actually correspond to the cumulative reaction probability[12,13,15,57] (CRP) and the corresponding energy derivative, which are equal to the sums and densities in the absence of quantum tunneling. The general output file also gives the thermal partition function (excluding the translations) for the transition state at a set of 100 fixed temperatures (25 K to 3383 K). The second output files takes the name of the transition state with ".dens" added: <name>.dens, just as in programs densum and adensum. These files are formatted for use by the MultiWell master equation code. In addition, a file <name>.qcrp is generated for use by program Thermo to calculate canonical rate constants (see Section 5.3 of this User Manual for a description of how the file is used by Thermo).

Running sctst

The executable (binary) program resides in the directory multiwell<version>/bin, where <version> is the version number (e.g. "2010.0") To execute the compiled program, enter the directory that contains the sctst data file ("sctst.dat") and then type ~/multiwell<version>/bin/sctst. Directory multiwell<version> is assumed to reside in your Home directory. The output file sctst.out is automatically placed in the directory that contains sctst.dat. A second output file <name>.dens, which is designed to be an input file for the MultiWell master equation code, is also generated, where `<name>` is the name of the transition state specified in `sctst.dat`.

Like `adensum`, `sctst` has been designed to run using a "checkpoint file", which is generated after the density of states of the fully coupled vibrations have been calculated, which is the most time-consuming part of the calculation. The next part of the calculation convolutes the separable degrees of freedom, if any are specified. The separable degrees of freedom may include free or hindered rotors, harmonic vibrations, etc. In particular, the 1-D external rotation (the K-rotor) and possibly also the 2-D external rotational degrees of freedom may be specified, depending on the application. If desired, the densities of states may be recalculated using various combinations of separable degrees of freedom, but retaining the same set of fully coupled vibrations. Repetitive calculations like these can be restarted using the checkpoint file, which saves considerable computer time.

Another reason for running repetitive calculations by starting with the checkpoint file is because whenever the reaction threshold energy is changed, the SCTST requires that the density of states of the transition state and the cumulative reaction probability be recalculated.[7] Adjusting reaction barriers may require multiple repetitive calculations, and restarting by using the checkpoint file will reduce the computer time by an order of magnitude.

Three output files are generated by `sctst`: `sctst.out`, `<name>.dens`, and `<name>.qcrp`. The first file (`sctst.out`) is a standard output file, which summarizes all input and output. The second file (`<name>.dens`), which takes a chemical species name as part of the file name, tabulates the cumulative reaction probability (CRP) and its derivative (analogous to sums and densities of states), formatted as double arrays; it can be used as input for the MultiWell master equation code. By using this input file, master equation calculations can be carried out with microcanonical rate constants (i.e. k(E)) computed by including the fully-coupled vibrations. The third file (`<name>.qcrp`) is formatted as a supplementary input file for program `Thermo`; it tabulates the partition function for the CRP and other thermodynamic properties of the coupled vibrational modes at a set of fixed temperatures from 50 K to >3000 K. `Thermo` can utilize these partition functions to compute thermal rate constants using the SCTST. See Section 5.3 of this User Manual for a description.

Data File Format

(For definitions of terms, see Technical Note #9 and Nguyen et al.[7])

Line 1: `Name` (transition state name; ≤10 characters)

Line 2: `Title 1` (≤100 characters)

Line 3: `Title 2` (≤100 characters)

Line 4: `ns` (number of orthogonal vibrational modes, not inclusing the reaction coordinate), `NY` (number of **Y** matrix elements), `NZ` (number of **Z** matrix elements), `WW` (a keyword to designate frequencies as ω_e or ω^0)

`WW` = `We` for ω

`WW` = `Wf` for fundamental vibrations (0-1 transitions)

`WW` = `W0` for ω^0 [note: `0` is a zero]

(The distinction between ω and ω^0 is explained by eqs. (II,267) through (II,269) of Herzberg [52]. The computer code carries out the conversion between them.)

Lines 5: `wa(i)` [i = 1 , `ns`] vibrational frequencies (cm^{-1}) of the orthogonal vibrations (i.e. not including the reaction coordinate)

Line 6: `KEYWORD` for reading the **X** anharmonicity matrix (not including the reaction coordinate):
= `lower` (for reading lower half of matrix, plus the diagonal)
= `upper` (for reading upper half of matrix, plus the diagonal)

Lines 7: `xa(k,j)` anharmonicities (cm^{-1}) in upper or lower halves (plus the diagonal) of the `ns`×`ns` anharmonicity matrix.

Line 8: `Nsep`, `Vrot`

`Nsep` = number of separable modes to be convoluted with the `ns` non-separable modes)
`Vrot` = KEYWORD for specifying units for internal and external rotations, except for hindered rotors of type **`hrd`** (see special instructions in the chapter on DenSum for type `hrd`)
`'AMUA'`: for moments of inertia input with units of amu Å^2.
`'GMCM'`: for moments of inertia input with units of g cm^2.
`'CM-1'`: for rotational constant input with units of cm^{-1}.
`'MHZ'`: for rotational constant input with MHz.
`'GHZ'`: for rotational constant input with GHz.
(some combinations of upper/lower case are also accepted)

Line 9: input data for the separable modes (classical or quantum free rotors, hindered rotors of several types, and Morse oscillators). Use the same input format for these degrees of freedom as used in `DenSum` (see the MultiWell User Manual)

Line 10: `Egrain1, imax1, Isize, Emax2, KEYWORD1`

`Egrain1` = energy grain (cm^{-1})
`imax1` = number of array elements in lower part of the Double Array (see MultiWell User Manual)
`Isize` = total number elements in the Double Array.
`Emax2` = upper energy limit
`KEYWORD1` = controls number of stochastic trials:
`Fair:` low statistical accuracy (10^2 trials per energy grain)
`Good:` good statistical accuracy (10^3 trials per energy grain).
`Better:` better statistical accuracy (10^4 trials per energy grain)
`Best:` best statistical accuracy (10^5 trials per energy grain)
`Extra:` best statistical accuracy (10^6 trials per energy grain, but no more than 2×10^9 total trials for all energy grains)

Line 11: `chekpoint`

If `chekpoint = "chekstart"`, then the new calculation starts by reading the pre-existing checkpoint file. Otherwise, the full calculation if is carried out. All data up to and including Line 10 must be identical to the data file used to create the pre-existing checkpoint file in the first place.

Line 12: `KEYWORD2`

`KEYWORD2` = controls number of samples for the semi-classical calculation (typically the same as `KEYWORD1`). The samples are used to compute average quantities employed by the theory.

`Fair:` low accuracy (10^2 samples per energy grain)
`Good:` good accuracy (10^3 samples per energy grain).
`Better:` better accuracy (10^4 samples per energy grain)
`Best:` best accuracy (10^5 samples per energy grain)
`Extra:` best accuracy (10^6 samples per energy grain, but no more than 2×10^9 total samples for all energy grains)

Line 13: `Vf` , `Vr` , `Eunits`

Barrier heights in forward (`Vf`) and reverse (`Vr`) directions, and their energy units (`Eunits`: `kcal`, `kjou`, or `cm-1`)

Line 14 : `vimag`, `Xff`

Imaginary frequency and the diagonal anharmonicity coefficient for the reaction coordinate (units of cm^{-1})

Line 15: `Xkf`

Off-diagonal anharmonicity coefficients (units of cm^{-1}) multiplied by i = $(-1)^{1/2}$: `ns` of them, in order. Since the $X_{k,F}$ coefficients are pure imaginary, they can be written $X_{k,F} = i\tilde{X}_{k,F}$, where $\tilde{X}_{k,F}$ is a real number. The quantities to be input are $i \cdot X_{k,F} = i \cdot i\tilde{X}_{k,F} = -\tilde{X}_{k,F}$, which is in the same format as GAUSSIAN output. In contrast, CFOUR output gives $X_{k,\mathrm{F}}$ expressed as $\tilde{X}_{k,F}$ followed by the symbol "i", i.e. " $\tilde{X}_{k,F}$ i". These conventions are equivalent, but it is easy to make a sign error. Be careful!

Notes

1. Energies are relative to the zero point energy.

2. In second order vibrational perturbation theory (VPT2), the zero point energy includes contributions from the harmonic frequencies, the anharmonicities, and an additional term denoted E_0 (see Eq. 9 in Section 9.9).[50,51] This term, which cancels-out in spectroscopy applications but must be included for thermochemistry and kinetics, is computed by electronic structure programs that employ VPT2 (e.g. GUSSIAN and CFOUR), but it is not computed by MultiWell. For *ab initio* kinetics calculations involving small molecules the term may be significant, but it is very small for molecules with small rotational constants. For high accuracy *ab initio* kinetics calculations (e.g. Ref. [7], E_0 can be included manually by modifying the reaction critical energy.

3. Be careful when using Gaussian! The Gaussian log file has two sections related to vibrations. The first section is for the usual vibrational analysis to obtain harmonic frequencies. These frequencies are listed from lowest to highest. The second section is for anharmonicities. In the second section, the harmonic frequencies are listed again (along with the fundamental frequencies), but the ordering may not be the same as in the first section. The input for the codes distributed in MultiWell will accept the frequencies in any order, but the ordering of the harmonic frequencies must be consistent with the ordering used for the anharmonicities. For consistency, the Gaussian user must be sure to use the ordering of the harmonic frequencies as listed in association with the anharmonicities in the Gaussian log file.

8. lamm: Effective mass for large amplitude motion

Program Author: **Thanh Lam Nguyen**
Contact: nguyenlt@umich.edu, Tel: (734) 936-0895

8.1 Introduction

This program is a tool for obtaining the effective mass (as a function of angle or distance) for large amplitude motions, such as internal rotation. The output is used as part of the input data file needed for computing sums and densities of states (using program DenSum) and partition functions (program Thermo).

Program lamm uses the ro-vibrational G matrix-based algorithm described in the papers of Harthcock et al.[58,59] and others to compute the effective reduced mass for one-dimensional torsions and other large amplitude motions, which are assumed to be separable from the other molecular degrees of freedom.

Molecular geometries can be computed at discrete values of χ by using any of the many available quantum chemistry codes, such as Gaussian,[55] Molpro,[60] CFour,[56] etc. Program lamm is intended to help users compute the torsion rotational constant $B_{hr}(\chi)$ and the potential energy $V(\chi)$, based on the output from the quantum chemistry program.

For hindered internal rotation, lamm employs Cartesian coordinates obtained from quantum chemical calculations carried out as a function of the dihedral angle χ ($0 \leq \chi < 2\pi$). The output of lamm consists of the torsion rotational constant function $B_{hr}(\chi)$ (or the corresponding moment of inertia function $I(\chi)$) tabulated as a function of χ. The potential energy can also be obtained from the quantum chemistry calculations and expressed as a function of χ. These functions must then be fitted by the user to any one of the several truncated Fourier series that are recognized by DenSum and Thermo (see the MultiWell User Manual). The Fourier coefficients are used by programs DenSum and Thermo to solve the Schrödinger equation for the energy eigenstates (see Section 9.8), which are needed to compute densities of states and partition functions, respectively.

A script is provided to assist users in extracting the required information from Gaussian output files (see Section 8.7).

8.2 Compiling and Running lamm

The source directory for lamm resides in the MultiWell src directory. The program is compiled on *Linux* by entering the MultiWell directory and typing “make” (omit the quotation marks), followed by a carriage return. The executable file is stored in the MultiWell bin directory. The default compiler is gfortran. The compiler can be changed by revising lines in file src/lamm/Makefile. To remove old object files and executables, enter directory src/lamm and type “make clean”, followed by a carriage return.

To execute the compiled program, enter the directory that contains the lamm data file (“lamm.dat”) and then type ~/multiwell-<version>/bin/lamm, where <version> refers to the multiwell version number (e.g. "2010.0"). This command executes the binary executable file. The output file (“lamm.out”) is automatically placed in the directory that contains lamm.dat.

8.3 Notes and Limits

For the current version of the `lamm` program, Cartesian coordinates of optimized geometries must be obtained by constrained optimizations. When using Gaussian, the keyword "`NOSYM`" and/or OPT="`Z-Matrix`", or the equivalent must be specified.

The `lamm` program does not automatically identify the molecular symmetry; it always assumes that the molecule has no symmetry (C_1 symmetry). Therefore the effective reduced moment of inertia will be computed only for the number of optimized geometries that are input.

Since the derivatives of the nuclear positions with respect to the torsional coordinate are computed by central finite differences, a fairly small torsional angle step-size must be used in order to achieve numerical accuracy. In our experience, a step-size of ≤10° is a good choice for most purposes.

In addition to torsional motions, the `lamm` program can be used for computing the effective mass for other separable large-amplitude vibrational motions. Specific implementations will be added in the future.

Users may be interested in program `I_Eckart`, which was coded in `MATLAB` by Wong, Thom and Field [61]. We would like to thank Dr. Wong and Prof. Field for useful discussions and for providing a copy of their program, which we used for benchmarks.

8.4 Data File Format

Line 1: Name of molecule (≤ 100 characters)

Line 2: Title (≤ 100 characters)

Line 3: $\mathtt{N}_{\mathtt{atom}}$ (number of atoms in molecule)

Line 4: `Amin` (lower limit of torsion angle in degrees), `M` (number of optimized geometries), `DELA` (step-size of torsion angle in degrees).

Line 5+ (lines 5 to 4 + N): list of atomic masses (in amu)

Line 6+ (lines 5 + N to 4 + N + N*M): list of Cartesian coordinates (in Angstroms) for `M` optimized geometries as a function of the torsion dihedral angle.

Line 7 (lines 5 + N + N*M): comment line (≤100 characters)

Line 8+ (lines 6 + N + N*M to 5 + N + N*M + M): list of torsional angle (in degrees) and relative energies (in cm^{-1}).

8.5 Example Data File

```
H2O2
Calculated at 6-311G** level of theory

4      ! Number of atoms in the molecule

0, 36, 10       ! Minimum angle, No. of optimized geometries and Step-size
(in degrees)

1.     ! list of atomic mass (in amu)
16.
16.
1.

0.000000 0.000000 0.000000 !list of X, Y, Z cartesian coordinate
0.000000 0.000000 0.966404
1.412522 0.000000 1.340839
1.891238 0.000000 0.501335

0.000000 0.000000 0.000000
0.000000 0.000000 0.966493
1.412776 0.000000 1.339296
1.886024 0.162275 0.512367
```

And so on ....

8.6 Example Output:

```
H2O2
Calculated at 6-311G** level of theory

WARNING: The zero of the relative energy is arbitrary minimum
INDEX    ANG(DEG)    ANG(RAD)      E(cm-1)      I(amu.A**2)   B(cm-1)
     10.0       0.000000      2980.5       0.387       43.6
     2        10.0       0.174533      2913.6       0.390       43.3
     3        20.0       0.349066      2720.4       0.393       42.9
     4        30.0       0.523599      2421.2       0.396       42.5
     5        40.0       0.698132      2047.2       0.400       42.2
     6        50.0       0.872665      1635.9       0.403       41.9
     7        60.0       1.047198      1224.6       0.405       41.6
     8        70.0       1.221730       846.6       0.407       41.4
     9        80.0       1.396263       527.1       0.410       41.2
    10        90.0       1.570796       282.2       0.412       40.9
    11       100.0       1.745329       116.9       0.415       40.6
    12       110.0       1.919862        27.0       0.418       40.4
    13       120.0       2.0943950.0       0.420       40.2
    14       130.0       2.268928        19.3       0.421       40.0
    15       140.0       2.443461        65.6       0.423       39.9
    16       150.0       2.617994       122.1       0.423       39.8
    17       160.0       2.792527       173.9       0.423       39.8
    18       170.0       2.967060       209.5       0.423       39.8
    19       180.0       3.141593       222.3       0.424       39.8
    20       190.0       3.316126       209.5       0.423       39.8
    21       200.0       3.490659       173.9       0.423       39.8
    22       210.0       3.665191       122.1       0.423       39.8
    23       220.0       3.839724        65.6       0.423       39.9
    24       230.0       4.014257        19.3       0.421       40.0
```

```
25        240.0        4.1887900.0       0.420        40.2
26        250.0        4.363323        27.0        0.418        40.4
27        260.0        4.537856       116.9        0.415        40.6
28        270.0        4.712389       282.2        0.412        40.9
29        280.0        4.886922       527.1        0.410        41.2
30        290.0        5.061455       846.6        0.407        41.4
31        300.0        5.235988      1224.6        0.405        41.6
32        310.0        5.410521      1635.9        0.403        41.9
33        320.0        5.585054      2047.2        0.400        42.2
34        330.0        5.759587      2421.2        0.396        42.5
35        340.0        5.934119      2720.4        0.393        42.9
36        350.0        6.108652      2913.6        0.390        43.3
37        360.0        6.283185      2980.5        0.387        43.6
```

8.7 gauss2lamm: A script for generating lamm.dat

The shell script **gauss2lamm.sh** is designed to create data files for program lamm (see Chapter 8), which computes effective mass for large amplitude motions, such as hindered internal rotation. The script is located in directory multiwell-<version>/bin, where <version> refers to the version number (e.g. "2010.0"). This script reads the input orientations or Z-matrix of optimized geometries and corresponding energies as a function of the appropriate coordinate (i.e. the torsion dihedral angle) from "____.log" or "____.out" output files produced by GAUSSIAN-03[46] or GAUSSIAN-09[55] software.

To execute the script, enter the directory that contains the Gaussian output file and then type ~/multiwell-<version>/bin/gauss2lamm.sh. This command goes to the directory multiwell-<version>/bin (which is assumed to reside in your Home directory) and executes the script file gauss2lamm.sh, which stored there.

The script generates lamm.dat in the Gaussian output file directory.

The script prompts the user to input the Gaussian output file name, number of atoms in the molecule and minimum angle, no. of points and stepsize (in degrees) used in the calculation. First it verifies whether the user has used the keyword “NOSYM and/or Z-Matrix” in Gaussian output. If the keyword is found the script writes lamm.dat containing geometry and energy (in cm^{-1}) information. The energy is tabulated as a function of angle (in degrees). The zero of relative energy is arbitrary minimum. If the keyword is not found in the Gaussian output, the script prompts the user to redo the Gaussian calculation using the keyword “NOSYM and/or Z-Matrix”.

It is important to note that the lamm.dat file created automatically from Gaussian output needs manual changes. Line 5+ (lines 5 to 4+N) in the lamm.dat created using the script is atomic number, where N is number of atoms in the molecule. This has to be replaced with corresponding atomic masses (amu units) for subsequent use by program lamm.

An example Gaussian output file lamm.log is available in the examples directory and procedure to execute gauss2lamm is shown below.

```
$ ./../bin/gauss2lamm.sh

Enter Gaussian Output Filename
lamm.log
```

```
     Keyword NOSYM found in the output

     Enter total number of atoms in the molecule
     5

     Enter MIN angle, No. of Points and Stepsize in degrees used
in the calculation
     0 36 10

     Proceeding to collect data from lamm.log

     Lowest energy in the surface scan: -740.457504043 hartrees
```

9. Technical Notes

9.1 Conversion factors

Moment of Inertia (Im):

$\mathrm{Im(amu\ \AA^2) = Im(g\ cm^2) / 1.66054\times10^{-40}}$
$\mathrm{Im(amu{*}\AA^2) = 16.85763 / B(cm^{-1})}$
$\mathrm{Im(amu{*}\AA^2) = 5.0538\times10^{5} / B(MHz)}$
$\mathrm{Im(amu{*}\AA^2) = 5.0538\times10^{2} / B(GHz)}$

Rotational Constant

$\mathrm{B(cm^{-1}) = B(Hz) / 2.99793\times10^{10}}$
$\mathrm{B(cm^{-1}) = B(MHz) / 2.99793\times10^{4}}$
$\mathrm{B(cm^{-1}) = B(GHz) / 2.99793\times10^{1}}$

9.2 Anharmonic Vibrations

Separable Anharmonic Vibrations

As described by Herzberg,[52] ω (= ν/c, where ν is the harmonic frequency and c is the speed of light) is used in the following expression for the vibrational energy, where s is the number of *separable* oscillators, the vibrational energy E_v is expressed in units of $\mathrm{cm^{-1}}$ (relative to the zero point energy E_z), and the other symbols have their usual meanings:

Error! Objects cannot be created from editing field codes. (1)

where the zero point energy is given by

$$E_z = \tfrac{1}{2}\sum_{i=1}^{N}\omega_i + \tfrac{1}{4}\sum_{i=1}^{N}X_{ii} \qquad (2)$$

In **`DenSum`** and **`Thermo`**, vibrational anharmonicities can be entered directly.

Evib = WE*(v+1/2) + ANH*(v+1/2)*(v+1/2), v=0,1,2,3,...

where parameter WE is the harmonic frequency and, for a Morse oscillator, ANH < 0. DenSum and Thermo assume *separable* degrees of freedom, where inter-mode couplings are neglected. Thus they can only consider the diagonal elements of the X_{ij} anharmonicity coefficient matrix. Thus parameter `ANH` is identified with the diagonal X_{ii} element.

Because Densum and Thermo are limited to separable modes, various choices can be made, depending on the information that is available. Sometimes the X_{ij} anharmonicity coefficients are available from experiments. They can also be computed as standard options by electronic structure codes, although they are expensive to calculate. They are computed from a perturbation theory treatment based on higher derivatives of the potential energy surface at the equilibrium geometry. Thus they are closely related to the computed harmonic frequencies, which are based on the 2nd derivatives of the same potential energy surface.

Possible Choices (in decreasing order of preference)

1. For densities and sums of states, treat the vibrations as *non-separable*. Use program **`adensum`** instead of **`DenSum`**. See the next section.

2. If the fundamental (i.e. "observed") frequency and the full (or diagonal) X_{ij} matrix are known, one should use the fundamental frequency as "`OBS`" and the X_{ii} diagonal term as `ANH`.

3. If the harmonic frequency and the full X_{ij} matrix are known, then one should first compute the fundamental frequency and then use Choice #1.

4. If the harmonic frequency and the diagonal terms of the X_{ij} matrix are known, then one should use the harmonic frequency as "`HAR`" and the X_{ii} diagonal term as `ANH`.

5. If the fundamental (i.e. "observed") frequency is known and the X_{ij} matrix is not, then one should use the fundamental frequency as "`OBS`" and `ANH=0.0`.

6. If the harmonic frequency is known and the X_{ij} matrix is not, then one should use the harmonic frequency as "`HAR`" and `ANH=0.0`.

7. If nothing is known, then one must use estimation methods.[62]

Non-Separable Anharmonic Vibrations

For asymmetric top molecules, the vibrational energy level relative to the zero point energy is given by the following perturbation theory expansion for the vibrational energy:[30,31,43]

$$\begin{aligned} E_v = -E_z + \sum_{i=1}^{N}\omega_i\left(\upsilon_i+\tfrac{1}{2}\right) + \sum_{i=1}^{N}\sum_{j=i}^{N}X_{ij}\left(\upsilon_i+\tfrac{1}{2}\right)\left(\upsilon_j+\tfrac{1}{2}\right) + \\ \sum_{i=1}^{N}\sum_{j=i}^{N}\sum_{k=j}^{N}Y_{ijk}\left(\upsilon_i+\tfrac{1}{2}\right)\left(\upsilon_j+\tfrac{1}{2}\right)\left(\upsilon_k+\tfrac{1}{2}\right) + \\ \sum_{i=1}^{N}\sum_{j=i}^{N}\sum_{k=j}^{N}\sum_{l=k}^{N}Z_{ijkl}\left(\upsilon_i+\tfrac{1}{2}\right)\left(\upsilon_j+\tfrac{1}{2}\right)\left(\upsilon_k+\tfrac{1}{2}\right)\left(\upsilon_l+\tfrac{1}{2}\right) + \ldots \end{aligned} \tag{3}$$

where ω_i is the harmonic oscillator frequency, X_{ij}, Y_{ijk}, and Z_{ijkl} are the anharmonicity constants, υ_i is the vibrational quantum number, N is the number of vibrational modes, and E_z is the zero-point vibrational energy, which is obtained by setting all $\upsilon_i = 0$:

$$E_z = \frac{1}{2}\sum_{i=1}^{N}\omega_i + \frac{1}{4}\sum_{i=1}^{N}\sum_{j=i}^{N}X_{ij} + \frac{1}{8}\sum_{i=1}^{N}\sum_{j=i}^{N}\sum_{k=j}^{N}Y_{ijk} + \frac{1}{16}\sum_{i=1}^{N}\sum_{j=i}^{N}\sum_{k=j}^{N}\sum_{l=k}^{N}Z_{ijkl} + \ldots \tag{4}$$

The array of harmonic frequencies (**ω**) and the matrices composed of the anharmonicity constants (**X**, **Y**, **Z**) can be either obtained from experiment or computed directly from first principles, e.g. by quantum chemistry software packages equipped with VPT2.

When all quantized vibrational energy levels are known, sums and densities of states can be counted exactly. From Eq. 3, all E_υ can in principle be computed when all allowed values of υ are known. Therefore, the problem reduces to finding all allowed values of υ_i up to a given total internal energy. For a separable harmonic oscillator, all quantum numbers from υ=0 to $\upsilon=\infty$ are allowed. A Morse oscillator, however, can dissociate and has only a finite number of vibrational states ranging from υ=0 to $\upsilon=\upsilon_{max}$. The allowed quantum numbers therefore range from υ=0 to $\upsilon=\upsilon_{max}$. For a *coupled* set of anharmonic vibrations, the maximum quantum number allowed for the k^{th} vibration, $v_{D,k}$, depends on the quantum numbers assigned for all other vibrations. When the state energies are described by a perturbation theory expansion containing only **X** (i.e. **Y** and **Z** = 0), $\upsilon_{D,k}$ can be found analytically[27]:

$$\upsilon_{D,k} = \left[-\frac{\omega_k + \sum_{j\neq i} X_{kj}\left(\upsilon_j + \frac{1}{2}\right)}{2X_{kk}} - \frac{1}{2} \right] \tag{5}$$

It must be remembered that $\upsilon_{D,k}$ depends on all of the other quantum numbers. When $X_{kk}<0$, then $\upsilon_{D,k}$ is the quantum number for the highest bound level of the k^{th} vibration and the corresponding dissociation energy (relative to the zero point energy) of this vibrational mode is D_k:

$$D_k = -\frac{\left(\omega_k + \sum_{j\neq k} X_{jk}\left(\upsilon_j + \frac{1}{2}\right)\right)^2}{4X_{kk}} - \left(\frac{1}{2}\omega_k + \sum_{j\neq k}\frac{1}{4}X_{jk}\right) \tag{6}$$

This dissociation energy corresponds to the energy at which E_υ reaches a maximum as υ_k is increased, i.e. $(\partial E_\upsilon/\partial \upsilon_k) = 0$ at $\upsilon_k = \upsilon_{D,k}$.

While computing the sums of states from E_υ=0 to a maximum value $E_\upsilon=E_{tot}$, the range of allowed quantum numbers is further restricted. Suppose one proceeds by first starting with all quantum numbers set to zero and then assigning quantum numbers one at a time. After assigning k-1 quantum numbers and before assigning the k^{th}, υ_k=0 and the remaining unassigned energy E_u is given by

$$E_u = E_{tot} - E_{v,k-1} \tag{7}$$

where $E_{\upsilon,k-1}$ is the vibrational energy E_υ computed using the k-1 quantum numbers already assigned. The range of possible quantum numbers for the k^{th} oscillator is therefore υ_k=0 to $\upsilon_{max,k}$, where $\upsilon_{max,k}$ depends on the amount of unassigned energy:

$$\upsilon_{\max,k} = \upsilon_{D,k} \qquad \text{for } E_u \geq D_k \tag{8a}$$

$$\upsilon_{\max,k} = \upsilon_{D,k}\left[1 - \sqrt{1 - E_u/D_k}\right] \qquad \text{for } E_u \leq D_k \tag{8b}$$

When the **Y** and/or **Z** higher-order anharmonicities are included in Eq. 3, analytical solution for $\upsilon_{D,k}$ is not obtainable and iterative calculations are necessary. In the present work, $\upsilon_{max,k}$ is computed iteratively by increasing trial values of υ_k by unit steps from 0 and monitoring $E_\upsilon(\upsilon_k)$ while all other quantum numbers (υ_i, $i \neq k$) are held constant. The iterative procedure stops as soon as $E_\upsilon(\upsilon_k)$ becomes smaller than $E_\upsilon(\upsilon_k\text{-}1)$. This decrease corresponds to passing the maximum where $(\partial E_\upsilon/\partial \upsilon_k) = 0$. For a state to be "bound" with respect to the k^{th} degree of freedom, the derivative must be greater than zero for the trial value of υ_k: $(\partial E_\upsilon/\partial \upsilon_k)>0$. This process is repeated by increasing υ_k until the $(\partial E_\upsilon/\partial \upsilon_k)<0$ for a trial value of υ_k. The assignment of $\upsilon_{D,k}$ is then based on the sign of the first derivative $(\partial E_\upsilon/\partial \upsilon_k)$ at $\upsilon_k\text{-}1$ (see Figure 1), which enables one to ensure that the state with $\upsilon_{D,k}$ is lower in energy than the classical maximum E_υ. When $(\partial E_\upsilon/\partial \upsilon_k)$ is >0 at $\upsilon_k\text{-}1$, then $\upsilon_{D,k} = \upsilon_k\text{-}1$; otherwise, $\upsilon_{D,k}$ is set equal to $\upsilon_k\text{-}2$.

At energies above the dissociation limit, quasi-bound states exist, where the quantum number for every mode does not exceed the dissociation limit for that particular mode. Of course, if the energy is redistributed, dissociation can take place. For computing RRKM rate constants, quasi-bound states must be counted. Quasi-bound states are identified when the partial derivatives $(\partial E_\upsilon/\partial \upsilon_i)>0$ for all i. Therefore, after all quantum numbers [63] have been assigned, the partial derivatives $\{\partial E_\upsilon/\partial \upsilon_i\}_{i=1,\,N}$ are computed and the assigned $\{\upsilon_i\}$ is accepted if all elements are positive; otherwise the state is rejected.

9.3 Vibrational Degeneracies

In **DenSum** and **Thermo**, vibrational degeneracies are treated as accidental. Identical results are obtained if each vibrational degree of freedom with degeneracy NG is entered in the data file NG times as a non-degenerate vibration.

For Harmonic oscillators, the state energies for accidental degeneracies are exactly the same as for true degeneracies that arise, for example, from symmetry considerations. For anharmonic vibrations, the state energies are no longer quite the same for accidental and for true degeneracies, but the differences are small and can be neglected for most kinetics purposes.

[G. Herzberg, "Infrared and Raman Spectra", D. van Nostrand Co., Inc., 1945, p. 210 ff]

9.4 External molecular rotations

All non-linear polyatomics have principal moments of inertia I_A, I_B, I_C and corresponding rotational constants $A = \hbar^2/2I_A$, etc. When two of the rotational constants are equal to each other, the molecule is termed a *symmetric top*. In the following, we will assume that $A \neq B = C$. The rotational energy of a symmetric top is given by

$$E_r(J,K) = BJ(J+1) + (A-B)K^2$$

where quantum number J refers to a **two-dimensional 2-D adiabatic rotor** (i.e. one that conserves angular momentum $\boldsymbol{J}$) and K refers to a **one-dimensional rotation** about the top axis (projection of $\boldsymbol{J}$ on the top axis). For a rigid rotor, the K quantum number is constrained to take integer values from $-J$ to $+J$. Thus each value of $|K|$ is doubly degenerate except for $|K|$=0, which is singly degenerate.

The rotational states of **asymmetric tops** are not easily calculated, but sufficiently accurate partition functions and densities of states (but not for high resolution spectroscopy) are obtained by averaging the two rotational constants that are most similar to each other and treating the asymmetric top *approximately* as a symmetric top. **For use in DenSum** one can use either of two reasonable approximations for the effective rotational constant B_{2D} for the *J*-Rotor (when $B \approx$ C, for example):

$$B_{2D} = (BC)^{1/2} \text{ [Ref. }^{16,18}\text{] or } B_{2D} \approx \frac{1}{2}(B+C) \text{ [Ref. }^{64}\text{]}$$

The effective rotational constant B_K for the K-rotor is given by

$$B_K = (A - B).$$

(This is often pragmatically approximated as $B_K \approx A$.) **For use in Thermo**, the best approximate forms are $B_{2D} = (BC)^{1/2}$ and $B_K = A$ since these expressions give the correct partition functions for all non-spherical, non-linear molecules. The corresponding rotational energy becomes

$$E_r(J,K) = B_{2D}J(J+1) + B_K K^2$$

Note that entering I_A, I_B, and I_C (or *A, B,* and *C)* in the data file for DenSum or Thermo is **NOT** correct, since the state densities and partition functions will not be computed correctly. In any non-linear polyatomic, the three external rotations are constrained by the total angular momentum: the three rotors are not independent. Thus inputting the three rotors as independent 1-D rotations leads to an incorrect result. The approximate symmetric top treatment, which is the convention adopted in MultiWell, includes the constraint and thus gives the right answer (with quite high accuracy).

Active Energy and the K-rotor

In a free molecule, only the total angular momentum is conserved. Thus the *J* quantum number is assumed to be adiabatic (conserved), but the *K* quantum number can vary within its constraints, as do the vibrational quantum numbers, subject to conservation of energy in the "active" degrees of freedom. The "active energy" is the energy associated with the vibrational quantum numbers and *K*, collectively. Thus it is said that the energy in the active degrees of freedom randomizes, or the *K*-rotor energy "mixes" with the vibrational energy.

For a **prolate symmetric top** (e.g. a cigar shape), $A>B$ and therefore $B_K >0$. For prolate symmetric tops, the rigorous constraint $-J \leq K \leq +J$ is often relaxed in the interest of more convenient computation of densities of states. This popular approximation is reasonably accurate except at the lowest total energies and is simple to implement because the rotational energy $E_r(J,K) \geq 0$ for all values of *J* and *K*.

For an **oblate symmetric top** (e.g. disk-shaped), $A<B$ and hence $B_K<0$; this can give $E_r(J,K)$ <0 when the unconstrained $K >> J$. *Thus the approximate treatment of the K-rotor may fail seriously for oblate tops* in Densum. Moreover, densities of states are usually computed by treating the J-rotor and the K-rotors as separable. Thus the term $B_K K^2$ in the expression for rotational energy is <0, confounding the usual methods for computing densities of states, which assume only positive energies.

To utilize the simple approximation in DenSum, the K-rotor is declared as a simple 1-D free rotation (type "rot" or "qro") and included with the vibrational degrees of freedom when computing the density of states.

The ***kro*** degree of freedom type employs the correct constraints on *K* for a single user-specified value of *J*. Because it considers only a single value of *J*, it is not used for general applications.

The rotational degrees of freedom in a MultiWell master equation calculation are distributed in two input files: the **K-ROTOR** properties are listed in **densum.dat** and included in density and sums of states calculations. The **2-D ADIABATIC ROTOR** moment of inertia is listed in **multiwell.dat** on Line 8 (for wells) or Line 14 (for transition states).

9.5 Symmetry numbers, internal rotation, and optical isomers

External Symmetry Numbers

For use in DenSum, MultiWell, and Thermo, we recommend the following convention for symmetry numbers (described by Gilbert and Smith[65]): The EXTERNAL SYMMETRY NUMBER should not contain any contribution from internal rotors and the external symmetry numbers are *not* used in the calculation of sums and densities of states. INTERNAL ROTOR SYMMETRIES are explicitly included in the data lines for the specific internal rotors (DenSum and Thermo data files).

According to this convention, the EXTERNAL SYMMETRY NUMBER for Ethane is 3x2=6. The symmetry numbers for the external K-rotor and adiabatic 2-D-rotor are set equal to unity in both DenSum and Thermo. (Of course the codes are flexible enough so that other self-consistent conventions may be used instead.)

Hindered Internal Rotation

Internal rotors are characterized by both the symmetry of the potential energy (i.e. the foldedness, or number of minima (and maxima) per 2π rotation), and the symmetry of the mass distribution (or substituents) around the rotor. A potential energy surface computed using the Born-Oppenheimer approximation is independent of the masses of the nuclei. The potential energy surface for ethane internal rotation, for example, is 3-fold symmetric, but isotopic substitution can give either unsymmetrical or symmetrical mass distributions. If both the PES and the mass distribution have identical symmetries, then that is also the symmetry number of the internal rotor. It is also the number of "indistinguishable minima"; the symmetry number compensates for over-counting the number of indistinguishable minima when computing the partition function:[66]

$$q_{hr} = \frac{1}{\sigma_{hr}} \sum_{i=1}^{\infty} \exp\left(\frac{-E_i}{k_B T} \right)$$

where σ_{hr} is the symmetry number for the hindered rotor and E_i is the ith energy eigenvalue.

MultiWell has provision in Thermo and DenSum for both symmetrical and unsymmetrical hindered internal rotations. For user convenience, the parameters for **symmetrical hindered internal rotations** may be entered in any one of three ways, designated by degree of freedom types "`hra`", "`hrb`", and "`hrc`". These types assume that the reduced internal moment of inertia is constant (independent of dihedral angle) and the potential energy of the internal rotation is given by a simple cosine function:

$$V(\chi) = \frac{U}{2}\left[1 - \cos\left(\sigma_{hr}\chi\right)\right]$$

where U is the height of the hindrance barrier, and χ is the dihedral angle (for more complex symmetrical potentials, use type "`hrd`", which is discussed below). If $U = 0$, then the internal rotation should be modeled using a 1-D free rotor (types "`rot`" or "`qro`") with the symmetry number for the internal rotation.

Types "`hra`", "`hrb`", and "`hrc`" offer three convenient methods for entering the identical information. The information is identical because the harmonic frequency, hindrance barrier height, and the reduced moment of inertia for a symmetrical 1-D hindered internal rotation are related by the single equation:

$$\nu_t = \frac{\sigma_{hr}}{2\pi}\left(\frac{U}{2I_r}\right)^{1/2}$$

where ν_t is the harmonic torsion frequency, U is the height of the hindrance barrier, and I_r is the reduced moment of inertia. With any two of these quantities, the third is defined. Note that this equation is only valid for $\sigma_{hr}>1$.

Unsymmetrical hindered internal rotations ($\sigma_{hr}=1$) are entered in Thermo and DenSum by designating degree of freedom type "`hrd`". Unsymmetrical rotation can be used to describe internal rotation with just one well, or with two or more distinguishable wells. It can also be used for symmetrical potentials with more complicated potential energy functions. To calculate the internal rotor eigenvalues requires the potential energy function and the reduced moment of inertia (or rotational constant) as a function of dihedral angle. For an example, see the data for ClOOCl in the thermodynamics data base (directory `multiwell/thermo-database/`).

For convenience, three forms of the torsional potential energy are accepted by DenSum and Thermo (all coefficients in units of cm^{-1}):

Type **`Vhrd1`** $$V(\chi) = \sum_{n=1}^{N}\frac{V_n}{2}\left[1 - \cos\left(n\sigma_V\left(\chi + \varphi_V\right)\right)\right]$$

Type **`Vhrd2`** $$V(\chi) = V_0 + \sum_{n=1}^{N} V_n \cos\left(n\sigma_V\left(\chi + \varphi_V\right)\right)$$

Type **`Vhrd3`** $$V(\chi) = V_0 + \sum_{n=1}^{N} V_n^c \cos\left(n\sigma_V\left(\chi + \varphi_V\right)\right) + \sum_{n=1}^{N} V_n^s \sin\left(n\sigma_V\left(\chi + \varphi_V\right)\right)$$

where χ is the dihedral angle (radians). The parameters σ_V (symmetry number for the potential energy: `SYMMV`) and φ_V (phase angle for the potential: `PHASEV` (radians)) are merely for convenience in curve-fitting.

Also for convenience, either the rotational constant or the moment of inertia, which are functions of the dihedral angle, can be entered (all coefficients in units of cm^{-1}).

Type **`Bhrd1`** (all coefficients in units of cm^{-1})

$$B(\chi) = B_0 + \sum_{n=1}^{N} B_n \cos\left(n\sigma_B\left(\chi + \varphi_B\right)\right)$$

Type **`Ihrd1`** (all coefficients in units of amu.Å^2)

$$I(\chi) = I_0 + \sum_{n=1}^{N} I_n \cos\left(n\sigma_I\left(\chi + \varphi_I\right)\right)$$

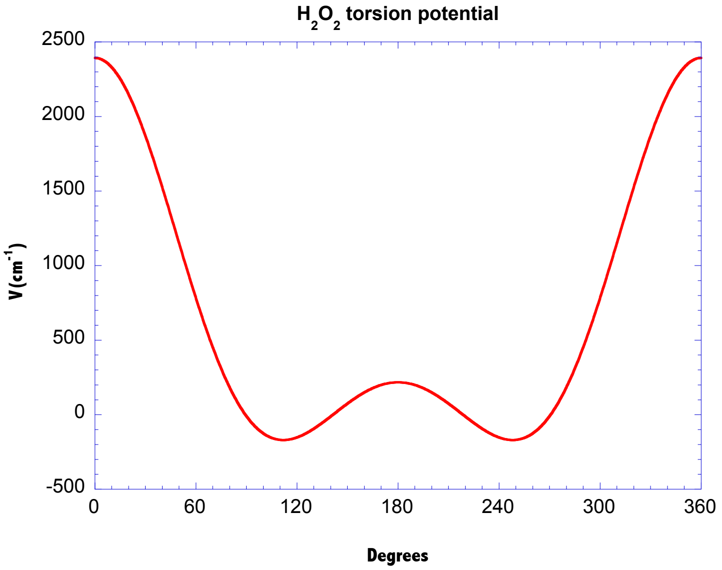
where χ is the dihedral angle (radians), σ_B and σ_I are symmetry numbers (`SYMMM`) and φ_B and φ_I are phase angles (`PHASEM`). *It is VERY IMPORTANT that the angles are defined in the same way both for the potential and for the mass factor.*

"`hrd`" Also Used for Symmetrical Internal Rotors. This degree of freedom type can also be used for symmetrical hindered internal rotations. The rotor symmetry number (`NG`) must be given, as well as coefficients for V(χ), which must have the same symmetry as the rotor symmetry number (the parameter `SYMMV` may equal unity or a convenient multiple of `NG`; it is only required that V(χ) have the correct symmetry). For an example, see the data for $HONO_2$ in the thermodynamics data base (directory `multiwell/thermo-database/`).

Optical Isomers

The classic example of an "optical isomer" (chiral stereoisomer) is methane substituted with four unique atoms or isotopes. There is no internal rotation in this classic example, but internal rotation can also produce optical isomers. For example, hydrogen peroxide has right-handed and left-handed forms, which are identical in all respects except for their handedness. The "gauche" structures of hydrocarbons may also be optical isomers. Consider the case of ethane substituted with one deuterium on each carbon: CH_2D-CH_2D. This molecule has three unique staggered conformers characterized by the D-C-C-D dihedral angle χ: right-handed gauche with $\chi = +60°$, left-handed gauche with $\chi = -60°$, and a conformer with C_s symmetry at $\chi = 180°$. Usually (and maybe always), an internal rotation that connects optical isomers involves an unsymmetrical rotor and therefore **care must be taken to treat the optical isomers and the unsymmetrical hindered internal rotor in a mutually consistent fashion**.

Consider, for example, hydrogen peroxide (H_2O_2), which has an internal rotation around the O-O bond and two non-planar equilibrium structures,[67] as shown below.



This V(χ) has symmetry number $\sigma_v = 1$. Over one full rotation, it has two wells of equal depth, but the two maxima are of unequal height. Furthermore, the two wells are distinguishable because their geometries cannot be superimposed by simple rigid external rotations: they are optical isomers. Essentially, this is a double-well potential.

There are several ways of modeling this double-well potential. The simplest, but least accurate way is to treat this double-well potential is to describe one well as a single harmonic oscillator with harmonic frequency evaluated at the minimum and then multiply the sum of states, density of states, or partition function by ×2, since there are twice as many states in a double-well potential. In other words, the number of optical isomers (parameter m) is equal to the number of wells and the exact partition function for this 1-D unsymmetrical hindered internal rotation is being approximated as follows:

$$q_{hr} = \sum_{i=1}^{\infty} \exp\left(\frac{-E_i}{k_B T}\right) \approx m \cdot q_{HO} \quad \text{and} \quad \rho_{hr}(E) \approx m \cdot \rho_{HO}(E)$$

where the harmonic oscillator partition function q_{HO} (and density of states $\rho_{HO}(E)$) is calculated using the harmonic frequency evaluated at the bottom of the well. Clearly, this approximation is only acceptable when the lowest torsion barrier is >>k_BT.

A second approximate method for treating these optical isomers is to treat the hindered internal rotation as *symmetric* with $\sigma_{hr} = m$ and an assumed hindrance barrier height or reduced moment of moment of inertia I_r. Reasonable estimates for I_r at the equilibrium structure can be obtained,[33,61] but this approximation cannot provide accurate results over the entire range of temperatures, since it is not possible to approximate both barriers (and the angle-dependent reduced moment of inertia) simultaneously. According to this method, the exact partition function is approximated as follows:

$$q_{hr} = \sum_{i=1}^{\infty} \exp\left(\frac{-E_i}{k_B T}\right) \approx m \cdot q_{shr} = m \cdot \left[\frac{1}{m} \sum_{i=1}^{\infty} \exp\left(\frac{-E_{s,i}}{k_B T}\right)\right] = \sum_{i=1}^{\infty} \exp\left(\frac{-E_{s,i}}{k_B T}\right)$$

$$\rho_{hr}(E) \approx m \cdot \rho_{shr}(E)$$

where the "shr" subscript designates the symmetric hindered rotor, E_i are the exact eigenfunctions for the unsymmetrical internal rotation, and $E_{s,i}$ are the eigenfunctions for the symmetric rotor. Although not accurate over the entire temperature range, this approximation performs slightly better than the harmonic oscillator approach at low to intermediate temperatures.

In both of the above approximations, m (the number of optical isomers) is used as a multiplier and must be entered in the MultiWell and Thermo input files.

The most accurate approach is to calculate the eigenvalues for the double-well potential and use them directly in calculating the partition function for the hindered internal rotor. This is done by specifying the "`hrd`" type of degree of freedom, entering the coefficients for the angle-dependent potential energy function and angle-dependent reduced moment of inertia[61] (or rotational constant). For this calculation, m must be set equal to m = 1, since the resulting partition function (and density of states) already has the requisite number of levels and should *not* be multiplied by an additional factor. For example, see the data for HOOH in the thermodynamics data base (directory `multiwell/thermo-database/`).

Of course, not all optical isomers are associated with hindered internal rotation. Those optical isomers that are *not* associated with unsymmetrical hindered rotation should still be included in m and entered in the data files.

Examples

	σ_V: Potential Symmetry	**σ_M: Mass Symmetry**	**σ_{hr}: Rotor Symmetry**	**m: Optical Isomers**	**External Symmetry**
CH_3-CD_3	3	3	3	1	3
CH_3-CD_2H	3	3	3	1	1
CH_3-CDH_2	3	3	3	1	1
CH_3-CH_3	3	3	3	1	6
CH_2D-CH_2D	3	1	1	2	2
CH_2D-CD_2H	3	1	1	2	1
CD_3-CD_3	3	3	3	1	6
HOOH	1	1	1	2	2
CHFClBr	—	—	—	2	1

When CH_2D-CH_2D or HOOH are treated using the "`hrd`" degree of freedom type (unsymmetrical hindered internal rotation), then set internal rotor symmetry to `NG`=1 and the number of optical isomers to `m`=1.

9.6 A Handy List of Lennard-Jones Parameters

Combining rules for Lennard-Jones parameters for A + B collisions:

$$\sigma_{AB} = \frac{1}{2}(\sigma_A + \sigma_B) \qquad \left(\varepsilon / k_B\right)_{AB} = \left[\left(\varepsilon / k_B\right)_A \left(\varepsilon / k_B\right)_B\right]^{1/2}$$

Species	**σ / Å**	**(ε/k$_B$) / K**	**Ref**
He	2.55	10	1
Ne	2.82	32	1
Ar	3.47	114	1
Kr	3.66	178	1
Xe	4.05	230	1
H_2	2.83	60	1
D_2	2.73	69	1
CO	3.70	105	1
N_2	3.74	82	1
NO	3.49	117	1
O_2	3.48	103	1
CO_2	3.94	201	1
N_2O	3.78	249	1
NO_2	4.68	146	4
"	3.46	357	6
H_2O	2.71	506	1
N_2O_5	4.93	380	9
NH_3	2.90	558	2
CH_4	3.79	153	1
C_2H_2	4.13	224	1
C_2H_4	4.23	217	1
C_2H_6	4.39	234	1
C_3H_6	4.78	271	1
c-C_3H_6	4.63	299	1
C_3H_8	4.94	275	1
1-C_4H_8	5.28	302	1
cis-2-C_4H_8	5.27	312	1
n-C_4H_{10}	5.40	307	1
i-C_4H_{10}	5.39	298	1
n-C_5H_{12}	5.85	327	1
neo-C_5H_{12}	5.76	312	1
C_6H_6	5.46	401	1
n-C_6H_{14}	6.25	343	1
c-C_6H_{12}	5.78	394	1
C_7H_8	5.92	410	1
C_7H_{16}	6.65	351	1
C_8H_{18}	7.02	359	1
C_9H_{20}	7.34	362	1
$C_{10}H_{22}$	7.72	363	1
$C_{11}H_{24}$	8.02	362	1
CF_4	4.40	166	1
C_2F_6	5.19	201	1
C_3F_8	5.75	228	1
c-C_4F_8	5.93	252	1

C_4F_{10}	6.30	247	1
C_5F_{12}	6.81	261	1
C_6F_{14}	7.17	265	1
C_7F_{16}	7.78	265	1
C_8F_{18}	7.91	268	1
CF_3Br	4.92	249	1
CF_3Cl	4.79	217	1
CF_2CCl_2	5.08	276	1
$CHClF_2$	4.30	331	1
$CHCl_2F$	4.57	405	1
$CHCl_3$	5.18	378	1
CHF_3	4.04	268	1
CH_2Cl_2	4.54	458	1
CH_3Br	4.31	416	1
CH_3Cl	4.07	373	1
1,1-$C_2H_4Cl_2$	4.85	469	1
1,2-$C_2H_4Cl_2$	4.78	503	1
i-C_3H_7Cl	4.81	435	1
t-C_4H_8Cl	5.23	455	1
CS_2	4.58	415	3
SO_2	4.11	336	4
SF_6	5.20	212	1
$C_4H_4N_2$ (Pyrazine)	5.35	307	5
C_2H_4O (ethene oxide)	4.08	421	1
SiH_4	4.08	208	7
SiH_3	3.94	170	8
SiH_2	3.80	133	8
SiH	3.66	95.8	8
Si	2.91	3036	7
Si_2H_6	4.83	301	8
Si_2H_5	4.72	306.9	8
H_3SiSiH	4.60	312.6	8
H_2SiSiH_2	4.60	312.6	8
Si_2H_3	4.49	318	8
Si_2H_2	4.36	323.8	8
Si_2	3.28	3036	7
Si_3H_8	5.56	331	8
Si_3	3.55	3036	8

References to Lennard-Jones Parameters

1. H. Hippler, J. Troe, and H. J. Wendelken, J. Chem. Phys. 78, 6709 (1983).
2. J. R. Barker and B. M. Toselli, Int. Rev. Phys. Chem., 12, 305 (1993).
3. F. M. Mourits and F. h. A. Rummens, Can. J. Chem., 55, 3007 (1977).
4. J. Troe, J. Chem. Phys., 66, 4758 (1977).
5. T. J. Bevilacqua and R. B. Weisman, J. Chem. Phys. 98, 6316 (1993).
6. R. Patrick and D. M. Golden, Int. J. Chem. Kinetics, 15, 1189-1227 (1983).
7. R. A. Svehla, NASA Report, NASA-TR 132 (1962), cited by Ref. 8.
8. Michael E. Coltrin, Robert J. Kee, and James A. Miller, Electrochem. Soc., 133, 1206-1213 (1986).
9. M. W. Makko and J. Troe, Int. J. Chem. Kinetics, 14, 399-416 (1982).

9.7 Elements and Isotopes Recognized by MomInert and Thermo

The following atomic symbols (atom types) are recognized by . Note that isotopic species have the atomic mass associated with the name, except for Deuterium (D) and Tritium (T), which are given their common symbols. Most atomic masses were taken from the "Table of the relative atomic masses of the elements, 1981" [Pure Appl. Chem. **55**, 1101 (1983)], which is cited by JANAF/NIST.[35] Isotopic masses were taken from N. E. Holden, "Table of the Isotopes (Revised 1998)".[68]

Recognized by MomInert

Species	Atomic Mass	Species	Atomic Mass
H	1.00794	P	30.97376
H1	1.007825	S	32.06
D	2.014102	S32	31.9720707
T	3.016049	S33	32.97145876
He	4.00260	S34	33.9678668
Li	6.941	Cl	35.453
Li6	6.0151228	Cl35	34.96885271
Li7	7.0160046	Cl37	36.9659
Be	9.012182	Ar	39.948
B	10.81	K	39.0983
B10	10.0129	V	50.9415
B11	11.0093	Zn	65.38
C	12.011	Zn64	63.929
C12	12.0	Zn66	65.926
C13	13.003355	Zn68	67.925
C14	14.003242	Se	78.96
C16	16.0147	Se78	77.917
N	14.00674	Se80	79.917
N14	14.003074	Br	79.904
N15	15.00010897	Br79	78.918338
O	15.9994	Br81	80.916291
O16	15.994915	Kr	83.80
O17	16.9991315	Rb	85.4678
O18	17.999160	Sn	118.69
F	18.998403	I	126.9045
Ne	20.179	Te	127.60
Na	22.98977	Xe	131.29
Mg	24.305	Hg	200.59
Mg24	23.985	U	238.0289
Mg25	24.985	U235	235.04393
Mg26	25.983	U238	238.050788
Si	28.0855		
Si29	28.98		
Si30	29.97		

9.8 Eigenstates for large amplitude motions

Energy eigenstates for large amplitude motions are needed for computing densities of states and partition functions. In the MultiWell Program Suite, the eigenstates are computed by solving the Schrödinger equation, as described here.

Theory

The Schrödinger equation for one-dimensional large-amplitude vibration separable from all other motions in the molecule is written as follows[58,59,69,70]

$$\left[-\frac{\hbar^2}{2}\frac{\partial}{\partial q}\frac{1}{I_{eff}}\frac{\partial}{\partial q}+V(q)\right]\psi(q)=E\psi(q) \tag{1}$$

where q is the vibrational coordinate, $\hbar$ is Planck's constant, E and ψ are energy eigenvalue and eigenvector, respectively. $V(q)$ is the vibrational potential energy function, and I_{eff} is the effective mass, which is in general a function of coordinate q, but which, depending on $V(q)$, is sometimes constant. I_{eff} can be derived from the ro-vibrational G matrix, which is defined as:[58,59,71-73]

$$G=\begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}^{-1} \tag{2}$$

where the $\boldsymbol{I}$ matrix is the 3×3 moment of inertia tensor, the $\boldsymbol{Y}$ matrix ($N_{vib}\times N_{vib}$) is the pure vibration contribution, and the $\boldsymbol{X}$ matrix ($3\times N_{vib}$) corresponds to the vibration-rotation interaction (Coriolis) terms. Here, N_{vib} is the number of vibration modes, which equals unity for a 1-D separable degree of freedom. All elements of $\boldsymbol{I}$, $\boldsymbol{X}$, and $\boldsymbol{Y}$ can be computed from the molecular structure as:

$$I_{kk}=\sum_{\alpha=1}^{N_{atom}} m_\alpha\left[\left(\vec{r_\alpha}\cdot\vec{r_\alpha}\right)-\left(r_{\alpha k}\right)^2\right] \text{ ; k = x, y, or z} \tag{3a}$$

$$I_{kk'}=\sum_{\alpha=1}^{N_{atom}} m_\alpha r_{\alpha k} r_{\alpha k'} \text{ ; k} \neq \text{k'} \tag{3b}$$

$$X_{ij}=\sum_{\alpha=1}^{N_{atom}} m_\alpha\left[\vec{r_\alpha}\times\frac{\partial\vec{r_\alpha}}{\partial q_j}\right]_i \tag{4}$$

$$Y_{ij}=\sum_{\alpha=1}^{N_{atom}} m_\alpha\left(\frac{\partial\vec{r_\alpha}}{\partial q_i}\right)\bullet\left(\frac{\partial\vec{r_\alpha}}{\partial q_j}\right) \tag{5}$$

where the α index runs on the number of atoms (N_{atom}) in the molecule.

For one-dimensional large-amplitude **vibrations**, the ro-vibrational G matrix is expressed as:

$$G = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} & X_{11} \\ -I_{yx} & I_{yy} & -I_{yz} & X_{12} \\ -I_{zx} & -I_{zy} & I_{zz} & X_{13} \\ X_{11} & X_{12} & X_{13} & Y_{11} \end{pmatrix}^{-1} \tag{6}$$

By inverting the right hand side of eq. (6), one obtains:

$$G = \begin{pmatrix} g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \\ g_{41} & g_{42} & g_{43} & g_{44} \end{pmatrix} \tag{7}$$

Finally, I_{eff} is obtained as: $I_{eff} = g_{44}^{-1}$.

Note that in this work, central finite differences are used to compute the derivatives in Eqs. (4) and (5). Thus the steps between adjacent positions (e.g. dihedral angles) must be small enough so that the derivatives are sufficiently accurate. Acceptable step-sizes must be found by trial and error.

Hindered internal rotations

For hindered internal rotations, the large-amplitude motions are torsional changes in dihedral angles. The Schrödinger equation (1) for a 1-D torsion can be rewritten as follows[70]:

$$\left[-\frac{\hbar^2}{2}\frac{\partial}{\partial\chi}\frac{1}{I_{eff}(\chi)}\frac{\partial}{\partial\chi}+V(\chi)\right]\psi(\chi)=E\psi(\chi). \tag{8a}$$

By defining the rotational constant $B_{hr} = \frac{\hbar^2}{2I_{eff}}$, we obtain:

$$\left[-\frac{\partial}{\partial\chi}B_{hr}(\chi)\frac{\partial}{\partial\chi}+V(\chi)\right]\psi(\chi)=E\psi(\chi) \tag{8b}$$

where χ is the torsional (dihedral) angle ($0 \le \chi < 2\pi$), $V(\chi)$ is the torsional potential energy function, and I_{eff} is the effective reduced moment of inertia. Both I_{eff} and B_{hr} are in general functions of dihedral angle, but are constants for a rigid rotor. When B_{hr} is assumed to be a constant, which is realistic only for symmetrical rotors, equation (8b) simplifies:

$$\left[-B_{hr}\frac{\partial^2}{\partial\chi^2}+V(\chi)\right]\psi(\chi)=E\psi(\chi) \tag{9}$$

Solutions of equation (9) are not given here, but are well known.[70,74,75]

For the purpose of calculating densities of states and computing partition functions in the MultiWell Suite, Eq. (8b) is diagonalized to obtain energy eigenvalues, whether or not B_{hr} is assumed to be constant. The solutions are obtained by using the method described by Meyer.[70] The Hamiltonian matrix of Eq. (8b) is given by:

$$\mathbf{H} = -\mathbf{D}B_{hr}\mathbf{D} + \mathbf{V} = \mathbf{D}^T B_{hr}\mathbf{D} + \mathbf{V} \tag{10}$$

where **D** is the matrix of the first-order derivative ($\partial/\partial\chi$) operator of the internal rotation angle, $\mathbf{D}^T$ is the transpose of **D** (i.e. $D^T(i,j) = D(j,i)$), **V**(i,i) is the diagonal matrix of the potential energy operator, and B_{hr}(i,i) is the diagonal matrix element of the rotational constant; the indices are for equally spaced torsion angle grid points. For 2N +1 grid points, the elements of **D** are:

$$D\left(i,i\right) = 0 \tag{11a}$$

$$D(i,j) = \left(-1\right)^{i-j}\left\{2\sin\left[\left(i-j\right)\pi/\left(2N-1\right)\right]\right\}^{-1} \quad \text{for } i{\neq}j \tag{11b}$$

To construct the symmetric matrix **H**, one requires both $V(\chi)$ and $B_{hr}(\chi)$. Several common representations of $V(\chi)$ and $B(\chi)$ (or the corresponding moment of inertia function $I(\chi)$) can be understood by MultiWell, as explained elsewhere in this manual. The matrix **H** is diagonalized in order to obtain a vector of energy eigenvalues, which are convoluted with states from the other degrees of freedom to compute ro-vibrational densities of states or partition functions.

Users must supply the functions $V(\chi)$ and $B_{hr}(\chi)$ (or moment of inertia function $I(\chi)$). Potential energies $V(\chi)$ and molecular geometries can be computed at discrete values of χ by obtained by using any of the many available quantum chemistry codes, such as Gaussian,[55] C-Four,[56], and Molpro.[60]. The results for $V(\chi)$ can be fitted to a suitable truncated Fourier series, and one may use codes like the I_Eckart program[61] (written for use with MatLab) or LAMM program given in the MultiWell Suite to compute $B_{hr}(\chi)$ or $I_{hr}(\chi)$.

9.9 Semi-Classical Transition State Theory

Much of this section was adapted from Ref. [7].

Program `sctst` uses a new algorithm[7] developed for computing rate constants with semi-classical transition state theory (SCTST), which was formulated by Miller and coworkers.[12-15] SCTST incorporates non-separable coupling among all degrees of freedom and multi-dimensional quantum mechanical tunneling along the curved reaction path. The algorithm, which is practical for reactants containing dozens of atoms, predicts both microcanonical and canonical rate constants. In addition, the quantum chemistry code CFOUR has been extended to efficiently compute fully coupled vibrational anharmonicities for transition states at the CCSD(T) level of theory.

Semi-classical transition state theory (SCTST), has been described by W. H. Miller and coworkers.[12-15] In combination with second-order vibrational perturbation theory (VPT2),[50] the SCTST accounts explicitly for non-separable coupling among all degrees of freedom (including the reaction coordinate) in the transition state region and multidimensional semi-classical tunneling.[14,15] Miller, Handy and coworkers have shown that the SCTST thermal rate constants are in close agreement with those of full quantum dynamic calculations and experiments.[14,76,77] However, the SCTST method has been limited to small reaction systems having only 3-4 atoms because of the large number of quantum states that must be computed for larger systems.[14,76,77]

The new computer code `sctst` makes it practical to apply the theory to polyatomic reactions. The new algorithm is based on an extension of methods for computing sums and densities of states for fully-coupled anharmonic vibrations, which were described recently.[6] In addition, an extension of the CFOUR electronic structure code [56] is described for computing the anharmonicity constants of transition states at the CCSD(T) level of theory. This new extension has enabled rate constant predictions based on a high level of theory.[7]

Theory

According to the SCTST of W. H. Miller and coworkers,[14] the microcanonical and canonical rate constants are written:

$$k(E) = \frac{1}{h}\frac{G^{\ddagger}(E)}{\rho(E)} \tag{1}$$

$$k(T) = \frac{1}{h}\frac{\int_{-\infty}^{+\infty} G^{\ddagger}(E)\exp(-E/k_BT)dE}{Q_{re}(T)} \tag{2a}$$

where h is Planck's constant, k_B is Boltzmann's constant, T is the temperature, $\rho(E)$ is the density of states of the reactant, Q_{re} is the total partition function of the reactant(s), and $G^{\ddagger}(E)$ is the cumulative reaction probability (CRP).[12]. The center of mass translations are rigorously separable. At temperatures that are not too high, it is a good approximation to treat the overall rotations as separable from the vibrations (including internal rotations), so that Eq. (2a) can be written:

$$k(T) = \frac{1}{h}\frac{Q_t^{\ddagger}Q_r^{\ddagger}}{Q_tQ_r}\frac{\int_{-\infty}^{+\infty} G_v^{\ddagger}(E_v)\exp(-E_v/k_BT)dE_v}{Q_v(T)} \tag{2b}$$

in which vibrational energy E_v is the variable of integration and the vibrational CRP is given by:

$$G_v^{\ddagger}(E_v) = \sum_{n_1=0}\sum_{n_2=0}\cdots\sum_{n_{F-2}=0}\sum_{n_{F-1}=0} P_n(E_v) \tag{3}$$

In Eq. 3, F-1 is the total number of vibrational degrees of freedom orthogonal to the reaction path. The semi-classical tunneling probability P_n is given by[14]

$$P_n(E) = \frac{1}{1+\exp[2\theta(n,E)]} \tag{4}$$

where

$$\theta(n,E) = \frac{\pi\Delta E}{\Omega_F}\frac{2}{1+\sqrt{1+4x_{FF}\Delta E/\Omega_F{}^2}} \tag{5}$$

$$\Delta E = \Delta V_o + \varepsilon_o - E + \sum_{k=1}^{F-1}\omega_k\left(n_k+\frac{1}{2}\right) + \sum_{i=1}^{F-1}\sum_{j=i}^{F-1}x_{ij}\left(n_i+\frac{1}{2}\right)\left(n_j+\frac{1}{2}\right) \tag{6a}$$

$$\Omega_F = \overline{\omega}_F - \sum_{k=1}^{F-1} \overline{x}_{kF} \left(n_k + \frac{1}{2} \right) \qquad \text{with } \overline{\omega}_F = -i\omega_F \text{ and } \overline{x}_{kF} = ix_{kF} \tag{7}$$

In these expressions, ω_k is the harmonic vibrational frequency, ω_F is the imaginary frequency of the reaction barrier, x_{ij} are the anharmonicity constants that describe coupling among the orthogonal degrees of freedom, x_{kF} are the (pure imaginary) coupling terms between the reaction coordinate and the orthogonal degrees of freedom, and x_{FF} is the real-valued anharmonicity constant for the reaction path, and ΔV_o is the classical barrier height. The term ε_0 is a constant, which is irrelevant in spectroscopic measurements (where only energy differences are important), but which must be included for thermochemistry.[51,76]

For convenience, the reaction barrier height is often expressed as the difference between the zero point energy of the transition state and that of the reactant(s) (i.e. the vibrationally adiabatic ground-state potential energy difference $\Delta V_a^{G\ddagger}$), giving the following expressions:

$$k(T) = \frac{1}{h} \frac{Q_t^\ddagger Q_r^\ddagger}{Q_t Q_r} \frac{\int_{-\infty}^{+\infty} G_v^\ddagger(E_v) \exp(-E_v / k_B T)\, dE_v}{Q_v(T)} \tag{2c}$$

$$\Delta E = \Delta V_a^{G\ddagger} - E + E_v^\ddagger \tag{6b}$$

$$E_v^\ddagger = \sum_{k=1}^{F-1} \omega_k \left(n_k + \frac{1}{2} \right) + \sum_{i=1}^{F-1} \sum_{j=i}^{F-1} x_{ij} \left(n_i + \frac{1}{2} \right) \left(n_j + \frac{1}{2} \right) - \frac{1}{2} \sum_{k=1}^{F-1} \omega_k - \frac{1}{4} \sum_{i=1}^{F-1} \sum_{j=i}^{F-1} x_{ij} \tag{8}$$

and ε_o is incorporated into the zero-point vibrational energy of the orthogonal modes:

$$E_{zpe} = \varepsilon_0 + \frac{1}{2} \sum_{k=1}^{F-1} \omega_k + \frac{1}{4} \sum_{i=1}^{F-1} \sum_{j=i}^{F-1} x_{ij} \tag{9}$$

In the next section, the new algorithm for computing the CRP is presented. For canonical (thermal) systems, Hernandez and Miller have described an efficient method for computing the rate constants directly from SCTST without the necessity for computing the CRP.[15] A small modification of the new algorithm presented below can be used for this purpose, but will not be described here because of our interest is in computing k(E), the microcanonical rate constant, which requires the use of the CRP. However, with the CRP in hand, thermal rate constants can easily be obtained by averaging k(E) over the thermal energy distribution function. With minor extensions, the methods described here are also appropriate for computing k(E,J), the microcanonical rate constant for a specified total angular momentum quantum number J.

New Algorithm

In Eqs. (1) and (2c), the CRP is obtained by summing over all states of the transition state. When the state energies can be estimated from harmonic frequencies and anharmonicity constants, the summations in Eq. 3 can be evaluated exactly. In this "exact count" method, a realization of the vector $\{n_k\}$ of the vibrational quantum numbers is first determined for every quantum state of the transition state and the CRP is obtained *via* nested DO-loops using Eqs. (3)-(8). Such calculations are quite fast for systems with 3-4 atoms, but they become impossibly slow for systems containing more atoms. The same obstacle is encountered when computing densities of states for coupled vibrational states.[6]

Recently, three practical methods for computing vibrational sums and densities of states were compared.[6] These included the brute force direct method, an "efficient" Monte Carlo method,[47] and an extended version of the algorithm published by Basire et al.,[49] which is based on the Wang-Landau algorithm.[48] The extended algorithm is particularly useful for large molecular systems. As explained elsewhere,[6] all three of the algorithms are based on a perturbation theory expansion of the state energies expressed as functions of the vibrational quantum numbers. (These algorithms are implemented in programs `doloops`, `ansum`, and `adensum`.) A logical extension of these algorithms allows computation of the CRP.

According to SCTST, only Ω_F in Eq. (7) and $E_v^{\ddagger}$ in Eq. (8) depend on the vector $\{n_k\}$ of vibrational quantum numbers; all of the other quantities are constants. In the implementation of the Wang-Landau algorithm,[6] the energy axis is divided into many small grains (i.e. energy bins, e.g. with δE=10 cm^{-1}). The algorithm is a random walk procedure in which each step is accepted or rejected, based on certain criteria. For each step that is accepted, a vector of quantum numbers has been generated, which is accepted as long as each quantum number is not greater than the highest bound state in each degree of freedom. This condition is evaluated by computing partial derivatives of the state energy with respect to the quantum numbers, as described by Nguyen and Barker [6]. Acceptable states include the metastable states, which have total energies above the dissociation energy (or reaction energy barrier), but do not have enough energy localized in the reaction coordinate to react. These are the vibrational states of the transition state.

The density of states algorithm is carried out iteratively. During each iteration, each energy bin that contains a vibrational state is sampled a large number of times. The relative density of states estimate is up-dated on the fly and, after a series of iterations, it converges to the exact relative density of states. After the relative density of states is normalized, the algorithm terminates (see Ref. [6] for a detailed description). The CRP can be calculated by using the vibrational density of states $\rho_v^{\ddagger}(E)$ (which may have been computed previously) [6] and by carrying out a single iterative cycle with the same stepping algorithm used for the density of states calculation. On every accepted step into an energy bin, Ω_F and $E_v^{\ddagger}$ are computed using the vector of quantum numbers $\{n_k\}$. From these quantities, average values and the corresponding average probability are obtained by using Eqs. 4-7 for the i^{th} energy bin at energy E_i: $<\Omega_F>_{\mathrm{i}} = \Omega_{F\mathrm{i}}/H_i$, $< E_v^{\ddagger}>_{\mathrm{i}} = E_v^{\ddagger}{}_{\mathrm{i}}/H_i$, and average probability $<P(E_i)>$, where H_i is the number of samples in the i^{th} bin. The averages become essentially independent of the number of samples for a sufficiently large number (typically 10^4/bin). The algorithm accumulates samples in all of the energy bins where acceptable states exist. If no acceptable states exist in a bin, then $<P(E_i)>=0$ in that bin. Finally, the CRP is obtained by using the average probability multiplied by the number of states in each energy bin (i.e. $\rho(E_i)\delta E$):

$$G_v^{\ddagger}(E_v) = \delta E \sum_{i=1}^{E_v/\delta E} \rho\left(E_i\right)\left\langle P\left(E_i\right)\right\rangle \tag{10}$$

In the published `adensum` computer program for computing sums and densities of states,[78] the number of samples in each energy bin is conveniently selected by specifying a key word: *FAIR*, *GOOD*, *BETTER*, *BEST*, or *EXTRA* specifies 10^2, 10^3, 10^4, 10^5, or 10^6, samples per bin, respectively. These key words are also used for specifying the number of samples used to compute the average quantities defined above for the SCTST.

The accuracy of this new random walk algorithm was tested by comparing results obtained using the new algorithm with those obtained from the exact count method (i.e. the nested DO-loops method) for the OH + H_2 reaction. For this reaction, the agreement between the two approaches is excellent, as shown in Figure 1, when ≥10^4 samples/bin are used for computing the averages. The differences between the "exact" CRP (obtained using explicit nested DO-loops) and the ones based on the new algorithm are less than 5% even at the highest energy (50000 cm^{-1}), resulting in differences of <0.5% between the vibrational partition functions at 2500 K for the transition state.

Program `sctst` (Section 7.4) generates a supplementary data filed named `<name>.qcrp`, which is required when using program Thermo to compute thermal rate constants using the SCTST. This data file contains the partition function corresponding to the CRP.

$$Q_v^{\ddagger}\left(T\right) = \int_0^{+\infty} \rho_v^{\ddagger}(E_v)\exp(-E_v / k_B T)\, dE_v = \left(k_B T\right)^{-1} \int_0^{+\infty} G_v^{\ddagger}(E_v)\exp(-E_v / k_B T)\, dE_v \qquad (11)$$

where $\rho^{\ddagger}{}_v(E_v)$ is the derivative of $G^{\ddagger}{}_v(E_v)$ with respect to E_v. The integral on the right hand side of Eq. 11 is needed for Eq. 2c. The expression for $Q^{\ddagger}{}_v(T)$ is more convenient for use with program `Thermo`.

VPT2/CCSD(T) for Transition States

All of the harmonic vibrational frequencies and x_{ij} anharmonicity coefficients can be obtained for stable species from first principles by using second order vibrational perturbation theory (VPT2), which has been implemented for stable molecules in quantum chemical software packages such as GAUSSIAN [55] and CFOUR [56]. The anharmonicity coefficients are obtained from the full cubic force field together with the semi-diagonal quartic force field, which can be computed by finite differences from the analytic first and second derivatives [56,79]. For highly accurate methods such as CCSD(T), analytic second derivatives are only available in CFOUR [56]. For this work, VPT2 computation of the anharmonicities was implemented for transition states at the CCSD(T) level of theory in CFOUR. Whenever Fermi resonances are detected, deperturbation was applied using a criterion of 100 cm^{-1} for the frequency difference. The combination of the new algorithm for the SCTST and implementation of VPT2 for transition states constitutes an effective tool for computing the CRP and/or thermal rate constants from first principles with high accuracy.

Reaction Critical Energy

In the absence of quantum mechanical tunneling, the critical energy E_0 for a unimolecular reaction (or a bimolecular reaction according to microcanonical TST) is the difference between the enthalpies at 0 K of the transition state and the reactant(s):

$$E_0 = \Delta_f H_{TS}(0) - \Delta_f H_{reac}(0) \qquad (12)$$

This definition of the reaction critical energy is **not** appropriate when quantum mechanical tunneling occurs, because tunneling takes place at lower energies. Tunneling occurs whenever the energy is below the top of the barrier (i.e. $E < E_0$, where E_0 is defined as above) but above that of the zero point energy of the product. Thus when tunneling is included the appropriate value to be utilized for "E_0" depends on whether the reaction is endothermic or exothermic (including zero point energy corrections), regardless of the reaction barrier height.

In the MultiWell master equation code, the critical energy is defined as in Eq. 12 even when tunneling through an Eckart barrier is invoked (keyword '`tun`' on Line 14 in the `multiwell.dat` data file), because the reactant and product enthalpies are employed automatically to account for tunneling below the top of the barrier. When using an external file containing the CRP generated by Program `sctst`, this step is NOT automated and the user must enter the value for critical energy obtained from the following:

Endothermic reactions: $E_0 = \Delta_f H_{prod}(0) - \Delta_f H_{reac}(0)$ (13a)

Exothermic reactions: $E_0 = 0$ (13b)

(Note: the keyword '`tun`' should NOT be employed when using the CRP, since it only applies to tunneling through an Eckart barrier.)

9.10 Legacy and Current Versions of k_0 and k_∞ in MultiWell

Early in 2011 it was discovered that the centrifugal corrections had not been implemented correctly. This affected the "informational" values if k_0 and k_∞ tabulated in the MultiWell output files. This section summarizes the actual implementations. See Appendix Section 3.2 of this User Manual for a summary of the theory.

Legacy Basis

Legacy expressions for k_{uni}, k_∞ and k_0, based on Robinson & Holbrook Eq. 4.30. In the following, $\omega = k_2[M]$ (in the notation of Robinson & Holbrook).

$$k_{uni} = \frac{Q_1^+}{Q_1}\frac{1}{Q_2}\exp\left(-E_0/kT\right)\omega\int_0^\infty \frac{\rho\left(E^+ + E_0 + \left\langle \Delta E_J \right\rangle\right)k_a\left(E^+ + E_0 + \left\langle \Delta E_J \right\rangle\right)e^{-E^+/kT}dE^+}{\omega + k_a\left(E^+ + E_0 + \left\langle \Delta E_J \right\rangle\right)}$$

where

$$k_a\left(E^+ + E_0 + \left\langle \Delta E_J \right\rangle\right) = \frac{L^\ddagger}{h}\frac{G\left(E^+\right)}{\rho\left(E^+ + E_0 + \left\langle \Delta E_J \right\rangle\right)}$$

$$\left\langle \Delta E_J \right\rangle = \left(1 - \frac{I^+}{I}\right)kT$$

$G(E^+)$ is the sum of states starting from the ZPE of the TS; $L^\ddagger$ is the reaction path degeneracy.

The limits:

$$k_\infty = \frac{Q_1^+}{Q_1}\frac{1}{Q_2}\exp\left(-E_0/kT\right)\int_0^\infty \rho\left(E^+ + E_0 + \left\langle \Delta E_J \right\rangle\right)k_a\left(E^+ + E_0 + \left\langle \Delta E_J \right\rangle\right)e^{-E^+/kT}\,dE^+$$

$$k_0 = \frac{Q_1^+}{Q_1}\frac{1}{Q_2}\exp\left(-E_0/kT\right)k_{LJ}\int_0^\infty \rho\left(E^+ + E_0 + \left\langle \Delta E_J \right\rangle\right)e^{-E^+/kT}\,dE^+$$

As actually implemented in the Codes:

Multiwell-2011

CENT

$$k_{\infty} = \frac{I^{+}}{I}\frac{1}{Q_2}\exp\left(-E_0/kT\right)\int_0^{\infty}\rho\left(E^{+}+E_0+\left\langle\Delta E_J\right\rangle\right)k_a\left(E^{+}+E_0+\left\langle\Delta E_J\right\rangle\right)e^{-E^{+}/kT}\,dE^{+}$$

$$k_0 = \frac{I^{+}}{I}\frac{1}{Q_2}\exp\left(-E_0/kT\right)k_{LJ}\int_0^{\infty}\rho\left(E^{+}+E_0+\left\langle\Delta E_J\right\rangle\right)e^{-E^{+}/kT}\,dE^{+}$$

NOCENT

$$k_{\infty} = \frac{I^{+}}{I}\frac{1}{Q_2}\exp\left(-E_0/kT\right)\int_0^{\infty}\rho\left(E^{+}+E_0\right)k_a\left(E^{+}+E_0\right)e^{-E^{+}/kT}\,dE^{+}$$

$$k_0 = \frac{I^{+}}{I}\frac{1}{Q_2}\exp\left(-E_0/kT\right)k_{LJ}\int_0^{\infty}\rho\left(E^{+}+E_0\right)e^{-E^{+}/kT}\,dE^{+}$$

For this case, $E' = E_0 + E^{+}$. Therefore:

$$k_{\infty} = \frac{I^{+}}{I}\frac{1}{Q_2}\int_{E_0}^{\infty}\rho\left(E'\right)k\left(E'\right)e^{-E'/kT}\,dE'$$

$$k_0 = \frac{I^{+}}{I}\frac{1}{Q_2}k_{LJ}\int_{E_0}^{\infty}\rho\left(E'\right)e^{-E'/kT}\,dE'$$

Multiwell-2011.1 and later versions

Note: x = variable of integration; may be E^{+} or E'.

CENT1, CENT2, and CENT3

$$k_a'\left(E'\right) = \frac{I^{+}}{I}\exp\left(\left\langle\Delta E_J\right\rangle/kT\right)k_a\left(E'\right)$$

$$k_{\infty} = \frac{1}{Q_2}\int_{E_0+\left\langle\Delta E_J\right\rangle}^{\infty}\rho\left(E'\right)k_a'\left(E'\right)e^{-E'/kT}\,dE'$$

$$k_0 = \frac{1}{Q_2}k_{LJ}\int_{E_0+\left\langle\Delta E_J\right\rangle}^{\infty}\rho\left(E'\right)e^{-E'/kT}\,dE'$$

CENTX

Here, $x = E^{+}$.

$$k_{\infty} = \frac{I^{+}}{I}\frac{1}{Q_2}\int_0^{\infty}\rho\left(x + E_0 + \left\langle \Delta E_J \right\rangle\right)k_a\left(x + E_0 + \left\langle \Delta E_J \right\rangle\right)e^{-\left(x+E_0\right)/kT}\,dx$$

$$k_0 = \frac{I^{+}}{I}\frac{1}{Q_2}k_{LJ}\int_0^{\infty}\rho\left(x + E_0 + \left\langle \Delta E_J \right\rangle\right)e^{-\left(x+E_0\right)/kT}\,dx$$

NOCENT

$$k_{\infty} = \frac{1}{Q_2}\int_{E_0}^{\infty}\rho\left(E'\right)k\left(E'\right)e^{-E'/kT}\,dE'$$

$$k_0 = \frac{1}{Q_2}k_{LJ}\int_{E_0}^{\infty}\rho\left(E'\right)e^{-E'/kT}\,dE'$$

These expressions differ from v.2011 by elimination of the factor I^{+}/I.

10. How to …

This section outlines how to carry out various tasks needed by many users. It is under development and more sections will be added in future releases of this User Manual.

How does one set up the double arrays in MultiWell?

The double arrays consist of two arrays in one. The lower energy portion ranges in energy from E = 0 up to `Emax1`; the upper energy portion ranges in energy from E = 0 up to `Emax2`. The total size of the double array is `Isize` and the number of "energy grains" (i.e. energy bins) in the lower and upper parts of the double array are `Imax1` and `Imax2`, respectively, and the respective energy grain sizes are `Egrain1` and `Egrain2`. The relationships among these quantities are as follows:

Emax1 = (Imax1-1)*Egrain1

Emax2 = (Imax2-1)*Egrain2

Isize = Imax1 + Imax2

To set up double arrays, users must specify `Egrain1`, `Imax1`, `Emax2`, and `Isize`; the remaining parameters are determined by the program from the relationships given above. The following guidelines are useful for deciding on the values for the parameters.

`Egrain1`

- Should be small enough so that further reductions in its size do not significantly affect the numerical results (i.e. it should achieve numerical convergence).
- Should not be too small, since that adds a computational burden.
- Typical choices: 5 or 10 cm^{-1} (usually 10 cm^{-1}).

`Imax1`

- Should be large enough so that the density of states in adjacent energy grains near `Emax1` differ by less than 5%. DenSum will give a warning if the grain-to-grain fluctuations are too large.
- Typically, Imax should be large enough so that `Emax1` is greater than the highest vibrational wavenumber (typically greater than ~3000 cm^{-1}).
- Typical choice: 400 (for `Egrain1` = 10 cm^{-1}).

`Emax2`

- Should be high enough so that none of the stochastic trials attempt to reach a higher energy.
- Typical choices: 50000 cm^{-1} for simulations at 300 K and 85000 cm^{-1} for temperatures up to 3000 K.

`Isize`

- Should be chosen large enough so that `Egrain2` $\leq k_BT/2$.
- Typical choices: 1000 for simulations at 300 K, 1500 for very high temperatures.

How does one test the double-array parameters in MultiWell?

In principle, it is wise to test your choices of parameters to determine whether the calculations are reaching numerical convergence. In other words, one should determine whether the choices of `Egrain1` and `Egrain2` are small enough and that `Emax1` and `Emax2` are high enough.

The tests are carried out on a small subset of representative simulations with pressures and temperatures at the middle and at the extremes of their ranges, for a total of nine simulations. For narrow pressure ranges and temperature ranges, only one representative set of conditions may be necessary. Thus the amount of testing depends on the system and on experience with similar simulations.

Numerical convergence has been achieved with the grain sizes are small enough and the maximum energies are high enough so that further grain size reductions and/or further maximum energy increases do not result in significant differences in the calculations.

The key quantities affected by grain size and maximum energy are `kinf` and `kosc` (i.e. the high pressure limit k_∞ and the low pressure limit k_0), which are printed by `MultiWell` for information purposes in the table summarizing transition state and reaction properties (preceding the table summarizing the stochastic simulations). These rate constants are not actually used by `MultiWell` in further calculations, but they are useful to users. The value for k_∞ should agree within a few percent with the corresponding canonical rate constant calculated by `Thermo` with the same input parameters.

How can one estimate Lennard-Jones and energy transfer parameters?

It is very difficult to compute proper energy transfer parameters. Instead, they are usually obtained by fitting experimental rate constant data. There are two parts to the problem: obtaining Lennard-Jones (LJ) parameters and estimating energy transfer parameter α for the exponential-down model, which is recommended for most purposes.

The LJ parameters can be estimated by using the critical properties of the compound, which may in turn be estimated using various empirical correlations. However, it is almost always sufficiently accurate to estimate the parameters, based on analogy. A table of LJ parameters can be found in Section 9.6 of this User Manual.

Experience shows that parameter α in the exponential model (i.e. a single component of the bi-exponential model) usually is in the range from 100 cm^{-1} to 1000 cm^{-1}; 250 cm^{-1} is a good first guess.

Experimental rate constant data can be fitted by varying the value of parameter α until a good fit has been achieved. It is efficient to start by computing results for several values of α that cover the whole range expected (typically 100, 250, 500, and 1000 cm^{-1}). For most molecular species, the "true" value is probably somewhere in this range. By examining a plot of the data superimposed on a plot of this family of simulations, it will be possible to get a better estimate of the fitted value. The plot will also demonstrate the sensitivity of the simulations to the assumed value.

How does one calculate a thermal unimolecular rate constant?

MultiWell does not calculate conventional unimolecular reaction rate constants directly, because it does not assume that the energy distributions are at steady state. To find k_{uni} in the pressure fall-off regime, you need to carry out simulations at the temperature and pressure of interest. Then, you can find k_{uni} in one of two ways (outlined below) which both depend on the population distributions reaching steady-state. The population distributions will approach steady-state values if the reaction rates are slower than the energy relaxation rates. Here are the two methods:

a) the output file 'multiwell.rate' gives the instantaneous average rate constants for all the reactions, based on the instantaneous population distributions and on k(E) for each reaction. AT STEADY STATE, the instantaneous average rate constant is equal to k_{uni}. To use this feature, however, requires many, many stochastic trials in a single simulation in order to reduce the stochastic sampling noise.

b) AT STEADY STATE, the decay of population in a SINGLE-CHANNEL reaction depends on time and on k_{uni}, which is obtained from a semi-log plot of population ("fraction") as a function of time. The slope of the plot is slope = $-k_{uni}$.

It is important to note that if the unimolecular decay requires more than a few thousand collisions, the simulations become very time-consuming. MultiWell is not designed for computing slow, steady-state unimolecular reactions.

How can one determine recombination rate constants?

In MultiWell, the chemical identity of each chemical species (`MOL`) is independent of its internal energy. Thus, molecules are assumed to have unique chemical identities even when they have energies in excess of reaction barriers and while they are undergoing activation and deactivation. This assumption mimics nature.

In the case of a simple ***recombination reaction***, one can use MultiWell to simulate the reaction over a time interval, typically a few hundred or a few thousand collisions.

$$A + B \rightleftharpoons C$$

The simulation is initialized with the highly excited reaction product, C. The initial energy distribution of C is the chemical activation distribution (keyword `CHEMACT`). In the simulation, the initial average vibrational energy of C is the exothermicity of the reaction, augmented by thermal energy.

Over the course of time, the population of C is affected simultaneously by both collisional deactivation and bond fission back to the initial reactants. Thus the energy of C decreases monotonically until it reaches steady state. At the same time, the yield of C (its "fraction") asymptotically decreases toward a constant value. When the energy of C is too low for it to react, its fraction no longer varies with time.

The recombination rate constant is obtained by taking the time independent final fraction of C (i.e. f_C) and multiplying by the high pressure limit for the bimolecular recombination rate coefficient:

$$k_{rec} = k_{rec}(\infty)^* f_C$$

If one runs a much longer simulation, one will often find that product C hasn't really become constant, but is decaying with a slower time constant. If this decay is slow enough, it can be neglected when assessing the much faster chemical activation process. This reliance on separation of time scales is typically a good method as long as the time scales for the recombination and unimolecular decomposition differ by an order of magnitude or more.

Note that the rate constant for ***unimolecular reaction*** can be found by using the relationship between the rate constants and the equilibrium constant:

$$K_{eq} = k_{rec} / k_{uni}$$

The equilibrium constant can be computed using Thermo. Furthermore, the high pressure limit rate constants in both directions can also be computed using Thermo.

For ***chemical activation*** reactions, the procedures are similar to those for recombination reactions.

$A + B \rightleftharpoons C \rightarrow D$

In a chemical activation reaction, the excited intermediate C initially has enough energy to react and produce product D. The yield ("fraction") of D depends on the competition between collisional deactivation and unimolecular reactions involving the intermediate C. The effective chemical activation rate constant describing the production of D is obtained from the yield of D and the high pressure limit of the rate constant for recombination of A + B:

$k_D = k_{rec}(\infty)*f_D$

How can one tell if the simulated time is long enough?

Pragmatically, one should run the simulations long enough to determine if two criteria are satisfied:

a) The average energy of a product molecule (C) is well below any of its reaction thresholds (also remember that the distribution of internal energy extends well above the average)

b) The yield of C is independent of time. It is best to carry out some preliminary calculations (using a smaller number of trials) to determine the time duration (or number of collisions) that will be needed for the excited species to be deactivated sufficiently.

11. Questions and Answers

1. QUESTION: Every molecule has a 2 dimensional external inactive rotor and an external active rotor (the K-rotor). I can see where the external inactive rotor goes but I'm not sure about the external active rotor.

ANSWER: The parameters "MolMom" on line #8 and "RR" on line #15 of the data file are moments of inertia for the 2-dimensional external (inactive) rotors for the reactant and the transitions state, respectively. The external active rotor is included with the other degrees of freedom in the density and sum of states calculation. Therefore it is included in the data file for SumDen. For example, O_3 (ozone) has 3 vibrations and 3 rotations. The SumDen data file should include 4 degrees of freedom: 3 vibrations + 1 rotor (the active external rotation, which I usually label the "K-rotor"). The moment of inertia for the remaining 2-D external (inactive) rotor appears on line #15.

2. QUESTION: If there is an internal active rotor, do I remove the vibration associated with it from the Densum input?

ANSWER: An internal degree of freedom is either a rotor, or a vibration...not both. Thus, if you have a vibrational assignment (from Gaussian, say), you may wish to replace a low frequency vibrational mode with a free rotor mode. Then you remove the vibration and insert the rotor in its place, so that the number of internal degrees of freedom is preserved.

3. QUESTION: In all your examples you use a spacing of 10 cm^{-1} and when I use 1 cm^{-1} the code freezes. Do you recommend using 1 cm^{-1} and if so should I increase the size of the array which is currently 14000?

ANSWER: I recommend the use of 10 cm^{-1}, because we have found that it gives excellent numerical convergence at room temperature and above, and because few thermochemical values are known to better accuracy. In previous work, I routinely used 25 cm^{-1} spacing and that also worked very well. If you want to use a smaller grain size, then the arrays should be increased in size. In principle, one should always test to make sure that the grain size is small enough so that it does not affect the results significantly.

4. QUESTION: In using DenSum, when should one use quantum state counting for rotors ("qro") and when should one use classical state densities ("rot")?

ANSWER: As a rough rule of thumb, if the rotational constant is greater than B = 1 cm^{-1}, use "qro"; if it is less than 0.1 cm^{-1}, use "rot"; in between, you can try it both ways and then decide if the difference would affect your calculations. If you use "qro" for a case where the rotational constant is extremely small, there may be computational problems...this limit has not been tested.

5. QUESTION: What will happen if the density of states function is not a smooth, continuous function at energies corresponding to the upper half of the double array?

ANSWER: Subroutine COLSTEP assumes that the density of states is smooth in the upper portion of the double array. In general, you should plot the density of states vs. E to confirm that the energy dividing the two portions of the double array is high enough so that the density of states does not fluctuate more than, say, 5%. If the fluctuations are larger, they will cause problems in selecting the step sizes, resulting in anomalously large probabilities of large activating steps.

6. QUESTION: Why does my compiler have trouble reading the example data files that contain names (CHARACTER CONSTANTS)?

ANSWER: If your compiler does not accept the CHARACTER CONSTANTS in the example data files, enclose the CHARACTER CONSTANTS in apostrophes (') and try again. "Free format" (list directed input) is used by the MultiWell suite. Different compilers use different delimiters to separate the input fields in free format. Most compilers use commas and spaces, but some also will recognize tabs, returns, or linefeeds. Most compilers will recognize CHARACTER CONSTANTS (e.g. file names and chemical species names) when they are contained in apostrophes (e.g. 'xyz'). Other compilers will also accept quotes (e.g. "xyz"), or no special enclosing delimiters (e.g. xyz).

7. QUESTION: How can the 2-D hindered rotations in loose transition states be handled using DenSum?

ANSWER: There are at least three different ways 2D-hindered rotors for loose transition states can be handled by DenSum. The first is by using the restricted Gorin prescription of Smith and Golden[43] (see Benson's discussion[62] of the Hindered Gorin Model), which is based on a modified version of the Gorin Model.[80] The second is to replace the restricted rotor formulation with one based on particle-in-a-box. The third is to use two 1-D hindered rotors (implemented in DenSum) instead of one 2-D hindered rotor (2-D hindered rotors are not implemented in DenSum).

Smith and Golden[43] use a "hindrance parameter" to modify the moment of inertia of a 2-D free rotor: the Hindered Gorin Model. They find the value of the hindrance empirically by varying it until they achieve a fit with experimental rate constant data. In my opinion, a potential drawback of the Smith and Golden approach is that the rotor model has no zero point energy, and restricting the range of rotational motion results in a model that is more like a particle-in-a-box, which has a finite zero point energy. It is for this reason that I added the particle-in-a-box degree of freedom type to DenSum. The zero point energy is important because of the role it plays in isotopic reactions. To use particle-in-a-box energy levels (instead of free rotor levels), one can empirically vary the "frequency" parameter until agreement with experiment is achieved, just as done by Smith and Golden.

The third alternative is to use two 1-D hindered rotors for each 2-D internal rotation. Here, the moment of inertia can be used directly in DenSum and the hindrance potential is then varied until agreement with the rate constant data is achieved. I've done a few brief tests of this approach and it seems to work well. The use of an actual hindered rotor is attractive to me, because it seems more physically realistic (one of these days we'll investigate this using quantum chemical calculations). The drawback is that state degeneracies may not be calculated correctly.

I don't know if there is a significant difference among rate constants calculated using the methods described above. Each of the methods is probably quite adequate in fitting almost any experimental data (a detailed comparison of the methods would be quite interesting). However, making predictions for temperatures where no experimental data are available requires formulation of an ad hoc model for hindrance as a function of temperature. Jordan, Smith, and Gilbert[81] have formulated such a model (based on free rotors) and have tested it for a few cases; it is possible that their method could be used with Densum. The Marcus and Wardlaw approach[82,83] is more predictive, but only if an accurate multi-dimensional potential energy surface is available; their approach is difficult to implement (and cannot be done in the present version of DenSum). Miller and Klippenstein[84] have used VariFlex software,[85] which implements an extended version of the Marcus-Wardlaw approach.

Appendix. Theoretical Basis

Much of this Appendix is based on an expanded and corrected version of the paper originally published in the International Journal of Chemical Kinetics: J.R. Barker, Multiple-well, multiple-reaction-path unimolecular reaction systems. I. MultiWell computer program suite. *Int. J. Chem. Kinetics*, **33**, 232-45 (2001).

A.1. Introduction

Here, the theoretical basis for MultiWell is summarized. Inevitably, various approximations and assumptions must be adopted due to computational limitations and to the absence of physico-chemical knowledge. The numerical approximations are described so that program users can better assess MultiWell's limitations and strengths. The principal assumptions made in formulating the master equation are reviewed.

In the next section, the master equation is described formally. In subsequent sections, the stochastic methodology is described along with a brief discussion of some of the merits and limitations of the hybrid master equation approach relative to other methods. Methods for computing microcanonical unimolecular reaction rates and energy transfer step sizes are described, followed by a description of various initial conditions that can be selected as options. Finally, the calculation input and output are outlined.

A.2. The Active Energy Master Equation

The current version of MultiWell is based on the one-dimensional master equation, in which the active (randomizable) internal energy is modeled, but it is planned that future extensions will explicitly include angular momentum (the two-dimensional master equation[86-94]). The master equation provides the fundamental theoretical basis for modeling systems in which both energy transfer and chemical reaction can occur.[1,16,18,30,65] It is comprised of a set of coupled integro-differential equations that describe the rates of production and loss of chemical species at specified energies.

A.2.1 Internal Energy and Active Degrees of Freedom

Unless slow intramolecular vibrational energy redistribution (IVR) is specified, the internal energy *E* is always assumed to be fully randomized among the active degrees of freedom. The internal energy for a particular species (stable molecule or transition state) includes the energy (measured from the zero point energy; see Figure 0) that resides in the internal modes (vibrations, torsions, and internal rotations) and an active external rotation. Nonlinear polyatomic species have three external rotational degrees of freedom characterized by moments of inertia I_A, I_B, and I_C. The usual pragmatic approach[16,18] is to assume the molecule can be approximated as a symmetric top with two of the moments of inertia equal to one another ($I_B = I_C$), producing a degenerate two-dimensional external rotation. The third external rotor is associated with the symmetric top figure axis and is sometimes termed the K-rotor. The K-rotor is assumed to exchange energy freely with the other internal degrees of freedom, while the degenerate two-dimensional external rotation is assumed to be inactive.[16,18,30,65,95] (More sophisticated treatments of rotations can be utilized in the present version of MultiWell by calculating specific rate constants (*k(E)*) externally and then providing them in data files read by MultiWell.)

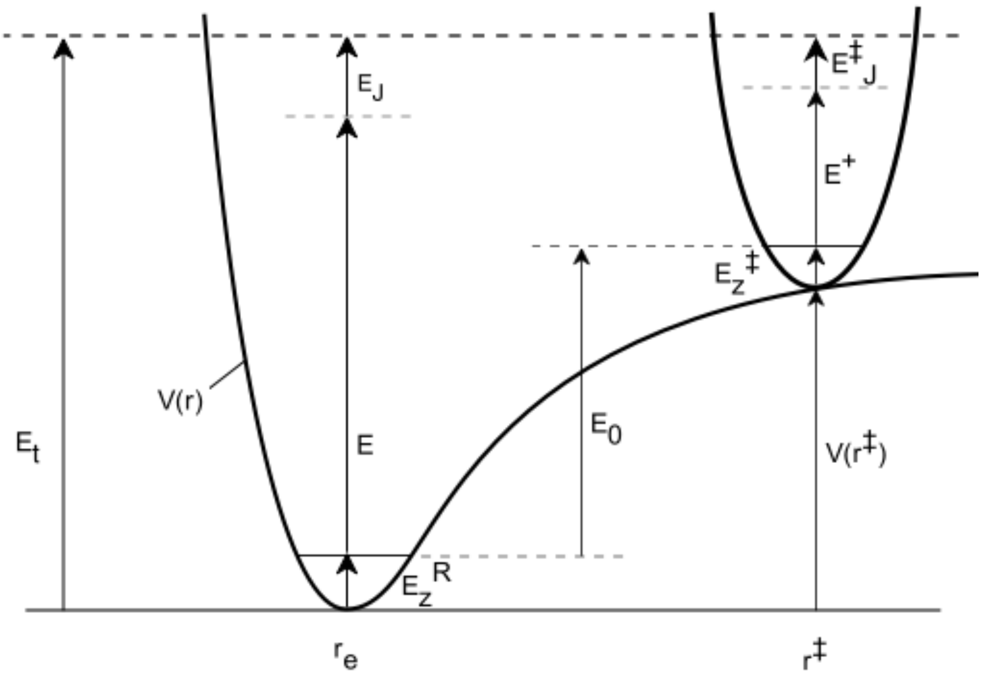


Figure 0. Energy schematic. The active energy *E* is measured from the zero point energy of the reactant well. The critical energy E_0 is the zero point energy difference between the transition state and the reactant well.

A.2.2 Sums and Densities of States

The MultiWell suite of computer codes includes `DenSum`, which utilizes the Stein-Rabinovitch[10] version of the Beyer-Swinehart algorithm[11] for exact counts of states for species comprised of separable degrees of freedom. The present version of `DenSum` can accommodate harmonic oscillators, Morse oscillators, and free rotors. The K-rotor is included with the internal degrees of freedom when calculating the sums and densities of states. There are two options for the treatment of rotations. The usual option is to use the convolution method developed by Astholz et al.,[29] which is computationally efficient and accurate for rotors with small rotational constants. The second method is to use exact counts of rotational states. The second method is preferred if the rotational constant is larger than ~1 $cm^{-1.}$ `DenSum` produces an output file that is subsequently used as an input file by `MultiWell`. The inactive two-dimensional external rotation is specified in the general MultiWell data file.

Hindered Internal Rotations

In the MultiWell Program Suite, torsional modes are assumed to be separable from other motions (i.e. translations, vibrations, external rotations and other internal rotations) and are treated as one-dimensional quantum hindered internal rotations. The Schrödinger equation for such 1-D torsional motions is solved as described in the Technical Notes section of this manual.

The 1-D torsion coordinate is defined as χ, the torsional dihedral angle ($0 \leq \chi < 2\pi$). Users must supply the potential energy V(χ) and the torsion rotational constant function $B_{hr}(\chi)$ (or the corresponding moment of inertia function I(χ)). Potential energies V(χ) and molecular geometries can be computed at discrete values of χ by using any of the many available quantum chemistry codes, such as Gaussian,[55] Molpro,[60] or CFour.[56] The results for V(χ) can be fitted to a suitable truncated Fourier series. **Program `lamm`**, a module in the MultiWell Program Suite, is intended to help users compute $B_{hr}(\chi)$, based on the output from quantum chemistry programs (see **Chapter 8** of this manual).

A.2.3 Master Equation for the Vibrational Quasi-Continuum

Because of the difficulties encountered in solving the 2-D master equation, it is usually reduced to 1-D by making simplifying assumptions, and then solving numerically by one of the standard methods.[1,22,96-98] However, Marcus showed that the microcanonical RRKM specific rate constant depends on the active energy E, i.e. the energy not constrained by conservation of angular momentum. Thus angular momentum has an effect that must be taken into account for accurate results. The first approach to this problem also originated with Marcus,[99] who invoked the concept of adiabatic rotation to make "centrifugal corrections". Marcus' approach has been reviewed by Waage and Rabinovitch[100] and is found in standard monographs.[18,31,65] In this approach, the RRKM specific rate constant $k(E_t,J)$, which is written as depending on total energy E_t and the angular momentum quantum number J is reduced to an approximate k(E) (at a given temperature), where E is the active energy of the reactant: all explicit reference to angular momentum quantum number J has been removed from the 2DME, which has thus been reduced to a 1DME. This approach is outlined in the following.

At high vibrational energies, a quasicontinuum of vibrational states exists and intramolecular vibrational redistribution (IVR) is rapid. Experiments show that IVR is slow at low energy, exhibits multiple time scales, and becomes rapid at energies where the vibrational state density is of the order of 10^2–10^3 states/cm^{-1}.[101] At these state densities, some vibrational states overlap significantly within their natural widths as governed by infrared spontaneous emission rates. At state densities greater than ~10^7 states/cm^{-1}, most states are overlapped within their natural widths. The onset of "rapid" IVR is a convenient marker for the onset of the vibrational quasicontinuum. However, this criterion leaves some uncertainty because IVR exhibits multiple time constants and thus some modes remain isolated even at higher vibrational state densities.[101]

In the vibrational quasicontinuum, individual quantum states cannot be resolved and the master equation can be written as a function of either of two equivalent sets of independent variables: E_t,*J* or *E*,*J*. In terms of active energy *E* and angular momentum quantum number *J* it takes the following form [18,65,96,98]:

$$\begin{aligned}\frac{dN(E',J',t)}{dt} &= F'\left(E',J',t\right)+\sum_J\int_0^\infty R\left(E',J';E,J\right)\,N\left(E,J;t\right)dE \\ &\quad -\sum_J\int_0^\infty R\left(E,J;E',J'\right)\,N\left(E',J',t\right)dE-\sum_{i=1}^{channels}k_i\left(E',J'\right)N\left(E',J',t\right)\end{aligned} \tag{2.3-1}$$

where the primed *E'* and *J'* refer to the current state and the unprimed quantities refer to a different state of the excited species, which is undergoing reaction. In Eq. 2.3-1, *N(E',J',t)dE'* is the concentration of a chemical species with active energy in the range *E'* to *E'+dE'*; *R(E,J;E',J')* is the (pseudo-first-order) rate coefficient for collisional energy transfer from initial energy *E'* to energy *E* and quantum number *J'* to *J*; *F'(E',J',t)dE'* is a source term (e.g. thermal, chemical, or photo activation, or isomerization); and $k_i(E',J')$ is the unimolecular reaction rate constant for molecules at energy E' and rotational quantum number J' reacting *via* the i^{th} reaction channel.

To reduce the 2DME to a 1DME, we first assume that the rotational population distribution describing the adiabatic rotations is always Boltzmann:

$$P_e\left(J\right) = Q_J^{-1} g_J \exp\left(-E_J / k_B T\right) \tag{2.3-2a}$$

where g_J = 2J+1 for a 2-D symmetric top (which is typically assumed) and other forms for 1-D and spherical tops; the rotational energy E_J and the rotational partition function Q_J for the adiabatic rotor(s) are given by Eq. 2.3-2b and 2.3-2c, respectively:

$$E_J = BJ\left(J+1\right) \tag{2.3-2b}$$

$$Q_J = \sum_J g_J \exp\left(-E_J / k_B T\right) \tag{2.3-2c}$$

The constant B is the rotational constant. By using these expressions, the concentrations can be written as

$$N\left(E,J,t\right) = N\left(E,t\right) P_e\left(J\right) \tag{2.3-3}$$

In the same spirit, we assume the source function produces the same rotational distribution:

$$F'\left(E',J',t\right) = F\left(E',t\right) P_e\left(J'\right) \tag{2.3-4}$$

Although recent trajectory calculations show that the assumption is probably not accurate,[26]) we follow Smith and Gilbert[88] and Miller et al.[97] in adopting the pragmatic assumption that the collision rates in Eq. 1 can be written in a separable form:

$$R\left(E,J;E',J'\right) = R\left(E,E'\right) P_e\left(J\right) \tag{2.3-5}$$

where R(E,E') has no dependence on angular momentum and can be expressed as the product of the total vibrationally inelastic collision frequency (ω) multiplied by the "collision step-size distribution", P(E,E'), which expresses the probability that a molecule initially in the energy range from E' to E'+dE' will undergo an inelastic transition to the energy range E to E+dE:

$$R(E,E')\,dE = \int_0^\infty R(E,E')\,dE \;\left\{ \frac{R(E,E')\,dE}{\int_0^\infty R(E,E')\,dE} \right\}, \tag{2.3-6a}$$

$$= \omega\, P(E,E')\, dE \tag{2.3-6b}$$

The first factor on the right hand side of Equation (2.3-6a), the integral over the rates of all inelastic transitions from initial energy E', is the frequency of inelastic collisions, ω, which may depend on the initial energy E'. Usually, the collision frequency is calculated from the expression $\omega = k_c[M]$, where k_c is the bimolecular rate constant for inelastic collisions and [M] is bath gas concentration. The second factor (in curly brackets) on the right hand side of Equation (2.3-6a) is $P(E,E')dE$. It is important to emphasize that the factorization of $R(E,E')$ in Equation (2.3-6) is merely for convenience and that k_c and $P(E,E')$ never occur independently of one another. Furthermore, *P(E,E')* is only a proper probability density function when ω is *exactly* equal to the inelastic collision rate constant. Under this assumption, *P(E,E')* is normalized:

$$\int_0^\infty P(E,E')\,dE = 1 \tag{2.3-7}$$

Note that collision step-size distributions for activating and deactivating collisions are connected *via* detailed balance:

$$\frac{P(E,E')}{P(E',E)} = \frac{\rho(E)}{\rho(E')}\exp\left\{-\frac{E-E'}{k_B T_{trans}}\right\}, \tag{2.3-8}$$

where $\rho(E)$ is the density of states at energy E, T_{trans} is the translational temperature, and k_B is the Boltzmann constant. The relationships among *P(E,E')*, k_c, and the normalization integral are further discussed below.

Using these definitions, we obtain

$$R\left(E,J;E',J'\right) = \omega P\left(E,E'\right)P_e\left(J\right) \tag{2.3-9}$$

Smith and Gilbert showed that the master equation can be solved approximately when using this separable form. In MultiWell, this separable form is adopted pragmatically in order to obtain a semi-analytical solution to the master equation. (Miller and coworkers have further generalized the function postulated by Smith and Gilbert,[88] and the details may be found in their papers.[97,98])

After combining Eq. 2.3-1-9, we obtain

$$\begin{aligned}\frac{dN(E',t)}{dt}P_e\left(J'\right) &= F\left(E',t\right)P_e\left(J'\right) + \sum_J P_e\left(J\right)\omega P_e\left(J'\right)\int_0^\infty P\left(E';E\right)N\left(E;t\right)dE \\ &- N\left(E',t\right)P_e\left(J'\right)\sum_J P_e\left(J\right)\omega\int_0^\infty P\left(E;E'\right)dE - N\left(E',t\right)P_e\left(J'\right)\sum_{i=1}^{channels} k_i\left(E',J'\right)\end{aligned} \tag{2.3-10}$$

In this expression, the sums over *J* are equal to unity because $P_e(J)$ is normalized. Every term contains the factor $P_e(J')$. By summing every term in Eq. 2.3-6 over *J'* and noting that the sum equals unity in each term except the last, we obtain

$$\begin{aligned}\frac{dN(E',t)}{dt} &= F\left(E',t\right) + \omega\int_0^\infty\left[P\left(E';E\right)N\left(E,t\right) - P\left(E;E'\right)N\left(E',t\right)\right]dE \\ &- N\left(E',t\right)\sum_{i=1}^{channels}\left\langle k_i\left(E'\right)\right\rangle_J\end{aligned} \tag{2.3-11}$$

where $<k_i(E')>_J$ is the RRKM specific rate constant, averaged over the rotational energy distribution. This is the general form of the 1-D master equation treated by MultiWell.

The averaging of $k_i(E',J')$ over the rotational energy distribution may be carried out as follows, using the variables defined in Fig. 2.3-1. The RRKM specific rate constant for a specific *J'* can be written[99]

$$k\left(E',J'\right)=\frac{L^{\ddagger}}{h}\frac{G^{\ddagger}\left(E^{+},J'\right)}{\rho\left(E',J'\right)} \tag{2.3-12}$$

where $L^{\ddagger}$ is the reaction path degeneracy, h is Planck's constant, $G^{\ddagger}(E^{+},J')$ is the sum of states of the transition state as a function of the active energy E^{+} in the transition state, and $\rho(E',J')$ is the density of "vibrational" states of the reactant when the rotational quantum number is J'. The various energies are related as follows (see Fig. 2.3-1):

$$E'+E_J=E^{+}+E_0+E_J^{+} \tag{2.3-13a}$$

or

$$E'=E^{+}+E_0+\Delta E_J \tag{2.3-13b}$$

where E_J and E_J^{+} are the energies of the adiabatic rotations in the reactant and in the transition state, respectively, and $\Delta E_J = E_J^{+}-E_J$. The rotational energies are given by Eq. 2.3-2b and by the analogous equation:

$$E_J^{+}=B^{+}J\left(J+1\right) \tag{2.3-14}$$

Using these relationships, $k_i(E',J')$ can be written

$$k\left(E',J'\right)=\frac{L^{\ddagger}}{h}\frac{G^{\ddagger}\left(E'-E_0-\Delta E_J\right)}{\rho\left(E',J'\right)} \tag{2.3-15}$$

From Eqs. 2.3-2b and 2.3-14 it is apparent that $E_J = (B/B^{+})E_J^{+}$, and ΔE_J can be written as a function of E_J^{+} alone: $\Delta E_J = (1-B/B^{+})E_J^{+}$. The rate constant must now be averaged over the rotational population distribution:

$$\left\langle k\left(E',J'\right)\right\rangle_J=Q_J^{-1}\sum_{J'}k\left(E',J'\right)g_J\exp\left(-E_J/k_BT\right) \tag{2.3-16a}$$

From the definition of ΔE_J, this expression can also be written

$$\left\langle k\left(E',J'\right)\right\rangle_J=Q_J^{-1}\sum_{J'}k\left(E',J'\right)g_J\exp\left[-\left(E_J^{+}-\Delta E_J\right)/k_BT\right] \tag{2.3-16b}$$

Equations 2.3-16a and 2.3-12b are suitable for directly averaging over the rotational quantum number. Care must be taken however, because the rate constant is only defined at energies $E' \geq (E_0+\Delta E_J)$. Because $\Delta E_J \leq 0$, the reaction threshold may be significantly lower in energy than E_0, the critical energy for $J = 0$.

The *ad hoc* approximate forms postulated by Smith and Gilbert[88] and by Miller and coworkers[97,98] are not accurate. thus we feel that elaborate treatments cannot be justified and have instead adopted the following pragmatic approach.

The active degrees of freedom may include one or more external rotations. Typically, for a prolate symmetric top, the K-rotor is active. For a rigid rotor, the K quantum number is a "good" quantum number and is restricted to the range from $-J$ to $+J$. However, a highly excited molecule is hardly rigid and K is not conserved, although it may exhibit a "propensity" to be conserved for some period of time. As a result, it is commonly assumed that the K-rotor of a non-rigid top can exchange energy freely with the internal degrees of freedom: i.e. the K-rotor is usually assumed to be an active degree of freedom.

It is common practice to allow K to vary freely, except for conservation of energy, and treat the K-rotor as just one more separable degree of freedom when computing densities of states.[16,18,30,31] In fact, treating the K-rotor in this way may be more accurate than restricting K to the range from $-J$ to $+J$.[102] However, there always exist $2J'+1$ states with the same value of J', even for a completely asymmetric top. Thus neglecting the restriction on K has the effect of increasing the density and sum of states slightly, but this error is insignificant except for molecules with large rotational constants at very low temperatures.

Here, it is pragmatically assumed that the K-rotor is active and the K quantum number has no restrictions other than conservation of energy. As a result, the approximate density of states of the reactant molecule has no explicit dependence on the J quantum number: $\rho(E',J')$ is replaced by $\rho(E')$ in Eq. 2.3-16 and the average specific rate constant can be written

$$\left\langle k\left(E',J'\right)\right\rangle_J = \frac{L^{\ddagger}\left\langle G^{\ddagger}\left(E'-E_0-\Delta E_J\right)\right\rangle_J}{hQ_J\rho\left(E'\right)} \tag{2.3-17a}$$

where the averaged sum of states is given by

$$\left\langle G^{\ddagger}\left(E'-E_0-\Delta E_J\right)\right\rangle_J = \sum_{J'} G^{\ddagger}\left(E'-E_0-\Delta E_J\right) g_J \exp\left(-E_J / k_B T\right) \tag{2.3-17b}$$

An additional pragmatic approximation is useful. Marcus showed that by substituting an average value $<\Delta E_J> = (1-B/B^+)<E_J^+>$ in place of ΔE_J in Eq. 2.3-13, a useful approximate expression is obtained,[99] which does not retain any explicit dependence on the rotational state J':

$$k_a\left(E'\right) = \frac{L^{\ddagger}}{h}\frac{G^{\ddagger}\left(E'-E_0-\left\langle\Delta E_J\right\rangle\right)}{\rho\left(E'\right)} \tag{2.3-18}$$

where the subscript "a" was used by Marcus to denote the approximate form. For present purposes, it is convenient to use the same approximation in the exponential factor in Eq. 2.3-16b, giving

$$\left\langle k\left(E'\right)\right\rangle_J = k_a\left(E'\right)\exp\left(\left\langle\Delta E_J\right\rangle\right)Q_J^{-1}\sum_{J'} g_J \exp\left[-E_J^+ / k_B T\right] \tag{2.3-19}$$

The remaining sum over J' in Eq. 2.3-19 can be recognized as $Q_J^{+,}$ the rotational partition function for the transition state. Marcus found $<E_J^+> = l\mathrm{k_B T}/2$ (where l is the number of adiabatic rotors), the average rotational energy in the transition state, by weighting the average according to the reactive flux[99,103] (also see the discussion in Robinson and Holbrook[30] and in Holbrook et al.[31]). Using this value of $<\Delta E_J> \approx (1-B/B^+)l\mathrm{k_B T}/2 = (1- I^+/I)\, l\mathrm{k_B T}/2$, where I and I^+ are the moments of inertia for the adiabatic rotor in the reactant and the transition state, respectively.[99] The final result is can be written as

$$\left\langle k\left(E'\right)\right\rangle_J = k_a\left(E'\right)\frac{Q_J^+}{Q_J}\exp\left(\left\langle\Delta E_J\right\rangle\right) \tag{2.3-20a}$$

or

$$\left\langle k\left(E'\right)\right\rangle_J = k_a\left(E'\right)\left(\frac{I^+}{I}\right)^{l/2}\exp\left(\frac{l}{2}\left[1-\frac{I^+}{I}\right]\right) \tag{2.3-20b}$$

where the ratio of the (classical) rotational partition functions has been replaced by the ratio of the moments of inertia, and l is the number of adiabatic rotors. Usually it is assumed that $l = 2$.

These simplified forms perform well in master equation simulations, although although the energy transfer parameters may differ to some extent from other treatments. These possible differences are generally not considered further, because no current master equation treatment of energy transfer is known to be correct and because energy transfer parameters and collision step-size distributions generally are not known from other sources, but are simply used as empirical fitting parameters.

High Pressure and Low Pressure Limits: k_∞ and k_0

Prior to the release of Multiwell-2011.1, we became aware of a problem with the implementation of the centrifugal factors in MultiWell. David Golden pointed out that at very high temperatures, where it is possible to simulate both a recombination reaction and its reverse dissociation, the two rate constants were inconsistent with each other. This discrepancy was traced to the centrifugal corrections. This problem has been corrected, and a discussion of the current and prior versions of k_0 and k_∞ can be found in Section 9.10 of this User Manual.

The high pressure rate constant is the average rate constant at pressures high enough so that differences of the actual population distribution from the thermal Boltzmann distribution are negligible. The thermal Boltzmann energy distribution for reactant A is

$$B_A(E',T) = Q_A^{-1}\rho_A\left(E'\right)\exp\left(-E'/k_BT\right) \tag{2.3-21a}$$

where the partition function Q_A is

$$Q_A = \int_0^\infty \rho_A\left(E'\right)\exp\left(-E'/k_BT\right)dE' \tag{2.3-21b}$$

The average rate constant is therefore

$$k_\infty(T) = \frac{1}{Q_A}\int_{\varepsilon_0'}^{\infty}\left\langle k_a\left(E'\right)\right\rangle_J \rho_A\left(E'\right)\exp\left(-E'/k_BT\right)dE' \tag{2.3-22a}$$

or

$$k_\infty(T) = \frac{1}{Q_A}\frac{Q_J^+}{Q_J}\int_{\varepsilon_0'}^{\infty} k_a\left(E'\right)\rho_A\left(E'\right)\exp\left[-\left(E'-\left\langle \Delta E_J\right\rangle\right)/k_BT\right]dE' \tag{2.3-22b}$$

By transforming the variable of integration from E' to E^+ (i.e. by replacing ΔE_J in Eq. 2.3-7b with the average value, $<\Delta E_J>$ and obtaining $E^+ = E'-E_0-<\Delta E_J>$) and changing the limits of integration, the resulting expression for k_∞ is the same as canonical TST (Eq. 4.30 of Robinson and Holbrook,[30] and Eq. 3.30 in Holbrook et al.[31] at the high pressure limit).

Comparison with Other Models

Equation 3.30 in Holbrook et al.[31] and Eq. 4.30 in Robinson and Holbrook[30] are based on the strong collision assumption, in which every collision results in producing a canonical active energy distribution. In this section, the strong-collision rate constant is derived using the rotationally averaged <*k*(*E'*)>J. For convenience, concentration is written as the product of a normalized energy distribution, *y*(*E',t*), and a time-dependent total concentration, *N*(*t*):

$$N(E',t) = N(t)\,y(E',t) \tag{2.3-23}$$

where *N(t)* decays with time and the energy distribution *y*(*E',t*) is normalized at all times. With this substitution, the LHS of Eq. 2.3-5 becomes

$$\frac{dN(E',t)}{dt} = y(E',t)\frac{dN(t)}{dt} + N(t)\frac{dy}{dt} \tag{2.3-18}$$

In the absence of chemical reaction and input flux, and given enough time, the population distribution relaxes to steady state and *dy/dt* = 0. Furthermore, the unimolecular rate constant (for channel *i*) is defined at steady state: $k_{uni,i} = -N^{-1}\mathrm{d}N/\mathrm{d}t$. Thus, after simplification the steady-state master equation can be written

$$-y(E')k_{uni,i} = \frac{F(E',t)}{N(t)} + \int_0^\infty R(E';E)\,y(E)dE - y(E')dE\left[\int_0^\infty R(E;E')dE + \sum_{i=1}^{channels}\left\langle k_i(E')\right\rangle_J\right] \tag{2.3-21}$$

When *F(E',t)* = 0, Eq. 2.3-21 can be solved for the steady-state population distribution *y(E')*:

$$y(E') = \frac{\int_0^\infty R(E';E)\,y(E)dE}{\left[\int_0^\infty R(E;E')dE + \sum_{i=1}^{channels}\left\langle k_i(E')\right\rangle_J\right] - k_{uni,i}} \tag{2.3-22}$$

To obtain $k_{uni,i}$ with the strong-collider approximation, we use the following identity:

$$R(E;E') = k_c[M]\frac{\rho(E)\exp(-E/k_BT)}{Q_2(T)} \tag{2.3-23}$$

where the rate of energy transfer is expressed as the product of a collision rate constant k_c and collider bath concentration [M], multiplying a thermal canonical energy distribution, which is normalized by the partition function. Note that according to the strong-collider assumption, the energy transfer rate has no dependence on the energy prior to collision. With this substitution, the integrals in eq. 2.3-22 can be evaluated readily:

$$\int_0^\infty R(E';E)\,y(E)dE = k_c[M]\frac{\rho(E')\exp(-E'/k_BT)}{Q_2(T)}\int_0^\infty y(E)dE \tag{2.3-24a}$$

$$= k_c[M]\frac{\rho(E')\exp(-E'/k_BT)}{Q_2(T)} \tag{2.3-24b}$$

$$\int_0^\infty R(E;E')dE \; = \frac{k_c[M]}{Q_2(T)}\int_0^\infty \rho(E)\exp(-E/k_BT)dE = k_c[M] \tag{2.3-25}$$

Thus eq. 2.3-22 becomes

$$y(E') = \frac{k_c[M]}{\left[k_c[M] + \sum_{i=1}^{channels}\langle k_i(E')\rangle_J\right] - k_{uni,i}}\frac{\rho(E')\exp(-E'/k_BT)}{Q_2(T)} \tag{2.3-26}$$

Except at extremely low internal energies, the magnitude of k_{uni} is much smaller than the rotationally averaged $<k(E')>_J$ and hence can be neglected in the denominator. The reaction rate at active energy E' is given by the product of $y(E')$ and $<k(E')>_J$. After integrating over all internal energies greater than $E_0' = E_0 + <\Delta E_J>$, the reaction critical energy, we obtain the strong-collider unimolecular rate constant for channel i:

$$k_{uni,i} = \int_{E_0+\langle\Delta E_J\rangle}^{\infty} \frac{k_c[M]\langle k_i(E')\rangle_J}{\left[k_c[M] + \sum_{i=1}^{channels}\langle k_i(E')\rangle_J\right]}\frac{\rho(E')\exp(-E'/k_BT)}{Q_2(T)}dE' \tag{2.3-27}$$

Low pressure limit:

$$k_{low,i} = \frac{k_c[M]}{Q_2(T)}\int_{E_0+\langle\Delta E_J\rangle}^{\infty} \rho(E')\exp(-E'/k_BT)dE' \tag{2.3-28}$$

High pressure limit:

$$k_{\infty,i} = \frac{1}{Q_2(T)}\int_{E_0+\langle\Delta E_J\rangle}^{\infty} \langle k_i(E')\rangle_J\,\rho(E')\exp(-E'/k_BT)dE' \tag{2.3-29}$$

A.2.4 Multiple Species (Wells) and Multiple Reaction Channels

Here we consider chemical species that can be identified with local minima (wells) on the potential energy hypersurface. These species are distinct from transition states, which are located at saddle points. In MultiWell, each well is assigned an arbitrary index for identification and reactions are conveniently labeled with two indices: one to designate the reactant and the other to designate the product. For simplicity in notation, one or more of these indices are omitted in some of the following discussion.

A master equation such as Equation (2.3-1) can be written for each well and the equations are coupled via the chemical reaction terms. Each reaction channel is associated either with another well, or with fragmentation products. Each isomerization is reversible and the transition state is the same for the corresponding forward and reverse reactions. In principle, the existence of isomers leads to splitting of vibrational levels, as in the inversion doubling of ammonia, but if tunneling is negligible, the wells can be considered independently.[104] Thus each well has its own vibrational assignment, molecular structure, and corresponding density of states.

Two technical problems arise when using an energy grained master equation[16,18,30,31,65] to simulate multiple-well systems. First, the number of coupled differential equations can grow prohibitively as the energy grain size (ΔE_{grain}) is reduced, making the numerical solution very difficult or impossible. Second, because each well has its own zero of energy and reaction threshold (critical) energies, it is difficult to match the energy grain boundaries. The reaction threshold energies for forward and reverse reactions are tied to one another. For accurate numerical results it is necessary to match the energy grains of the coupled wells. The matching of energy grains at one reaction threshold may lead to mis-matches at other thresholds and to artificially shifted energies of the wells, relative to one another. These energy shifts produce anomalous results for large grain sizes. This problem can be neglected if the energy grains are very small, but small energy grains lead to very large sets of coupled equations. In all cases, the calculations should be repeated with successively smaller energy grains until the results are independent of ΔE_{grain}: convergence must be achieved.

When a continuum master equation is used, energy mis-matching and anomalous shifts never create problems. However, the sparse density of states regime at low energies within wells and for transition states near reaction thresholds is not well represented by a continuum model. This difficulty is minimized in MultiWell by using a hybrid master equation approach.

A.2.5 Hybrid Master Equation Formulation

Effectively, the hybrid master equation formulation uses a continuum master equation in the quasicontinuum at high vibrational energies, and an energy-grained master equation at low energies, where the state density is distinctly discontinuous. This is accomplished by using Equation (2.3-1) for the continuum master equation throughout the entire energy range but discretizing the state density, population, and transition rates at low energy. At high energy, Multiwell employs interpolation to determine the density of states and specific rate constants (k(E)). Values of ρ(E) and k(E) are stored in ordered arrays at specific values of E and intermediate values are determined by interpolation. At low energies, ordered arrays of ρ(E) and k(E) are stored at smaller energy spacing (ΔE_{grain}) and interpolation is not used: the array entries nearest in energy are utilized directly. The two ordered arrays used for each energy-dependent quantity (ρ(E), k(E), etc.) are combined in "double arrays" which are discussed in the next section. At all energies, numerical integration is carried out with the trapezoidal rule, which introduces an energy grain in the low energy regime (where state densities are sparse), but gives good continuum results at high energy (where the state densities are smooth).

If a stochastic trial (see below) calls for a transition from the continuum space to an energy in the discrete space, the energy is aligned with the discrete energy grain. At low energy, many energy grains do not contain states (ρ(E) = 0) and transitions are not allowed to those states in MultiWell. As a result, population only resides in energy grains that contain states and collisional transitions low on the energy ladder can only take place with relatively large energy changes, due to the sparse density of states.

A.2.6 Energy Grain in the Hybrid Master Equation

Through the use of double arrays, high energy resolution is achieved in densities and sums of states at low energy and near reaction thresholds. By default, the double arrays have 500 elements (the dimensions can be changed, if desired). The low energy portion of the array is specified according to ΔE_{grain} and the number of array elements assigned to the low energy portion of the double array. The high energy portion is specified only according to the maximum energy. Thus the number of array elements used in the high energy portion and the energy grain in the high energy portion both depend on how many array elements remain after assigning the low energy portion. The same specifications are used for all double arrays, including arrays for densities of states ($\rho(E)$), sums of states ($G^{\ddagger}(E-E_0)$), specific rate constants ($k(E-E_0)$), etc. The discretization of these quantities is the natural result of exact count algorithms.

An example of a double array for the density of states $\rho(E)$ is shown in Figure 1 for benzene (vibrations[105] + K-rotor). In this example, the density of states was calculated using an energy grain of $\Delta E_{grain} = 10$ cm^{-1} and exact counts up to an energy of 85000 cm^{-1}, although only energies up to 10000 cm^{-1} are shown in the figure. In this example, the low energy regime was defined as the first 250 elements of a double array and thus covered the range from 0 to 2490 cm^{-1}. The remaining 250 elements of the double array overlap the low energy portion and cover the range all the way from 0 to 85000 cm^{-1} (the high energy regime) with an energy grain of 341.4 cm^{-1}. In Figure 1, $\rho(E)$ calculated with $\Delta E_{grain} = 10$ cm^{-1} is shown as the thin solid line and the double array elements are shown as the solid dots. The upper energy boundary for low energy range was chosen to fall within the vibrational quasicontinuum, as evidenced by $\rho(E) \geq$ 100 states/cm^{-1} and by the relative smoothness of the plot of $\rho(E)$. When $\rho(E)$ is sufficiently smooth, relatively little error is introduced by interpolating between the double array points.

In principle, convergence tests should be carried out for each simulation. Tests for convergence as ΔE_{grain} is reduced reflect several simultaneous effects: the $\rho(E)$ grain size is varied for every well, the $G^{\ddagger}(E-E_0)$ grain size is varied for every reaction, and the energy range covered by the low energy portion of every double array is varied. Because several attributes are affected by ΔE_{grain}, the variation of results with grain size cannot be interpreted precisely without extensive tests. However, as long as the energy range covered by the low energy portion of the double arrays is sufficient, smaller grain sizes will produce more accurate results and the results are seen to converge at small ΔE_{grain}, as illustrated in Figure 4 of Reference [23].

Typical convergence tests[23] show that 250 bins with $\Delta E_{grain} = 10$ cm^{-1} are usually suitable for the low energy portion of the double arrays. The upper energy bound for the high energy portion is typically 85000 cm^{-1} to 100000 cm^{-1}, depending on the temperature range and activation method being simulated. The small grain at every reaction threshold gives accurate results for the unimolecular reaction rates. The small grain at low energy within each well gives a good representation of the sparse density of states regime in every well. To achieve comparable numerical results by the matrix solution[65] of an energy-grained master equation for just a single well would require finding the eigenvalues of a matrix with 8500×8500 elements...a difficult task. The hybrid master equation approach has a distinct advantage in this regard.

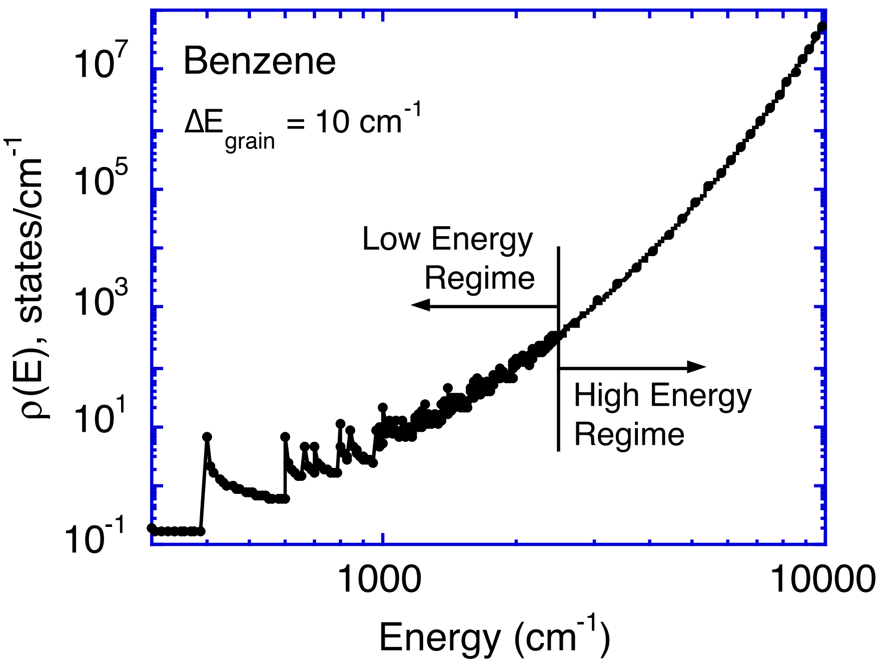


Figure 1. Density of states for benzene (including vibrations and one active external rotation). Solid line: density of states from exact count (ΔE_{grain} = 10 cm^{-1}); solid dots: elements of double array (see text for details).

A.3. Stochastic Method

A.3.1 Gillespie's (Exact) Stochastic Simulation Algorithm

Gillespie showed that a stochastic method gives the exact solution to a set of ordinary differential equations in the limit of an infinite number of stochastic trials.[20,21] The algorithm has been described in the context of chemical kinetics.[1,19,106] If a Markovian system is in a given state and can make transitions to other states via a set of transition rate coefficients, then for a given step in a stochastic simulation, Gillespie's algorithm gives a prescription for a) finding the duration of the time step and b) selecting the transition from among the choices. This algorithm is repeated step-by-step as long as desired and as long as transitions are possible.

Gillespie's method can be applied to both linear and non-linear systems.[19] Equation (2.3-1) is linear in y(E,t), which leads to a particularly convenient result which is described below. If Equation (1) contained non-linear terms to describe energy pooling, for example, the terms would contain factors such as the product y(E,t)×y(E',t). To solve this system numerically requires using an energy-grained master equation with a swarm of stochastic trials and storing an evolving vector of populations as a function of energy. Here, the number of stochastic trials can be identified with a number of pseudo molecules that initially are placed in a set of energy grains. At each time step, a pseudo molecule is moved from one energy grain to another as described by Gillespie and the swarm of pseudo molecules maps out the evolving energy distribution. This approach has been used by Veerecken et al.[94,107] to simulate unimolecular and recombination reactions and it can in principle be extended to non-linear systems. The difficulties in this approach are associated with the energy-grained master equation (see above) and with the requirement for storage of the entire vector of y(E,t) at every time step. Given the current availability of inexpensive computer memory, the latter is not a serious limitation for single-well reaction systems. When several wells are involved, the bookkeeping is cumbersome. Moreover, the memory requirements of this technique can become prohibitive in the future if the one-dimensional master equation is to be extended to two dimensions by explicitly including angular momentum. MultiWell is designed so that the future extension to two-dimensions will be feasible.

For linear master equations, a different strategy[1] is possible using Gillespie's algorithm. Instead of using a swarm of stochastic molecules and storage of y(E,t) at every step, stochastic trials are run one at a time and snapshots of E and other variables are stored at convenient time intervals. The vector y(E,t) does not need to be stored. A "snapshot" simply records the energy and other properties of a single stochastic molecule as it progresses through a stochastic trial. The snapshot has no effect on the physics of the trial. Since the system is linear, the averaged result of an ensemble of stochastic trials gives the same result as a swarm of stochastic molecules. By retaining only the averaged results of the snapshots, the memory storage requirements are greatly reduced.

For a linear master equation, the loss terms can be expressed as first order in y(E,t) with first order rate coefficients A_j for k paths. These rate coefficients can be identified with the unimolecular rate constants and the collision frequency in Equation (2.3-1). According to Gillespie's algorithm, the duration of the next time step is chosen by using the uniform random deviate (i.e. random number) r_1 :

$$\tau = \frac{-\ln(r_1)}{A_T} \tag{3.1-1a}$$

where

$$A_T = \sum_{j=1}^{k} A_j \tag{3.1-1b}$$

The transition is selected from among the k paths by using a second random number r_2:

$$\sum_{j=1}^{n-1} A_j < r_2 A_T \leq \sum_{j=n}^{k} A_j \tag{3.1-2}$$

Here, the transition takes place via path n at time t+τ.

According to Gillespie's algorithm, the time intervals between stochastic steps are chosen randomly by Equation (3.1-1a). Thus the progress of the stochastic simulation is monitored via snapshots, as mentioned above. If collisional activation or deactivation is the result of a transition, then the next stochastic step is calculated using rate coefficients appropriate to the new energy. If isomerization to another well is the result of a transition, then the next stochastic step is calculated using the first order rate coefficients appropriate to the new well, based on E measured from the zero point energy of the new well. The snapshots from many stochastic trials are averaged. The results include the time-dependent average fractional populations of the isomers, the average internal energy of each isomer, and fractional yields of the fragmentation products, etc.

The computer time required for any given stochastic simulation depends on N_{trials}, the simulated time duration, and on the properties of the system that affect A_T in Equation (3.1-1b). For example, one of the A_j terms is the collision frequency, which is proportional to pressure. If the collision frequency is the dominant term in Equation (3.1-1b), then the average stochastic time-step is inversely proportional to pressure and the number of time steps (and the corresponding computer execution time) for the given simulated time duration is proportional to pressure. Of course, collision frequency is not always the dominant term in Equation (3.1-1b), but the same qualitative considerations can help in estimating required computer time.

Note that the effectiveness of Equation (3.1-2) is limited by the properties of the random number generator. The characteristics of various random number generators are discussed elsewhere,[90,108,109] where many potential pitfalls are described. It is important to use random number generators that have been thoroughly tested. Even assuming the random number generator produces a sequence that has no serial correlations, the number of random numbers in a sequence is limited and this imposes a limitation on the relative magnitudes of the A_m terms that can be selected according to Equation (3.1-2). For a 32-bit computer, a typical random number sequence contains $2^{31}-1 \approx 2.1\times10^{9}$ equally-spaced numbers. Thus if the ratio of minimum to maximum values of the rate constants is less than $\sim0.5\times10^{-9}$, then the path with the smaller rate can *never* be selected. Thus the random number generator places a rigorous upper bound on the dynamic range of rates that can be selected. A more serious limitation, however, is that an extraordinarily large number of stochastic trials is required in order to sample rare events with useful precision, as discussed in the next section.

A.3.2 Stochastic Uncertainties

The precision of the results obtained using stochastic methods depends on the number of stochastic trials. In the systems simulated by MultiWell, several species coexist and their relative populations sum to unity:

$$1 = f_1 + f_2 + \ldots = \sum_{i=1}^{species} f_i \tag{3.2-1}$$

The standard deviation in the instantaneous relative population of the i^th species is the square root of the variance calculated according to the multinomial distribution[110]:

$$\sigma_i = \sqrt{\frac{1}{N_{trials}} f_i\left(1-f_i\right)} \tag{3.2-2}$$

where f_i is the fractional population of the ith species and N_{trials} is the number of stochastic trials. Note that the standard deviation is reduced as the number of trials increases. Also note that the product $f_i(1-f_i)$ appears in Equation (3.2-2). Thus, the standard deviation is the same when, for example, $f_i=0.01$ and when $f_i=0.99$. These standard deviations are calculated and reported by MultiWell in its general output.

A large number of stochastic trials is needed when rare events must be simulated with high precision. Suppose that $f_i = 0.01$ and the desired precision corresponds to a relative statistical error of 1% (i.e. $\sigma_i/f_i = 0.01$). From Equation (8), one finds the required number of stochastic trials: $N_{trials} \approx 10^6$. For a relative error of 10%, only about 10^4 trials are needed. Thus, the required number of stochastic trials places a practical limit on the precision attainable for minor pathways.

A.4. Processes

A.4.1 Unimolecular Reactions

The energy-dependent specific unimolecular rate constant k(E) is given by the RRKM statistical theory[17,18,30,31,65]:

$$k(E)=\left[\frac{m^{\ddagger}}{m}\frac{\sigma_{ext}}{\sigma_{ext}^{\ddagger}}\right]\frac{g_e^{\ddagger}}{g_e}\frac{1}{h}\frac{G^{\ddagger}(E-E_0)}{\rho(E)} \qquad (4.1\text{-}1)$$

where m‡ and m are the number of optical isomers,[65] $\sigma_{ext}{}^{\ddagger}$ and σ_{ext} are the external rotation symmetry numbers, and $g_e{}^{\ddagger}$ and g_e are the electronic state degeneracies of the transition state and reactant, respectively; h is Planck's constant, $G^{\ddagger}(E-E_0)$ is the sum of states of the transition state, E_0 is the reaction threshold energy, and ρ(E) is the density of states of the reactant molecule. The internal energy E is measured relative to the zero point energy of the reactant molecule and the reaction threshold energy (critical energy) is the difference between the zero point energies of reactant and transition state. Equation (4.1-1) was written by assuming that the rotational external symmetry numbers were not used in calculating the sums and densities of states.[65] It is, however, assumed that internal rotor symmetry numbers are used explicitly in the sum and density calculations and hence do not appear in Equation (4.1-1). Note that the quantity set off in square brackets is the *reaction path degeneracy*.[65]

At low energies, where densities of states may be very sparse, ρ(E) may be very small or zero in a given energy grain (lower energy portion of the double array). In MultiWell, this is treated as if there are no reactant states in the energy grain: k(E) is set equal to zero in that grain. During evolution of the population distribution, population never resides in energy grains that do not contain reactant states.

Tight Transition States

For a tight transition state, $G^{\ddagger}(E-E_0)$ can be calculated from a vibrational-rotational assignment and the reaction threshold energy can be corrected approximately for angular momentum effects by using a pseudo-diatomic model.[16,18,65] All polyatomics are treated by DenSum and MultiWell as SYMMETRIC TOPS with moments of inertia I_A, I_B = I_C and corresponding rotational constants A = $\hbar^2/2I_A$ and B ==$\hbar^2/2I_B$. The rotational energy of a symmetric top is given by

$$E_r\left(J,K\right)=J\left(J+1\right)B+(A-B)K^2$$

where quantum numbers J and K refer to the **two-dimensional 2-D adiabatic rotor** (i.e. the one that conserves angular momentum ***J***) and to **one-dimensional rotation** about the top axis (projection of ***J*** on the top axis), respectively. For a given value of J, quantum number K can take values from -J to +J, inclusive; all such states are doubly degenerate except for K=0, which is singly degenerate. *In the present version of DenSum, the K-rotor is normally designated as a simple 1-D rotation (either quantized or classical)*. The moment of inertia for the K-rotor I_K is given by

$$I_K=\left[I_A^{-1}-I_B^{-1}\right]^{-1}$$

In many applications, $I_K \approx I_A$.

Because the K-rotor is normally designated as a simple rotation in MultiWell simulations, the range of K is normally **not** restricted to ±J. When $I_A < I_B$ (prolate top), then $I_K > 0$. When $I_A > I_B$ (oblate top), then $I_K < 0$. If the *K* quantum number is not constrained to $K \leq |J|$, then $E_r(J,K)$ can be <0. The unrestricted range of *K* is not serious a approximation for prolate tops, since the rotational energy is >0 for all values of *J* and *K*. For oblate tops, however, $E_r(J,K)$ can be <0 when $K >> J$. Thus the approximate treatment of the K-rotor may fail seriously for oblate tops. The ***kro*** degree of freedom type (see DenSum) employs the correct treatment for a user-specified value of *J*.

Thus the three external rotations of a non-linear molecule consist of a 1-dimensional rotor (the 1-D "**K-rotor**") and a 2-dimensional top (the **2-D adiabatic rotor**). The **K-ROTOR** properties are listed in **densum.dat** and included in density and sums of states calculations. Pragmatically, I_B is usually not equal to I_C, since most chemical species are not true symmetric tops. For *almost* symmetric tops (where $I_B \approx I_C$), one can use either of two reasonable approximations for I_{2D}:

$$I_{2D} \approx \left[I_B I_C\right]^{1/2} \text{ [Ref. }^{16,18}\text{] or } I_{2D} \approx \left[I_B^{-1} + I_C^{-1}\right]^{-1} \text{ [Ref. }^{64}\text{]}$$

The **2-D ADIABATIC ROTOR** moment of inertia is listed in **multiwell.dat** on Line 8 (for wells) or Line 14 (for transition states). The moment of inertia I_K of the K-rotor is given by

$$I_K = \left[I_A^{-1} - I_{2D}^{-1}\right]^{-1}$$

This is sometimes approximated as $I_K \approx I_A$. The K-rotor is normally listed as a 1-D rotation in the DenSum data file.

When *k(E)* is calculated according to RRKM Theory, **centrifugal corrections** (recommended Keyword `'CENT2'`) are applied by averaging *k(E,J)* over a thermal distribution of *J*. Here, *k(E,J)* is given by Eq. 4.1-1, but with the density of states in the denominator written as in Section 4.10.1 in Robinson and Holbrook[30] (or Section 3.10 of Holbrook et al.[31]). Essentially, *k(E,J)* can be written

$$k(E,J) = \left[\frac{m^{\ddagger}}{m}\frac{\sigma_{ext}}{\sigma_{ext}^{\ddagger}}\right]\frac{g_e^{\ddagger}}{g_e}\frac{1}{h}\frac{G^{\ddagger}(E^{+})}{\rho(E^{+} + E_0 + \Delta E_r(J))} \tag{4.1-2a}$$

where E^+ is the active energy that is assumed to randomize rapidly and $\Delta E_{rot}(J)$ is the difference in the adiabatic rotational energy between transition state and reactant molecule:

$$\Delta E_{rot}(J) = \left\{B_{\ddagger} - B_e\right\}J(J+1) \tag{4.1-2b}$$

Here, $B_{\ddagger}$ and B_e are the rotational constants (for the adiabatic 2-D rotations) of the transition state and the equilibrium reactant molecule, respectively. Since B_e is usually significantly larger than $B_{\ddagger}$, $\Delta E_{rot}(J)$ is usually negative. The rotational averaging is carried out as usual:

$$k(E) = \frac{1}{Q_{2D}}\sum_{J=0}^{\infty} k(E,J)\cdot(2J+1)\cdot\exp\left[\frac{-B_e \cdot J \cdot (J+1)}{k_B T}\right] \tag{4.1-2c}$$

This model amounts to assigning a fixed transition state, since $B_{\ddagger}$ does not change with rotational state (or inter-fragment distance), and thus is a relatively poor approximation for loose transition states. The limitations of this approach are overcome to a large extent if a semi-empirical approach like the Hindered Gorin Model (see below) is applied, where $B_{\ddagger}$ is assumed to vary with temperature.

Loose Transition States

For loose transition states, more elaborate techniques are needed for calculating *k(E)* accurately. Such techniques include Variational Transition State Theory,[65,95] Adiabatic Channel Model,[111] and Flexible Transition State Theory.[83] Computer codes have been published for some of these theories.[85,112] These methods can be used in the current version of MultiWell by calculating k(E) externally and supplying the k(E) values an appropriately formatted external data file. MultiWell will accept double arrays (from external files) that specify $G_{TS}(E-E_0)$ (which can be calculated conveniently using codes like `DenSum`, part of the MultiWell suite), or that specify $k(E-E_0)$. Since a double array is used, the effective energy grain can be very small near the reaction threshold, where high energy resolution is important.

The choice of approach depends on several factors, including personal preferences, computational costs, required accuracy, and whether experimental data are available. MultiWell can in principle be used for any method that one wishes to use, but the most sophisticated methods require considerable work outside the MultiWell package; the simpler methods can be implemented with built-in options. Even when computer resources are not a limitation, the limitations of current electronic structure theories are significant, possibly leading one to choose one of the simpler methods.

In the following, we list several methods, starting with the simplest, and describe strengths and weaknesses. These methods require either some experimental data, or an empirical estimation method (e.g. thermochemical kinetics[62]), or perhaps both. They are based on fitting canonical (thermal) rate constants and then using the appropriate parameters in microcanonical master equation simulations.

Inverse Laplace Transform Method

With this method, experimental Arrhenius parameters must be known or estimated for the high-pressure limit of the thermal decomposition reaction. If the Arrhenius parameters are known for the recombination reaction, the equilibrium constant K_{eq} can be used to determine the corresponding parameters for the unimolecular decomposition reaction. Since rate constants for recombination reactions are usually only weakly dependent on temperature, the activation energy for the recombination may be assumed to equal zero. In most cases, it is a good approximation to assume that E_∞, the Arrhenius activation energy at the high pressure limit, is equal to the critical energy for reaction (E_0). MultiWell input for `ILT` calls for only two parameters: E_0 and the corresponding Arrhenius A-factor.

A selectable option in MultiWell is to calculate k(E) using the Inverse Laplace Transform method described by Forst:[16,18,113]

$$k(E)=\left[\frac{m^{\ddagger}}{m}\frac{\sigma_{ext}}{\sigma_{ext}^{\ddagger}}\right]\frac{g_e^{\ddagger}}{g_e}A_\infty\frac{\rho(E-E_\infty)}{\rho(E)} \tag{4.1-3}$$

where A_∞ and E_∞ are the Arrhenius parameters for the corresponding high pressure limiting thermal rate constant. Note that the reaction path degeneracy (the quantity in square brackets) can be absorbed into A_∞ if desired. For added accuracy near the reaction threshold, E_∞ may be replaced in Equation (4.1-3) by E_0, the threshold energy. This substitution may improve the threshold behavior, but it introduces a small error in the calculated high pressure limit activation energy.

Required info:

- Experimental (or estimated) Arrhenius parameters
- Reaction enthalpy and K_{eq} are needed to obtain the rate constant for the reverse reaction.

Strengths:

- Quickest and easiest method (a built-in option in MultiWell)
- Does not require TST parameters for the transition state.

Drawbacks:

- Over wide temperature ranges, the thermal rate constants may not obey the strict Arrhenius form. Errors from this source, however, are usually relatively small.
- This method for computing k(E) is not as accurate as transition state theory (i.e. RRKM theory).
- The ILT does not allow the use of centrifugal correction factors, which are needed to account for conservation of angular momentum. Neglect of angular momentum can affect the magnitude and temperature-dependence of the energy transfer parameters needed to describe pressure fall-off.

Hindered Gorin Model

When the rate constant is known from experiments, it is often convenient to use a "restricted" or "hindered" Gorin Model with a "hindrance parameter" selected to reproduce a known rate constant for a recombination reaction.[43,62,65] The same transition state is appropriate for the corresponding unimolecular decomposition reaction. The basic idea is that the TST internal degrees of freedom consist of vibrations (the unperturbed vibrational frequencies of the reactants) plus internal rotations (the relative rotations of the two reactants with rotational constants modified by an empirical multiplicative constant) plus external rotations (two adiabatic rotations and the K-rotor). The empirical factor is adjusted in order to fit the experimental canonical recombination rate constant.

According to the Gorin model,[80] the two molecular fragments rotate independently of one another while separated at the distance corresponding to the centrifugal maximum (r_{max}) of the effective potential of the bond being broken. In the present work, the rotations of both fragments and the over-all transition state are treated approximately as symmetric tops. The over-all transition state has a 2-D external adiabatic rotation with moment of inertia given by $I_{2D}^{\ddagger} = \mu r^2_{max}$, where μ is the reduced mass of the two fragments, and a 1-D external rotation (the "K-rotor") with moment of inertia I_k. The K-rotor is not adiabatic and is assumed, according to the usual approximation,[65] to mix energy freely with the active vibrations. The internal rotations of fragments A and B are characterized by 2-D rotations with moments of inertia I_a and I_b, respectively, and an internal rotation with reduced moment of inertia I_r. The moments of inertia I_r and I_k are obtained by combining the K-rotors of the individual fragments, as described by Gilbert and Smith.[65]

In the restricted, or Hindered Gorin Model,[43,62,65] it is assumed that the two fragments interfere sterically with each another and thus cannot rotate freely. The effect is to reduce the available phase space and hence reduce the sum of states. Operationally, a "hindrance" parameter η is defined,[43] which can vary from zero (free rotation) to unity (completely hindered. The 2-D moments of inertia I_a and I_b are multiplied by the factor $(1-\eta)^{1/2}$ to obtain the effective 2-D moments of inertia used for calculating the sum of states. Examples of the Hindered Gorin Model used with MultiWell can be found elsewhere.[39,114]

`THERMO` (in the MultiWell Program Suite) has built-in capability for using the Hindered Gorin Model for fitting thermal recombination rate constants.

Required info:

- Experimental (or estimated) recombination rate constant
- Reaction enthalpy and K_{eq} are needed to obtain the rate constant for the reverse reaction.
- Vibrational frequencies and rotational constants of the recombination reactants.
- An assumed function for V(r) along the reaction coordinate (usually a Morse or Varshni potential) described by the center-of-mass distance between the two reactants.

Strengths:

- Relatively easy method.
- Generates TST parameters that fit the experimental canonical recombination rate constant at a given temperature.
- TST (i.e. RRKM) is a proven method; centrifugal corrections are employed.

Drawbacks:

- Experimental or estimated rate constants are required at every temperature.
- Fitting of the rate constants must be carried out at each temperature.
- TST parameters are different at each temperature.
- Separable harmonic oscillator and rigid rotor approximations may not be sufficiently accurate. Using separable hindered internal rotations is significantly more accurate, but more expensive.

Semi-empirical Canonical Variational Transition State Theory

We are currently using Semi-empirical Canonical Variational Transition State Theory (VTST) for applications where electronic structure methods are not reliable, but can still supply useful information. The basic idea is to use methods like Density Functional Theory (DFT; e.g. B3LYP) and large basis sets to carry out constrained optimizations at fixed distances along a reaction path in order to obtain the rotational constants and the vibrational frequencies orthogonal to the path, but then use empirical expressions for V(r), the potential energy along the path. For example, we have used $V(r) = -Cr^{-n}$, where C is a constant and n = 6.

The vibrational frequencies, rotational constants, and V(r) vary with position along the path. At each of the fixed points along the path, one can use those quantities as input to `THERMO` (in the MultiWell Program Suite) for computing a trial rate constant for the thermal reaction. The minimum trial rate constant corresponds to the VTST rate constant. The minimum rate constant is best obtained by interpolation between the fixed points, but for points separated by 0.1 Å or less it is almost as good to use the lowest trial rate constant, without interpolation. We find that by varying the constant C, it is possible to fit the experimental recombination rate constant at a given temperature (e.g. 298 K). According to our limited experience with this method, the value of the constant C fitted at a single temperature is in reasonable agreement with the temperature dependence of the recombination reaction, when known.

For a master equation simulation at temperature T, one uses the transition state parameters (vibrational frequencies, rotational constants, and V(r)) corresponding to the variational transition state at T.

Required info:

- Experimental (or estimated) recombination rate constant
- Reaction enthalpy and Keq are needed to obtain the rate constant for the reverse reaction.
- Constrained geometry optimizations at fixed distances along the reaction path; and corresponding vibrational frequencies and rotational constants.
- An assumed function for V(r) along the reaction coordinate (e.g. $V(r) = -C/r^{-n}$ with n=6).

Strengths:

- Uses theoretically calculated geometries and vibrational frequencies (which are more trustworthy than total energies).
- Generates TST parameters that fit the experimental canonical recombination rate constant at a given temperature.
- Temperature dependence of the rate constant can be estimated from fitting to the rate constant at a single temperature.
- TST (i.e. RRKM) is a proven method; centrifugal corrections are employed.

Drawbacks:

- Experimental or estimated rate constants are required at every temperature.
- Even for "economical" methods, the constrained optimizations and vibrational analysis can be expensive.
- For best accuracy, fitting of the rate constants must be carried out at each temperature. (However, fitting at a single temperature provides at least a rough estimate of the temperature dependence and may be adequate.)
- TST parameters are different at each temperature.
- Separable harmonic oscillator and rigid rotor approximations may not be sufficiently accurate. Using separable hindered internal rotations is significantly more accurate, but more expensive.

High and Low Pressure Limits

Regardless of the method for calculating k(E), the unimolecular rate constant at the high pressure limit $k_\infty(T_{trans})$ is calculated in MultiWell by using ρ(E) and k(E). The strong-collider rate constant at the low pressure limit is proportional to bath gas concentration: $k_0^{SC}(T_{trans})[M]$. The proportionality constant $k_0^{SC}(T_{trans})$ is calculated by using ρ(E) and k_c, the bimolecular rate constant for collisions (see Section A.4.3):

$$k_\infty(T_{trans}) = \frac{1}{Q(T_{trans})}\int_{E_0}^{\infty} k(E)\,\rho(E)\exp(-E/k_B T_{trans})dE \qquad (4.1\text{-}4a)$$

$$k_0^{SC}(T_{trans}) = \frac{k_{coll}}{Q(T_{trans})}\int_{E_0}^{\infty} \rho(E)\exp(-E/k_B T_{trans})dE \qquad (4.1\text{-}4b)$$

In these expressions, $Q(T_{trans})$ is the partition function of the reactant internal degrees of freedom (the degrees of freedom used to calculate $\rho(E)$ and $k(E)$) at translational temperature T_{trans}:

$$Q(T_{trans}) = \int_0^{\infty} \rho(E)\exp(-E/k_B T_{trans})dE \qquad (4.1\text{-}5)$$

The numerical integrations are carried out using the trapezoidal rule, because ρ(E) fluctuates wildly at low energies. Test show that the numerical integration produces values for $k_\infty(T_{trans})$ that are accurate within a fraction of 1% for usual values of ΔE_{grain}.[23] The activation energy is obtained by calculating $k_\infty(T_{trans})$ at two closely-spaced temperatures:

$$E_\infty = -R\frac{\ln\left[k_\infty(T_2)/k_\infty(T_1)\right]}{\left[T_2^{-1} - T_1^{-1}\right]} \qquad (4.1\text{-}6)$$

From the activation energy and the rate constant at one temperature, the A-factor (A_∞) can be calculated. Values for $k_\infty(T_{trans})$, E_∞, and A_∞ calculated in this way are reported (for each reaction) in the general output file. Note that the numerical values for the high pressure rate constant $k_\infty(T_{trans})$ (also, E_∞ and A_∞) should be the same, regardless of whether centrifugal corrections are used or not (Keywords `'CENT2'` or `'NOCENT'`), but $k_0^{SC}(T_{trans})$ depends on the particular choice. The value for $k_\infty(T_{trans})$ is formally identical to that given by canonical transition state theory.

Effects of Slow IVR

RRKM theory is based on the premise that energy is completely randomized on a time scale that is fast compared to chemical reaction. For most experiments, this condition appears to be met, but some examples of "intrinsic non-RRKM" behavior are known. For these reactions, the sow transfer of internal energy to the reaction coordinate from the other degrees of freedom limits the rate constant, which falls below the RRKM statistical limit. Several theories for the effects of slow IVR have been proposed,[31,115-119] but perhaps the most successful of these is the local random matrix model of Wolynes and coworkers.[8,9,120-122]

According to most of the IVR models,[8,119] the effects of slow IVR can be accounted for with an IVR transmission coefficient:

$$\kappa_{IVR}(U,[M]) = \frac{k_{IVR}^{q}(U) + k_{IVR}^{c}[M]}{k_{IVR}^{q}(U) + k_{IVR}^{c}[M] + v_{ivr}} \qquad (4.1\text{-}7)$$

where U is the total vibrational energy and $k^q_{IVR}(U)$ is the collision-free IVR rate constant, which must be calculated by one of the theories mentioned above. In particular, the Wolynes-Leitner[8] has been used with MultiWell.[123,124] For convenience in MultiWell, the total vibrational energy is measured from the reaction critical energy (*U* = *E* - *E*$_{0r}$), and $k^q_{IVR}(U)$ is expressed as a polynomial:

$$k^q_{IVR}(U) = C_{IVR,1} + C_{IVR,2}U + C_{IVR,3}U^2 \tag{4.1-8}$$

Parameter ν_{ivr} is a characteristic frequency for IVR (identified by Leitner and Wolynes[8] as the imaginary frequency for an isomerization reaction) and parameter k^c_{IVR} is the bimolecular collision rate constant (expressed in units of cm^3 s^{-1}) for collision-induced IVR. A threshold for IVR must also be specified: t_{IVR}, which in MultiWell is measured from the reaction critical energy. It can be used in cases where there is a sudden on-set of $k_{IVR}(U)$ at the threshold, followed by a quadratic increase with energy. The specific unimolecular rate constant as modified by IVR is finally given by the product of the transmission coefficient $\kappa_{ivr}(E,[M])$ and the specific rate constant $k_{RRKM}(E)$ calculated using RRKM theory, where *E* is the total vibrational energy measured from the zero point energy of the reactant, as usual:

$$k(E,[M]) = \kappa_{IVR}(E,[M]) \cdot k_{RRKM}(E) \tag{4.1-9}$$

The resulting specific (energy dependent) unimolecular rate constant depends on pressure, as indicated.

Using the Leitner-Wolynes model for IVR, simulations of data for trans-stilbene isomerization are in very good agreement with a large body of experimental data.[8,124] They indicate that collision-induced IVR occurs with a rate constant (k^c_{IVR}) similar in magnitude to the total collision rate constant k_q calculated using the method developed by Durant and Kaufman[24] (see the next section).

Tunneling: Unsymmetrical Eckart Barrier

[NOTE: for a far more sophisticated treatment of tunneling, see Semi-Classical Transition State Theory (SCTST) in Chapter 7 and Section 9.9.]

(This section on Eckart Tunneling contributed by Philip J. Stimac.)

Quantum mechanical tunneling corrections to the microcanonical rate constants k(E) have been implemented in MultiWell using a one dimensional unsymmetrical Eckart barrier.[125] The modification of k(E) within MultiWell is accomplished by evaluating the sum of states of the transition state according to Eq. 9 of the paper by Miller[126]:

$$N(E) = \int_{-V_o}^{E-V_o} P(E_1) N'(E - E_1) dE_1 \tag{4.1-10}$$

where V_o is the classical barrier height (in the direction of the forward reaction); E_1 is the energy in the reaction coordinate, relative to the top of the energy barrier; E is the total energy; $N'(E-E_1)$ is the density of states at energy $(E-E_1)$; $P(E_1)$ is the tunneling probability:

$$P(E_1) = \frac{\sinh(a)\sinh(b)}{\sinh^2((a+b)/2)\cosh^2(c)} \tag{4.1-11a}$$

$$a = \frac{4\pi}{h\nu_i} \frac{\sqrt{E_1 + V_o}}{(V_o^{-0.5} + V_1^{-0.5})} \tag{4.1-11b}$$

$$b = \frac{4\pi}{h\nu_i} \frac{\sqrt{E_1 + V_1}}{(V_o^{-0.5} + V_1^{-0.5})} \tag{4.1-11c}$$

$$c = 2\pi \sqrt{\frac{V_o V_1}{(h\nu_i)^2} - \frac{1}{16}}. \tag{4.1-11d}$$

In these equations, ν_i and V_1 are the imaginary frequency (related to the curvature of the saddle point on the potential energy surface at the transition state) and the classical barrier height relative to the products, respectively. The tunneling corrections to the sum of states of the transition state were applied using Eqs. (4.1-10) and (4.1-11) after transforming the integral in Eq. (4.1-10) to the expression

$$N(E) = \int_0^E P(E - V_o - E^\dagger) N'(E^\dagger) dE^\dagger. \tag{4.1-12}$$

$E^\dagger$ is the energy in the vibrational modes orthogonal to the reaction coordinate and the quantity $(E - V_o - E^\dagger)$ is the energy in the reaction coordinate. The primary difference between Equation (4.1-12) and Equation (4.1-10) is that the zero of energy is chosen to be at the minimum of the reactant in Equation (4.1-12), whereas the zero of energy is chosen to be at the transition state in Equation (4.1-10). Numerical tests showed that Equation (4.1-12) reproduces Fig. 2 of Miller.[126]

Equation (4.1-12) is evaluated in the subroutine Eckart within MultiWell. The integral is only evaluated at the grain energies. Evaluation of Equation (4.1-12) begins with all the energy in the reaction coordinate. This means that the tunneling probability is at a maximum while the density of states of the transition state is at a minimum. The integration continues until the tunneling probability $P(E - V_o - E^\dagger)$ become less than some cutoff value 'tunthresh', which is specified in the include file declare1.inc. The default value in declare1.inc is tunthresh=1.0E-12.

The k(E) calculated using the modified sums of states of the transition state reflect the tunneling effects. These tunneling corrected k(E) are also used to calculate the high-pressure rate constant k_∞, and are used to initialize the chemical activation distribution if both the CHEMACT and TUN keywords are selected. Please note that the TUN keyword cannot be used with the ILT or RKE keywords.

A.4.2 Competitive Pseudo-First-Order Reaction

In many practical systems, unimolecular and recombination reactions may be in competition with bimolecular reactions involving the same vibrationally excited species.[127] For example, a vibrationally excited species (A*) produced by an exothermic reaction may both undergo isomerization and react in a bimolecular reaction with another species (B), prior to collisional deactivation. If B is present in great excess, the pseudo-first-order approximation is applicable and one can define a pseudo-first order rate coefficient: $k^I = k_{bim}[B]$, where k_{bim} is the bimolecular rate coefficient and [B] is the concentration of the reaction partner B, which is present in great excess.

Prior to incorporation of this feature, Moriarity and Frenklach[128] used MultiWell for assessing several complicated reaction paths that may lead to aromatic ring formation in combustion systems. They found that certain vibrationally excited intermediates persist for relatively long periods and therefore *bimolecular* reactions between energized adducts and gaseous partners may need to be included in future calculations.

The procedure for implementation of a pseudo-first-order competitive reaction in MultiWell depends on whether it is assumed that k_{bim} is independent of the energy distribution of A*. In that case, k^I can be calculated using the canonical bimolecular rate constant: $k^I = k_{bim}$[B]. This value for k^I is then used to construct an external rate constant data file (with file name suffix ".rke"), which MultiWell will treat just like an ordinary unimolecular reaction. *Centrifugal corrections should not be employed for this reaction and the reaction should be treated as non-reversible.*

If it is assumed that the bimolecular rate constant depends on the vibrational energy distribution of species A*, then an energy dependent expression for $k_{bim}(E)$ must be used. In principle, this energy-dependent rate constant can be calculated from classical trajectories, quantum scattering, or other dynamical theories. However, since MultiWell is for the most part based on statistical theory, it is also appropriate to use microcanonical transition state theory.[129,130]

Theory

The discussion in this section was adapted from our recent paper on reactions of acetyl radical with O_2.[131]

Bimolecular reactions can be treated by using the bimolecular pseudo-first-order microcanonical approach. The microcanonical bimolecular rate constant is given by

$$k_{bim}(E^+) = \left[\frac{m^{\neq}_{AB}\ \sigma_A \sigma_B}{m_A m_B\ \sigma^{\neq}_{AB}} \right] \frac{g^{\neq}_{e(AB)}}{g_{e(A)} g_{e(B)}} \ \frac{1}{h} \frac{G^{\neq}_{AB}(E^+ - E_0)}{\rho_{AB}(E^+)} \qquad (4.2\text{-}1)$$

$$\rho_{AB}\left(E^+\right) = \int_0^{E^+} \rho_A\left(E^+ - x\right) \rho_B(x)\ dx \qquad (4.2\text{-}2)$$

where E^+ is the total rovibrational energy, $\rho_{AB}(E^+)$ is density of states of the A+B supermolecule, i.e. $\rho_{AB}(E^+)$ is the convolved density of states of the two reactants A and B, including all degrees of freedom except for the three coordinates of the supermolecule center of mass. The factor $G^{‡}_{AB}(E^+\text{-}E_0)$ is the sum of states of the transition state. The other symbols (m, g_e, and σ) have the same meaning as in Eq. 4.1-1. Here, however, the 2D-rotors are assumed to be active and hence are included with the K-rotor in calculating $\rho_{AB}(E^+)$ and $G^{‡}_{AB}(E^+\text{-}E_0)$. The canonical bimolecular rate constant can be calculated by averaging over the canonical energy distribution of the A+B supermolecule:

$$\left\langle k_{bim}(T) \right\rangle = \frac{\int_{E_0}^{\infty} k_{bim}(E^+)\ \rho_{AB}(E^+)\ e^{-E^+/RT} dE^+}{Q_{tot}(T)} \qquad (4.2\text{-}3)$$

where $Q_{tot}(T)$ is the total partition function of the supermolecule at temperature T:

$$Q_{tot}(T) = \int_0^{\infty} \rho_{AB}(E^+)\ e^{-E^+/RT} dE^+ \qquad (4.2\text{-}4)$$

We assume that reaction occurs when two conditions are met: E^+ is greater than the reaction threshold energy and the translational energy is greater than zero. For present purposes, we assume that the degrees of freedom of the supermolecule are partitioned into two groups. The degrees of freedom (DOF) in Group 1 are associated with the vibrations and K-rotor of excited species A* (i.e. the active DOF of species A) and all of the remaining DOF are collected in Group 2. We also assume that the energy distribution of the Group 2 models is thermal, while the energy of the Group 1 DOF is held fixed (adiabatic). Thus the total energy can be expressed as $E^+ = E_1 + E_2$. Note that other groupings can be postulated, depending on the system to be simulated, and centrifugal corrections may be included in future work.

From the partitioning of the DOF, a semi-microcanonical bimolecular rate constant that depends on E_1 and T is obtained by averaging over the canonical E_2 energy distribution:

$$\left\langle k_{bim}(E_1,T)\right\rangle_2 = \frac{\int_{E_{low}}^{\infty} k_{bim}(E_1+E_2)\ \rho_2(E_2)\ \exp\left(\frac{-E_2}{kT}\right)dE_2}{Q_2(T)} \qquad (4.2\text{-}5)$$

where E_{low} is the lower limit to the integral (the larger of 0 or (E_0 - E_1)), $\rho_2(E_2)$ is the density of states for the Group 2 DOF of the supermolecule and $Q_2(T)$ is the corresponding partition function:

$$Q_2(T) = \int_0^{\infty} \rho_2(E_2)\ \exp\left(\frac{-E_2}{kT}\right)\ dE_2 \qquad (4.2\text{-}6)$$

The canonical bimolecular rate constant can be obtained from $<k_{bim}(E_v,T)>_2$ by averaging over E_1:

$$\left\langle k_{bim}(T)\right\rangle = \frac{\int_0^{\infty}\left\langle k_{bim}(E_1,T)\right\rangle_2\ \rho_1(E_1)\ \exp\left(\frac{-E_1}{kT}\right)dE_1}{Q_1(T)} \qquad (4.2\text{-}7)$$

where $\rho_1(E_1)$ and $Q_1(T)$ are the density of states and the corresponding partition function, respectively, of the Group 1 DOF:

$$Q_1(T) = \int_0^{\infty} \rho_1(E_1)\ \exp\left(\frac{-E_1}{kT}\right)dE_1 \qquad (4.2\text{-}8)$$

Numerical integrations of Eq. 4.2-3 and of 4.2-7, carried out by the trapezoidal rule, give results at 300 K for the **A2** + O_2 reaction that agree to within ~1% when the energy grain is ≤5 cm^{-1}.[131]

The semi-microcanonical pseudo-first order rate constant (units of s^{-1}), which depends only on the active energy of reactant A*, is obtained by multiplying $<k_{bim}(E_1,T)>_2$ by [B], the concentration of the reactant that is present in great excess:

$$k^I(E,T) = \left\langle k_{bim}(E_1,T)\right\rangle_2\ [B] \qquad (4.2\text{-}9)$$

The rate constants $<k_{bim}(E_1,T)>_2$, $k_{bim}(E_1+E_2)$, and $<k_{bim}(T)>$ are compared in Figure 4.2.[131]

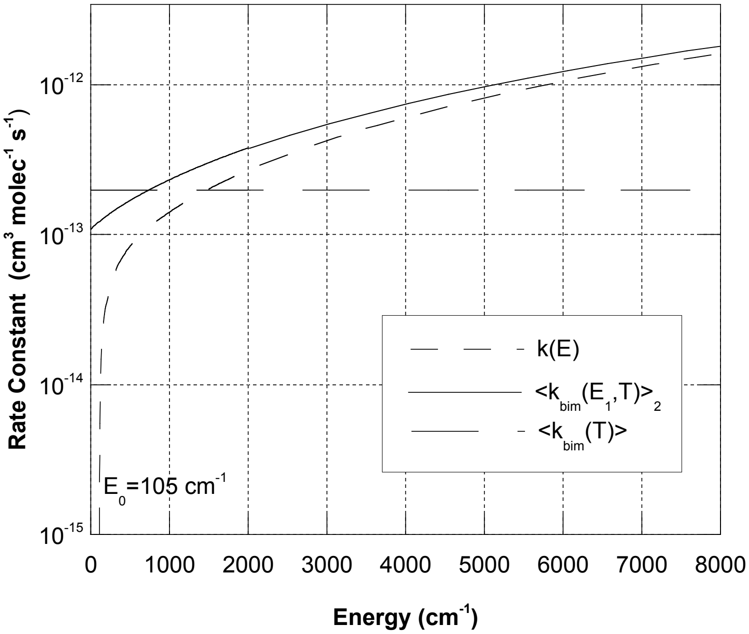


FIGURE 4.2: A comparison of k(E), $<k_{bim}(E_1,T)>_2$, and $<k_{bim}(T)>$ at T=298 K for the reaction **A2'** + O_2 → **B1** [see reference [131] for details].

In order to treat competition between unimolecular and bimolecular pathways, $k^I(E,T)$ is calculated and stored in an external data file, which is used as input to the MultiWell master equation computer program. Group 1 consists of the active DOF in species A. Thus $k^I(E,T)$ can be used directly in the master equation simulations as if it were an ordinary microcanonical rate constant for a unimolecular reaction. Care must be taken, however, to treat it as an irreversible reaction in MultiWell, since the detailed balance relations[22] have been coded only for unimolecular reactions. To include the reverse reaction would require using a theory for partitioning energy between unimolecular reaction products[95,132] in order to obtain the energy distribution in A (i.e. dissociation of the reaction product to regenerate A + B). Since such a treatment has not been implemented in MultiWell, one should treat the bimolecular reaction as irreversible. Treating such bimolecular reactions as irreversible may be a good approximation at ambient temperature, but a poor approximation at combustion temperatures.

In master equation simulations, unimolecular reaction, bimolecular reaction, and collisional deactivation of A* take place concurrently. Furthermore, the concentration of species B determines both the rate of collisional deactivation and the rate of the pseudo-first-order bimolecular reaction. Consequently, the importance of the microcanonical method is expected to be significant only at low pressures. After A* is thermalized by collisions, the microcanonical treatment coincides with the canonical approach.

DenSum Calculation

Using DenSum, it is easy to calculate the sum and density of states for the bimolecular reaction. For the transition state, all vibrations, torsions, and external rotations should be included in calculating the sum of states. The same is true in calculating the combined density of states of A+B. In addition, the 3-dimensional relative translation between A and B must also be included. *Once again, centrifugal corrections should not be employed for this reaction and the reaction should be treated as non-reversible in MultiWell.*

DenSum first uses the analytic convolution of classical translational states with the classical rotor states. The analytic expression for convolved classical trans-rot states is obtained as follows. First, note that the classical partition function is the Laplace transform of the classical density of states:[16,18] $Q(\beta) = \mathfrak{L}[\rho(E)]$, where β is the Laplace transform parameter and can be identified as $\beta = (k_BT)^{-1}$. The corresponding sum of states is $G(E) = \mathfrak{L}^{-1}[Q(\beta)/\beta]$, where the right hand side is the inverse Laplace transform. Now consider a microcanonical system with total energy E partitioned into two parts, the total sum of states is given by the convolution integral:[18]

$$G_{12}(E) = \int_0^E G_1(x)\rho_2(E-x)dx \tag{4.2-10}$$

where $G_1(x)$ is the sum of states for the first group of degrees of freedom, which contain energy x, and $\rho_2(E-x)$ is the density of states for the second group, which contain the remaining energy. This convolution can also be expressed as an inverse Laplace transform:[18]

$$G_{12}(E) = \mathfrak{L}^{-1}\left\{\frac{Q_1(\beta)Q_2(\beta)}{\beta}\right\} \tag{4.2-11}$$

where $Q_i(\beta)$ is the partition function of the ith group of modes. For p independent classical rotations, the partition function can be written[18,31]

$$Q_{rot}(T) = \prod_{i=1}^{p}\left\{\left(\frac{8\pi^2 I_i k_B T}{h^2}\right)^{d_i/2}\Gamma\left(\frac{d_i}{2}\right)\right\} = a_r\left(k_BT\right)^{r/2} \tag{4.2-12a}$$

$$r = \sum_{i=1}^{p} d_i \tag{4.2-12b}$$

where d_i is the dimensionality of the ith rotation, I_i is its moment of inertia, and the other symbols take their usual definitions.

For 3-D relative translation, the partition function can be written[18]

$$Q_{trans}(T) = \left\{\frac{2\pi\mu}{h^2}\right\}^{3/2} V\left(k_BT\right)^{3/2} = a_t\left(k_BT\right)^{3/2} \tag{4.2-13}$$

where μ is the reduced mass and V is the volume. Therefore the total sum of states for combined translation and rotation is obtained using Equation (4.2-11):

$$G_{rt}(E) = a_r a_t \mathfrak{L}^{-1}\left(p^{-1-(r+3)/2}\right) = \frac{2a_r a_t E^{(r+3)/2}}{(r+3)\Gamma\left(\frac{(r+3)}{2}\right)} \tag{4.2-14}$$

When no rotations are present, the sum of states is just the sum of translational states:[18,130]

$$G_t(E) = \frac{4\pi(2\mu E)^{3/2}}{3h^3}V \tag{4.2-15}$$

These expressions are used by DenSum to calculate the number of states in each energy bin when initializing the sums of states array, according to the Astholz et al.[29] modification of the Stein-Rabinovitch state-counting method.[10]

A.4.3 Collisions

Frequency of Inelastic Collisions

It is conventionally assumed that the inelastic collision frequency is the same as that experienced by molecules subject to a Lennard-Jones intermolecular potential. For the Lennard-Jones potential, k_c takes the following form:[133]

$$k_c = \pi\sigma^2 \langle v \rangle \Omega^{(2,2)*} \tag{4.3-1}$$

where <v> is the average speed at the translational temperature, σ is the Lennard-Jones diameter, and $\Omega^{(2,2)*}$ is the collision integral,[134] which depends on the Lennard-Jones parameters. Since only the product k_cP(E,E') appears in the master equation, if k_c is underestimated, then normalization of the step-size distribution is not appropriate. If, on the other hand, k_c is overestimated, then P(E,E') must include elastic collisions.[135] The inclusion of elastic collisions in the master equation causes no problems in principle, except to reduce the efficiency of certain numerical solutions. However, the fundamental question remains: is the frequency of inelastic collisions the same as the Lennard-Jones collision frequency?

Lawrance and Knight[136] used single vibrational level fluorescence and found that the observed total cross sections for inelastic collisions are in quantitative agreement with the Lennard-Jones collision frequency for a moderately high density of vibrational states. Classical trajectory calculations support this assumption,[137,138] but the argument is somewhat circular in this case since the assumed potential energy functions are often constructed from pair-wise Lennard-Jones potentials. Recently, Xue et al.[139] used quantum beat spectroscopy to investigate a single vibrational level of SO_2 at high vibrational energy, and found cross sections substantially greater than predicted by the Lennard-Jones interaction potential. However, in the sparse density of states regime at low vibrational energies it is well known that the inelastic collision cross section is small[140] and thus the total inelastic collision rate constant is probably smaller than k_{LJ}.

A rigorous upper limit to k_c is provided by the total collision rate constant k_q, which is based on the total quantum cross section. Because of concern about the proper choice of k_c and normalization of the step size distribution (see below), MultiWell provides an option for utilizing the total collision rate constant, which can be estimated from Lennard-Jones parameters by using the method of Durant and Kaufman.[24]

In a new development,[5] it is now assumed in MultiWell that the rate constant for *inelastic collisions* depends on the internal energy, reflecting the reduced rate constant expected at lower energies where the density of states is sparse. This new development is described in the following section.

Normalization

The rate coefficient *R(x,y)* is conventionally written as the product of the total vibrationally inelastic collision frequency $k_c(y)$[M] multiplied by the "collision step-size distribution", *P*(*x*,*y*), which expresses the probability density that a molecule initially with initial energy *y* will undergo an inelastic transition to the energy range *x* to *x*+*dx*:

$$R(x,y)dx = \int_0^\infty R(x,y)dx \left\{ R(x,y)dx \Big/ \int_0^\infty R(x,y)dx \right\}, \tag{4.3-2a}$$

$$= k_c(y)P(x,y)dx \tag{4.3-2b}$$

The first factor on the right hand side of Equation (4.3-2a), the integral over the rates of all inelastic transitions from initial energy *y*, is the frequency of inelastic collisions, $k_c(y)$[M] and the second factor (in curly brackets) is *P(x,y)dx*. Note that *P(x,y)* is normalized:

$$\int_0^\infty P(x,y)\,dx = 1 \tag{4.3-3}$$

It is important to emphasize that the factorization of *R(x,y)* in Equation (4.3-2) is merely for convenience and that $k_c(y)$[M] and *P(x,y)* never occur independently of one another. Furthermore, *P(x,y)* only has an unambiguous physical interpretation when $k_c(y)$[M] is *exactly* equal to the total inelastic collision rate constant. Since the exact inelastic collision frequency is not known, the inevitable errors in $k_c(y)$ are compensated in part by errors in *P(x,y)*, when experimental data are fitted to this prescription. Thus it is important to use $k_c(y)$ and *P(x,y)* in a matched pair whenever possible.[16,31,65]

By considering detailed balance at equilibrium in the absence of reactions, the relationship between *R(x,y)* and *R(y,x)* can be found. Detailed balance requires that in every increment of energy, the rates of forward and reverse processes must balance. The Detailed Balance relationship between the probability densities for up- and down-collisions is given by

$$\frac{P(x,y)}{P(y,x)} = \frac{k_c(x)}{k_c(y)}\frac{\rho(x)}{\rho(y)}\exp\left[-(x-y)/k_BT\right] \tag{4.3-4}$$

The total probability density for an energy changing collision is normalized (see Eq. 4.3-3) and can be written as the sum of two integrals corresponding to down- and up-collisions:

$$1 = \int_0^y P(x,y)\,dx + \int_y^\infty P(x,y)\,dx \tag{4.3-5}$$

In order to construct a normalized collision step size distribution (the probability density), it is common practice to specify a (dimensionless) non-normalized function *f(x,y)*, which is assumed to be proportional to *P(x,y)*:

$$P(x,y) = \frac{f(x,y)}{N(y)} \tag{4.3-6}$$

where *N(y)* is a normalization constant. With this definition, the normalization equation becomes

$$1 = \int_0^y \frac{f(x,y)}{N(y)}\,dx + \int_y^\infty \frac{f(x,y)}{N(y)}\,dx \tag{4.3-7}$$

After rearranging this expression, we obtain a formal expression for the normalization constant *N(y)*, expressed as a sum of integrals:

$$N(y) = \int_0^y f(x,y)\,dx + \int_y^\infty f(x,y)\,dx \tag{4.3-8a}$$

$$= N_d(y) + N_u(y) \tag{4.3-8b}$$

where subscripts *d* and *u* denote down-steps ($x<y$) and up-steps ($x>y$), respectively.

For convenience, the un-normalized function $f(x,y)$ is usually specified for down-steps, but one could choose to specify a function for up-steps instead. We will follow convention and specify the function for down-steps, $f(x,y) = f_d(x,y)$ with $x<y$. Thus $N_d(y)$ is easily evaluated:

$$N_d(y) = \int_0^y f_d(x,y)\,dx \tag{4.3-9}$$

and $N_u(y)$ can be expressed in terms of $f_d(x,y)$ by the detailed balance relationship:

$$N_u(y) = \int_y^\infty f_d(y,x)\frac{N(y)}{N(x)}\frac{k_c(x)}{k_c(y)}\frac{\rho(x)}{\rho(y)}\exp\left[-(x-y)/k_BT\right]dx \tag{4.3-10}$$

If we had assumed that $f(x,y)$ was specified for up-steps, an analogous procedure would be followed.

Since $N(x)$ appears in the integral expression for $N_u(y)$, the solution of Eq. 4.3-10 is not completely straightforward. Normalization constant $N(y)$ can be found by using trial values for $N(x)$ and employing an iterative solution[1] of Eq. 4.3-10, or by rearranging the equation as follows:

$$N(y) = \frac{\displaystyle\int_0^y f_d(x,y)\,dx}{1-\displaystyle\int_y^\infty \frac{f_d(y,x)}{N(x)}\frac{k_c(x)}{k_c(y)}\frac{\rho(x)}{\rho(y)}\exp\left[-(x-y)/k_BT\right]dx} \tag{4.3-11}$$

Eq. 4.3-11 can be solved with the finite difference algorithm described by Gilbert and coworkers [65,141]. Both of these approaches to finding $N(y)$ are based on specifying $f_d(x,y)$ and requiring that $N(y)$ first be estimated at very high energies, well above the energies of interest, where $N_d(y)$ and $N_u(y)$ tend to become independent of energy (at least when the average energy transferred per collision is independent of energy).

As discussed in the previous section, $k_c(y)$ is expected to depend on the initial energy, *y*, but it is common practice to assume that the inelastic collision rate constants are independent of internal energy [1,18,31,141]. This constant is conventionally identified with k_{LJ}, the bimolecular rate constant for collisions between particles governed by a Lennard-Jones intermolecular potential.[18,31,65]

Experience has shown that the assumption the k_c is independent of energy leads to problems with normalization.[5] Iterative normalization[1] converges reasonably rapidly at high energies, but problems emerge at low energies, where the density of states is sparse and has large relative fluctuations. The problems are most severe when an energy grain that contains just a few states is bracketed on both sides by energy grains containing much higher densities of states. For these cases, the normalization factors for some of the energy grains tend to diverge, instead of converging during the iterative calculation. Because of this problem, it was necessary to limit the number of iterations to e.g. 2-5, so that normalization at high energy converges sufficiently, while normalization at low energy does not diverge too much. This strategy, although not completely satisfactory, is reasonably effective in producing steady-state energy distributions that simulate the equilibrium Boltzmann distribution.

Related problems arise at low energies when using the finite difference algorithm of Gilbert and coworkers.[65,141] The Gilbert algorithm is quite general, but in practice the pragmatic assumption is made that the collision frequency is independent of excitation energy. These problems result in normalization constants that are negative and therefore un-physical in some energy grains.

Starting with MultiWell v.2009.0, we have incorporated a new treatment of collisions that solves the problems outlined above.[5]

In principle it should be possible to express $k_c(y)$ as a function of $N(y)$, but the specific functional dependence is not known. In the absence of specific knowledge about $k_c(y)$, we make the simplest possible assumption: that $k_c(y)$ is directly proportional to $N(y)$.[5] With this assumption, the following ratio, which appears in Eq. 4.3-10, equals unity

$$\frac{N(y)}{N(x)}\frac{k_c(x)}{k_c(y)} = 1 \tag{4.3-12}$$

The assumption that $k_c(y)$ is directly proportional to $N(y)$ requires that the collision frequency for a molecule with excitation energy y be calculated using $k_c(y) = C\bullet N(y)$ (see Eq. 4.3-12), where the proportionality constant C must be established by some other means. Since the Lennard-Jones rate constant k_{LJ} conventionally has been used for calculating the low pressure limit of unimolecular and recombination reactions, we have adopted the following form for the total energy-dependent rate constant for the inelastic collision frequency:

$$\frac{N(y)}{N(x)}\frac{k_c(x)}{k_c(y)} = 1 \tag{4.3-13}$$

where E_{ref} is a reference energy.

In reaction studies, the rate of energy transfer is most important at energies near the reaction critical energy. Thus we identify E_{ref} with the critical energy of the lowest reaction threshold energy (when multiple reaction channels are involved) that is higher in energy than the energy boundary (parameter Emax1) between the lower and upper portions of the double array in MultiWell. If no reaction threshold energies are below Emax1, we arbitrarily specify E_{ref} as equal to Emax1. At Emax1, $N(y)$ is a relatively smooth function and the density of states is typically >10-100 states/cm^{-1}. Except at low energies, the numerical results obtained using this new approach are nearly the same as those obtained using the old conventional approach.[5]

For convenience in the Monte Carlo selection of step sizes, both the normalization factor $N(E')$ and the probability of an activating collision P(E,E') =[$N_a(E')/N(E')$] are stored in double arrays for each well. At low state densities, P(E,E') exhibits random fluctuations and some energy grains may contain no states, while the function is quite smooth at high energies. Since it is desirable to be able to use arbitrary functions for the collision step size distribution, it is not feasible to employ analytic expressions for the integrals in the normalization equation, which would allow much shorter computer execution times. In fact, several approximate analytical expressions were tested, but none was sufficiently accurate in the sparse density of states regime. For this reason, normalization is carried out numerically using the open-ended trapezoidal rule, which is a particularly robust algorithm.[90] For low energies, the energy step size is set equal to that used in the lower energy portion of the double arrays (ΔE_{grain}). At higher energies, the energy step size is set equal to a fraction (typically 0.2) of the magnitude of a characteristic energy transfer step:

$$\delta E_d = \left| \frac{d(\ln f_d(E,E'))}{dE} \right|^{-1}, \qquad \text{for E' > E,} \tag{4.3-14a}$$

$$\delta E_a = \left| \frac{d(\ln f_a(E',E))}{dE} \right|^{-1}, \qquad \text{for E' < E,} \tag{4.3-14b}$$

where $f_a(E,E')$ and $f_d(E,E')$ were defined above. For the exponential model, δE_d is equal to $\alpha(E)$, which varies with internal energy. In general, both the characteristic energy length and the integration step size vary with energy.

Collision Step-Size Distribution

Many step-size distribution models have been used in energy transfer studies and there is still considerable uncertainty about the appropriate collision model and functional form of $P(E,E')$.[25,133,142] Note that $P_d(E,E')$ for de-activating collisions is expressed in terms of an un-normalized function $f_d(E,E')$ and normalization factor $N(E)$:

$$P_d(E,E') = \frac{1}{N(E')} f_d(E,E') \qquad \text{for } E' > E, \tag{4.3-15}$$

To offer a wide selection, MultiWell includes a number of different optional functional forms for $f_d(E,E')$, including biexponential, Gaussian, Weibull distribution, etc. The best information currently available suggests that a generalized exponential function is most appropriate for deactivation steps:[143]

$$f_d(E,E') = \exp\left\{ - \left| \frac{E' - E}{\alpha(E')} \right|^{\gamma} \right\}, \qquad \text{for } E' > E, \tag{4.3-15}$$

where $\alpha(E)$ is a linear function of vibrational energy, and γ is a parameter that ranges from ~0.5 to ~1.5. The corresponding expression for activation collisions is obtained from detailed balance. When the parameter γ is less than unity, the wings of the step-size distribution have enhanced relative probabilities that qualitatively resemble the bi-exponential distribution. When γ=1, Equation (4.3-15) gives the venerable exponential model.

Monte Carlo Selection of Step-Size

Two random numbers are used for selecting the collision step size. The first random number selects activating, or deactivating collisions by comparison to the up-transition probability $P_{up}(E)$:

$$P_{up}(E) = N_a(E) / N(E) \qquad \text{up-transition probability} \tag{4.3-16}$$

$$0 \le r_3 < P_{up}(E), \qquad \text{activating} \tag{4.3-16a}$$

$$P_{up}(E) \le r_3 < 1, \qquad \text{de-activating} \tag{4.3-16b}$$

To select the step-size, the second random number is used with the cumulative distribution for $P(E,E')$ to find the final energy E, given initial energy E':[1]

$$r_4 = \frac{1}{N_a(E')} \int_{E'}^{E} f_a(x,E')dx\,, \qquad \text{activating} \tag{4.3-17a}$$

$$r_4 = \frac{1}{N_d(E')}\int_{E'}^{E} f_d(x,E')dx\,, \qquad \text{de-activating} \qquad (4.3\text{-}17b)$$

The integrals are evaluated by the trapezoidal rule, just as described in the preceding section, until the equalities in Equation (4.3-17) are satisfied. In the high energy regime, this is accomplished by integrating step-by-step until an integration step gives a value for the right hand side of Equation (4.3-17) that is larger than r_4. Linear interpolation is then used to find the value of final energy *E* that satisfies the equality. In the low energy regime, the integration is carried out step-by-step to find the energy step which gives the best agreement between the LHS and right hand side of Equation (4.3-17). Note that the normalization integrals in the low energy regime are stored in the lower energy portion of the double arrays. In the high energy regime, the normalization integrals are found by interpolation of values stored the high energy portion of the double arrays.

Occasionally, the normalization integrals are overestimated due to imperfect interpolation and thus the equalities in Equation (4.3-17) cannot be satisfied. In such a case, the integral is evaluated step-by-step until the additional partial sum is less than a selected relative error (typically 10^{-6}). This procedure yields an explicitly calculated value for the normalization integral. The interpolated normalization integral is then replaced with this new value and the energy step selection process is repeated. This procedure is somewhat cumbersome and computationally intensive, but it was found to produce more accurate thermal distribution functions.

A.4.4 Other Processes

Additional processes can be incorporated into MultiWell calculations by using the capability of reading rate constants from external data files.

Several processes have been neglected in the present version of MultiWell. For example, spontaneous infrared emission[144] by the vibrationally excited species, which is particularly important at low pressure,[3] has not been included. Similarly, stimulated emission, which is important in laser-induced chemical reactions,[145,146] has also been neglected. Future versions of MultiWell may include these processes, especially if the kinetics community expresses an interest in them.

A.5. Initial Conditions

At the start of each stochastic trial, initial conditions must be specified. MultiWell selects the initial energy *via* Monte Carlo selection techniques that are based on the cumulative distribution function corresponding to a selected physical process. It is assumed that the reactant is at infinite dilution in a heat bath and thus there are no temperature changes due to reaction exothermicity or energy transfer. For most laboratory experiments, this is an acceptable approximation.

A.5.1 Monte Carlo Selection of Initial Energies

Monte Carlo selection of the initial internal energy is carried out by equating random number r_5 to the cumulative distribution function $Y_0(E)$ corresponding to a given initial energy density distribution $y_0(E')$:

$$r_5 = Y_0(E) = \int_0^E y_0(E')dE' \tag{5.1-1}$$

where E' is the integration variable. In MultiWell, $Y_0(E)$ is found by rectangular rule in the lower portion of the double array and by trapezoidal integration in the upper portion; the values are stored as a function of initial energy in a linear array. Jsize (which user-selected) array elements are used to cover the relevant energy range. For a thermal distribution (see below), the relevant energy range is assumed to be ~20k_BT. The Monte Carlo selection is carried out by interpolating in the stored array to find the value of E at which $Y_0(E)$ = r_5. Interpolation in this fashion is much more computationally efficient than calculating the integral in Equation (5.1-1) for each stochastic trial.

A.5.2 Optional Initial Energy Density Distributions

The initial energy density distributions that are included as options in MultiWell are described here. In addition to these choices, there is also a provision for providing a user-defined double array of $Y_0(E)$ values and for a delta function (which does not require Monte Carlo selection). Examples of user-defined functions include prior distributions[94,107,147] and energy distributions that are the result of bond fission.[148]

Thermal Activation

In an ordinary thermal unimolecular reaction system that takes place at infinite dilution, the translational and vibrational temperatures are equal and do not change during reaction (T_{trans}=T_{vib}). For shock wave simulations, it is assumed that T_{trans} changed instantaneously when the shock occurred and therefore is elevated at t=0, but T_{vib} remains at the temperature that described the thermal system prior to the shock. Subsequent vibrational energy transfer collisions cause the internal energy to increase. The only difference between shock tube and isothermal simulations is that in the former, the two temperatures are unequal. In both cases, the initial internal energy distribution function is a Boltzmann distribution characterized by T_{vib}. The probability of the initial energy E falling in the range between E and E+dE is given by the probability density function

$$y_0^{(therm)}(E)dE = \frac{\rho(E)e^{-\frac{E}{k_BT_{vib}}}dE}{\int_0^\infty \rho(E')e^{-\frac{E'}{k_BT_{vib}}}dE'} \tag{5.2-1}$$

Single Photon Photo-Activation

The energy distribution produced by absorption of a single photon is assumed to be described by the thermal population at the ambient vibrational temperature added to the energy of the photon ($h\nu$). Hence, the probability density function for photo-activation is given by Equation (5.2-1) and the selected thermal energy is then increased by $h\nu$.

Chemical Activation and Recombination Reactions

Chemical activation is the process by which a single vibrationally excited species C(E) is produced from the bimolecular reaction of two precursor species (A and B):

$$A + B \rightarrow C(E) \tag{5.2-2}$$

where E is the vibrational energy. The excited molecule C(E) can then react by passing back out of the entrance channel, or proceeding forward through the product channel, as shown in Fig. 5.1. The energy distribution of C(E) before it has a chance to react or be collisionally stabilized is the known as the chemical activation distribution function. It is appropriate for any recombination reaction that takes place under thermal conditions.

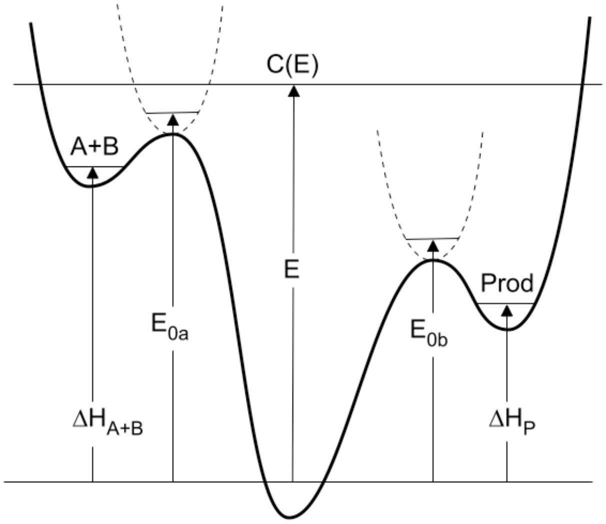


Figure 5.1. Energy diagram for chemical activation.

The chemical activation distribution function is obtained from the reverse reaction by using detailed balance.[16,18,30,31,65] The reverse reaction is the unimolecular decomposition reaction with rate constant $k_a(E)$ that produces the product set A+B. The index a specifies the particular unimolecular reaction channel C(E) → A + B. The resulting probability density function is a thermal distribution weighted by $k_a(E)$. The probability density function and corresponding Monte Carlo selection expression are as follows:

$$y_0^{(ca,i)}(E)dE = \frac{k_a(E)\rho(E)e^{-\frac{E}{k_B T_{vib}}}dE}{\int\limits_{E_{0a}}^{\infty} k_a(E')\rho(E')e^{-\frac{E'}{k_B T_{vib}}}dE'}, \quad \text{for } E \geq E_{0a} \tag{5.2-3}$$

$$r_5 = \int\limits_{E_{0a}}^{E'} y_0^{(ca,i)}(E)dE \tag{5.2-4}$$

where the lower limits of the above equations are equal to E_{0a}, the unimolecular reaction threshold energy. The density of states ρ(E) is for the molecule C. The trapezoidal rule is used in the selection procedure, as described above for thermal activation.

A recombination reaction produces a recombination product, which is a chemically activated species. The chemically activated recombination product C(E) can react *via* the reverse of Reaction (5.2-2), and possibly by other unimolecular pathways, in competition with collisional energy transfer. Several quantities may be of interest, including branching ratios, net rates of reaction to produce specific final products, etc. In all cases, the first step is to simulate the reactions of the chemically activated recombination product C(E) under the desired conditions of temperature and pressure. The results of the simulation can be used in various ways to find the quantities of interest.[23]

The total rate constant for the recombination reaction at the high pressure limit is obtained from detailed balance by using the equilibrium constant $K(T_{trans})$ at translational temperature T_{trans}:

$$k_{rec,\infty} = k_{uni,\infty} / K(T_{trans}) \tag{5.2-5}$$

where $k_{rec,\infty}$ and $k_{uni,\infty}$ are the high pressure limiting recombination and unimolecular decomposition rate constants, respectively; the latter of these is calculated and reported in the MultiWell standard output. The equilibrium constant $K(T_{trans})$ is calculated using the program Thermo (part of the MultiWell computer program suite), which employs standard statistical mechanics formulas[149,150] for the partition functions of the reactants A and B.

To calculate the over-all rate constant for producing the i^{th} product, the relative population (fraction) f_i of that species at the end of the simulation is multiplied by $k_{rec,\infty}$:

$$k = f_i\, k_{rec,\infty} = f_i\, k_{uni,\infty} / K(T_{trans}) \tag{4.2-6}$$

This procedure is appropriate whether or not there is an intrinsic energy barrier for the recombination reaction.

A.6. Input

A6.1 Major Options

Densities of States: ρ(E)

Densities of states for the wells are provided in an external file (in the form of a double array). DenSum is provided as a tool to calculate sums and densities of states according to the Whitten-Rabinovitch approximation,[27,28] or according to the Stein-Rabinovitch method[10] of exact counts. Molecular assignments for use in the current version of DenSum can be expressed in any combination of separable harmonic oscillators, Morse oscillators, and free rotors. For non-separable degrees of freedom, other approaches will be needed (see references [47,151,152], for example). The moments of inertia needed for calculating rotational constants are evaluated with the program MomInert. This code requires Cartesian coordinates for the molecular structure. Such structures can be calculated with good accuracy by using quantum chemistry programs.

Specific Unimolecular Rate Constants: k(E)

Specific rate constants are needed for each reaction. There are three ways to provide rate constants: a) they may be calculated internally via the Inverse Laplace Transform (ILT) method, b) the sums of states can be provided in an external file, and c) the *k(E)* values can be provided in an external file. Data provided in an external file is in the form of a double array with energy origin at the reaction threshold energy. The double array allows high energy resolution near the reaction threshold where it is most important. For most purposes, it is most efficient to use DenSum, which calculates sums of states ($G^{\ddagger}(E\text{-}E_0)$) and generates an external file suitable for input into MultiWell. However, Densum is only suitable for fixed transition states with separable degrees of freedom and therefore other methods must be used to calculate *$G^{\ddagger}(E-E_0)$* or *$k(E-E_0)$* for non-separable and flexible transition states. If the reaction is a reversible isomerization reaction, MultiWell uses the same external data file to calculate *$k(E-E_0)$* for both forward and reverse reactions. By using the same external file for both forward and reverse reactions, the reversible isomerization rates are internally self-consistent.

A6.2 Properties of Wells and Transition States

Energies (e.g. $\Delta H_f°$ at 0 K) are required for all wells and transition states, in order to establish the relative energies of isomers and reaction thresholds. Moments of inertia are needed for the inactive degenerate two-dimensional external rotation. Energy transfer parameters are needed for each well and MultiWell does not require that they be the same for all wells. One would expect the energy transfer parameters for a cyclic species to differ from those of a linear isomer. However, to the best of my knowledge the energy transfer parameters are not known for more than one isomer in any system. Until additional information becomes available, it is pragmatic to assume that all isomers have the same energy transfer parameters.

A.7. Output

MultiWell generates several output files that summarize the input data and the calculation results.

A.7.1 multiwell.out

This general output file summarizes the input parameters, thermochemistry, high pressure limit rate constants for each reaction, time-dependent average fractional populations (with standard deviations from Equation (11)), and average vibrational energies. The time-dependent quantities are the instantaneous (snapshot) values averaged over N_{trials} stochastic trials: they are not averaged over the time interval, as was done in previous master equation codes from this laboratory.[1,2,4]

A.7.2 multiwell.rate

This file stores the time-dependent output of average unimolecular rate "constants" (which vary with time in non-steady-state systems) for every reaction pathway:

$$< k_j(t) >= \frac{1}{N_{trials}} \sum_{i=1}^{N_{trials}} k_j(E_i(t)) \qquad (7.1\text{-}1)$$

where *j* designates the reaction channel. Many trials are needed to accumulate good statistics. To improve statistics, the binned results correspond to the number of visits to the bin (which can be many times larger than N_{trials}) and thus are averaged over the time-duration of the bin. Note that that this averaging method differs from the snapshot method described above, where the number of snapshots is equal to $N_{trials.}$ In an equilibrium thermal system, *<k>* is independent of time and equal to the average unimolecular rate constant *$k_{uni}(T)$*. In non-equilibrium systems, *<k(t)>* varies with time and relaxes to a constant value as the system itself undergoes relaxation. As relaxation takes place, some reactions achieve steady-state, which is apparent as *<k(t)>* approaches a constant value. Thus, this output file is useful for several purposes, including monitoring relaxation and the approach to steady state.

A.7.3 multiwell.flux

This file stores the average time-dependent chemical "flow" along each reaction path:

$$< F(t) >= f_{react} < k(t) > \tag{7.2-1}$$

where f_{react} is the time-dependent average fractional concentration of the reactant species and *<k(t)>* is the average unimolecular rate constant described above. When two reactions come into pseudo-equilibrium with one another, their fluxes are equal. Thus, this output file is useful for several purposes, including monitoring the evolution toward equilibrium and diagnosing pseudo-equilibrium conditions.

A.7.4 multiwell.dist

This file stores time-dependent vibrational distributions within each well. Only the non-zero array elements are tabulated. Many trials are needed to accumulate good statistics and thus the binned results correspond to the number of visits to the bin (which can be many times larger than N_{trials}) and are averaged over the time-bin. To limit the size of this file, each (default) time bin is set at ten times that of the time-bins used for the other time-dependent output.

A.7.5 multiwell.array

This file tabulates all energy-dependent input data, including densities of states, specific rate constants for every reaction, collision up-transition probabilities and normalization factors, and initial energy distributions.

A.8. Concluding Remarks

MultiWell calculates time-dependent concentrations, yields, vibrational distributions, and rate constants as functions of temperature and pressure for unimolecular reaction systems that consist of multiple stable species and multiple reaction channels interconnecting them. Users may supply unimolecular reaction rates, sums of states and densities of states, or optionally use the Inverse Laplace Transform method. For weak collision effects, users can select different collision models for down-steps including exponential, biexponential, generalized exponential, etc., and user-defined functions.

The code is intended to be relatively easy to use. It is designed so that even the most complicated unimolecular reaction systems can be handled via the data file without restructuring or recompiling the code.

MultiWell is most suitable for time-dependent non-equilibrium systems. The real time needed for a calculation depends mostly upon the number of collisions during a simulated time period and on the number of stochastic trials needed to achieve the desired precision. For slow reaction rates and precise yields of minor reaction products, the code will require considerable computer time, but it will produce results. For long calculation runs, we often just let the simulation run overnight or over a weekend.

References

[1] J. R. Barker, Chem. Phys. **77** (2), 301 (1983).

[2] J. Shi and J. R. Barker, Int. J. Chem. Kinet. **22**, 187 (1990).

[3] J. R. Barker, J. Phys. Chem. **96**, 7361 (1992).

[4] J. R. Barker and K. D. King, J. Chem. Phys. **103**, 4953 (1995).

[5] J. R. Barker, Int. J. Chem. Kinetics **41**, 748 (2009).

[6] T. L. Nguyen and J. R. Barker, J. Phys. Chem. A **114**, 3718 (2010).

[7] T. L. Nguyen, J. F. Stanton, and J. R. Barker, Chem. Phys. Letters **499**, 9 (2010).

[8] D. M. Leitner, B. Levine, J. Quenneville, T. J. Martinez, and P. G. Wolynes, J. Phys. Chem. A **107** (49), 10706 (2003).

[9] D. M. Leitner and P. G. Wolynes, Chem. Phys. Lett. **280** (5-6), 411 (1997).

[10] S. E. Stein and B. S. Rabinovitch, J. Chem. Phys. **58** (6), 2438 (1973).

[11] T. Beyer and D. F. Swinehart, Comm. Assoc. Comput. Machines **16** (6), 379 (1973).

[12] W. H. Miller, J. Chem. Phys. **62**, 1899 (1975).

[13] W. H. Miller, Faraday Discuss. Chem. Soc. **62**, 40 (1977).

[14] W. H. Miller, R. Hernandez, N. C. Handy, D. Jayatilaka, and A. Willets, Chem. Phys. Letters **172**, 62 (1990).

[15] R. Hernandez and W. H. Miller, Chem. Phys. Lett. **214** (2), 129 (1993).

[16] W. Forst, *Theory of Unimolecular Reactions*. (Academic Press, New York, 1973).

[17] W. Forst, J. Phys. Chem. **76** (3), 342 (1972).

[18] W. Forst, *Unimolecular Reactions. A Concise Introduction*. (Cambridge University Press, Cambridge, 2003).

[19] D. T. Gillespie, J. Phys. Chem. **81** (25), 2340 (1977).

[20] D. T. Gillespie, J. Comp. Phys. **22** (4), 403 (1976).

[21] D. T. Gillespie, J. Comp. Phys. **28** (3), 395 (1978).

[22] J. R. Barker, Int. J. Chem. Kinetics **33** (4), 232 (2001).

[23] J. R. Barker and N. F. Ortiz, Int. J. Chem. Kinetics **33** (4), 246 (2001).

[24] J. L. Durant and F. Kaufman, Chem. Phys. Lett. **142** (3-4), 246 (1987).

[25] J. R. Barker, L. M. Yoder, and K. D. King, J. Phys. Chem. A **105** (5), 796 (2001).

[26] J. R. Barker and R. E. Weston, Jr., J. Phys. Chem. A **114**, 10619 (2010).

[27] G. Z. Whitten and B. S. Rabinovitch, J. Chem. Phys. **38**, 2466 (1963).

[28] G. Z. Whitten and B. S. Rabinovitch, J. Chem. Phys. **41**, 1883 (1964).

[29] D. C. Astholz, J. Troe, and W. Wieters, J. Chem. Phys. **70** (11), 5107 (1979).

[30] P. J. Robinson and K. A. Holbrook, *Unimolecular Reactions*. (Wiley-Interscience, London; New York, 1972).

[31] K. A. Holbrook, M. J. Pilling, and S. H. Robertson, *Unimolecular Reactions*, 2 ed. (Wiley, Chichester, 1996).

[32] J. L. McHale, *Molecular Spectroscopy*. (Prentice Hall, Upper Saddle River, 1999).

[33] A. L. L. East and L. Radom, J. Chem. Phys. **106** (16), 6655 (1997).

[34] NIST, (National Institute of Science and Technology, 2003).

[35] M. W. Chase, Jr., J. Phys. Chem. Ref. Data **Monograph No. 9**, 1 (1998).

[36] D. M. Golden, J. Phys. Chem. A **111** (29), 6772 (2007).

[37] D. M. Golden, J. Phys. Chem. A **110** (9), 2940 (2006).

[38] D. M. Golden, Int. J. Chem. Kinet. **37** (10), 625 (2005).

39 D. M. Golden, J. R. Barker, and L. L. Lohr, J. Phys. Chem. A **107** (50), 11057 (2003).

40 D. M. Golden, Int. J. Chem. Kinet. **35**, 206 (2003).

41 D. M. Golden and G. P. Smith, J. Phys. Chem. A **104** (17), 3991 (2000).

42 R. Patrick and D. M. Golden, Int. J. Chem. Kinet. **15**, 1189 (1983).

43 G. P. Smith and D. M. Golden, Int. J. Chem. Kinet. **10** (5), 489 (1978).

44 Y. P. Varshni, Rev. Mod. Phys. **29**, 664 (1957).

45 M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, V. G. Zakrzewski, J. Montgomery, R. E. Stratmann, J. C. Burant, S. Dapprich, J. M. Millam, A. D. Daniels, K. N. Kudin, M. C. Strain, O. Farkas, J. Tomasi, V. Barone, M. Cossi, R. Cammi, B. Mennucci, C. Pomelli, C. Adamo, S. Clifford, J. Ochterski, G. A. Petersson, P. Y. Ayala, Q. Cui, K. Morokuma, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. Cioslowski, J. V. Ortiz, A. G. Baboul, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. Gomperts, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, J. L. Andres, C. González, M. Head-Gordon, E. S. Replogle, and J. A. Pople, Gaussian 98, Revision A.7 (Pittsburgh, 1998).

46 M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery, Jr., T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, and J. A. Pople, Gaussian 03, Revision C.02 (Gaussian, Inc., Wallingford CT, 2004).

47 J. R. Barker, J. Phys. Chem. **91**, 3849 (1987).

48 F. Wang and D. P. Landau, Phys. Rev. Letters **86**, 2050 (2001).

49 M. Basire, P. Parneix, and F. Calvo, J. Chem. Phys. **129**, 081101 (2008).

50 I. M. Mills, in *Molecular Spectroscopy: Modern Research*, edited by K. N. Rao and C. W. Mathews (Academic Press, New York, 1972), Vol. 1, pp. 115.

51 D. G. Truhlar and A. D. Isaacson, J. Chem. Phys. **94** (1), 357 (1990).

52 G. Herzberg, *Molecular Spectra and Molecular Structure. II. Infrared and Raman Spectra*, 2 ed. (Van Nostrand Reinhold Co., Cincinnati, 1945).

53 F. Wang and D. P. Landau, Phys. Rev. E **64**, 056101 (2001).

54 E. Cané, A. Miani, and A. Trombetti, J. Phys. Chem. A. **111**, 8218 (2007).

55 M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. J. A. Montgomery, J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K.

Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, O. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, and D. J. Fox, Gaussian 09, Revision A.1 (Gaussian, Inc., Wallingford CT, 2009).

[56] J. F. Stanton, J. Gauss, M. E. Harding, P. G. Szalay, w. c. f. A. A. Auer, R. J. Bartlett, U. Benedikt, C. Berger, D. E. Bernholdt, Y. J. Bomble, O. Christiansen, M. Heckert, O. Heun, C. Huber, T.-C. Jagan, D. Jonsson, J. Jusélius, K. Klein, W. J. Lauderdale, D. A. Matthews, T. Metzroth, D. P. O'Neill, D. R. Price, E. Prochnow, K. Ruud, F. Schiffmann, S. Stopkowicz, J. Vázquez, F. Wang, and J. D. Watts, and the integral packages MOLECULE (J. Almlöf and P.R. Taylor); PROPS (P.R. Taylor); ABACUS (T. Helgaker, H.J. Aa. Jensen, P. Jørgensen, and J. Olsen); and ECP routines by A. V. Mitin and C. van Wüllen. CFOUR (2009).

[57] J. H. Miller, presented at the Twenty-Third Symposium (International) on Combustion, 1990 (unpublished).

[58] M. A. Harthcock and J. Laane, J. Mol. Spectros. **91**, 300 (1982).

[59] M. A. Harthcock and J. Laane, J. Phys. Chem. **89**, 4231 (1985).

[60] H.-J. Werner, P. J. Knowles, R. Lindh, F. R. Manby, M. Schütz, P. Celani, T. Korona, G. Rauhut, R. D. Amos, A. Bernhardsson, A. Berning, D. L. Cooper, M. J. O. Deegan, A. J. Dobbyn, F. Eckert, C. Hampel, G. Hetzer, A. W. Lloyd, S. J. McNicholas, W. W. Meyer, M. E. Mura, A. Nicklass, P. Palmieri, R. Pitzer, U. Schumann, H. Stoll, A. J. Stone, R. Tarroni, and T. Thorsteinsson, MOLPRO version 2006.1 (a package of *ab initio* programs) (see http://www.molpro.net., 2006).

[61] B. M. Wong, R. L. Thom, and R. W. Field, J. Phys. Chem. A. **110**, 7406 (2006).

[62] S. W. Benson, *Thermochemical Kinetics*, 2nd ed. (Wiley, New York, 1976).

[63] T. C. Allison, B. Ramachandran, J. Senekowitsch, D. G. Truhlar, and R. E. Wyatt, Theochem-Journal of Molecular Structure **454** (2-3), 307 (1998).

[64] C. H. Townes and A. L. Schlalow, *Microwave Spectroscopy*. (McGraw-Hill, New York, 1955).

[65] R. G. Gilbert and S. C. Smith, *Theory of Unimolecular and Recombination Reactions*. (Blackwell Scientific, Oxford, 1990).

[66] B. A. Ellingson, V. A. Lynch, S. L. Mielke, and D. G. Truhlar, J. Chem. Phys. **125** (8) (2006).

[67] O. V. Dorofeeva, V. S. Iorish, V. P. Novikov, and D. B. Neuman, J. Phys. Chem. Ref. Data **32** (2), 879 (2003).

[68] D. R. Lide, *CRC Handbook of Chemistry and Physics*, 81 ed. (CRC Press, Boca Raton, 2000).

[69] A. Meyer, J. Schroeder, and J. Troe, J. Phys. Chem. A **103**, 10528 (1999).

[70] R. Meyer, J. Chem. Phys. **52**, 2053 (1970).

[71] R. Meyer and H. H. Günthard, J. Chem. Phys. **49**, 1510 (1968).

[72] H. M. Pickett, J. Chem. Phys. **56**, 1715 (1972).

[73] B. Podolsky, Phys. Rev. E **32**, 812 (1928).

[74] C. C. Marston and G. G. Balint-Kurti, J. Chem. Phys. **91**, 3571 (1989).

[75] D. T. Colbert and W. H. Miller, J. Chem. Phys. **96**, 1982 (1992).

[76] M. J. Cohen, N. C. Handy, R. Hernandez, and W. H. Miller, Chem. Phys. Lett. **192** (4), 407 (1992).

[77] M. J. Cohen, A. Willetts, and N. C. Handy, J. Chem. Phys. **99** (8), 5885 (1993).

[78] J. R. Barker, N. F. Ortiz, J. M. Preses, L. L. Lohr, A. Maranzana, P. J. Stimac, T. L. Nguyen, and T. J. D. Kumar, MultiWell-2010.1 Software (Ann Arbor, Michigan, USA (http://aoss.engin.umich.edu/multiwell/), 2010).

[79] V. Barone, J. Chem. Phys. **122**, 014108 (2005).

[80] E. Gorin, Acta Physicochim., URSS **9**, 681 (1938).

[81] M. J. T. Jordan, S. C. Smith, and R. G. Gilbert, J. Phys. Chem. **95**, 8685 (1991).

[82] D. M. Wardlaw and R. A. Marcus, J. Phys. Chem. **90** (21), 5383 (1986).

[83] D. W. Wardlaw and R. A. Marcus, J. Chem. Phys. **110** (3), 230 (1984).

[84] J. A. Miller and S. J. Klippenstein, J. Phys. Chem. A **104** (10), 2061 (2000).

[85] S. J. Klippenstein, A. F. Wagner, S. H. Robertson, R. Dunbar, and D. M. Wardlaw, (Argonne National Laboratory, 1999).

[86] A. P. Penner and W. Forst, Chem. Phys. **13**, 51 (1976).

[87] S. C. Smith and R. G. Gilbert, Int. J. Chem. Kinet. **20**, 979 (1988).

[88] S. C. Smith and R. G. Gilbert, Int. J. Chem. Kinet. **20**, 307 (1988).

[89] J. Troe, J. Chem. Phys. **66**, 4745 (1977).

[90] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN. The Art of Scientific Computing*, 2 ed. (Cambridge University Press, Cambridge, 1992).

[91] S. C. Smith, M. J. McEwan, and R. G. Gilbert, J. Chem. Phys. **90**, 1630 (1989).

[92] S. J. Jeffrey, K. E. Gates, and S. C. Smith, J. Phys. Chem. **100** (17), 7090 (1996).

[93] S. H. Robertson, A. I. Shushin, and D. M. Wardlaw, J. Chem. Phys. **98** (11), 8673 (1993).

[94] P. K. Venkatesh, A. M. Dean, M. H. Cohen, and R. W. Carr, J. Chem. Phys. **107** (21), 8904 (1997).

[95] T. Baer and W. L. Hase, *Unimolecular Reaction Dynamics. Theory and Experiments*. (Oxford University Press, New York, 1996).

[96] M. J. Pilling and S. H. Robertson, Ann. Rev. Phys. Chem. **54**, 245 (2003).

[97] J. A. Miller, S. J. Klippenstein, and C. Raffy, J. Phys. Chem. A **106** (19), 4904 (2002).

[98] J. A. Miller and S. J. Klippenstein, J. Phys. Chem. A **110**, 10528 (2006).

[99] R. A. Marcus, J. Chem. Phys. **43** (8), 2658 (1965).

[100] E. V. Waage and B. S. Rabinovitch, Chem. Rev. **70**, 377 (1970).

[101] D. S. Perry, in *Highly Excited States: Relaxation, Reaction, and Structure*, edited by A. Mullin and G. C. Schatz (American Chemical Society, Washington DC, 1997), Vol. 678, pp. 70.

[102] E. E. Aubanel, D. M. Wardlaw, L. Zhu, and W. L. Hase, Int. Rev. Phys. Chem. **10** (3), 249 (1991).

[103] R. A. Marcus, J. Chem. Phys. **52**, 1018 (1970).

[104] E. B. Wilson, Jr., J. C. Decius, and P. C. Cross, *Molecular Vibrations. The Theory of Infrared and Raman Vibrational Spectra*. (McGraw-Hill Book Company, Inc., New York, 1955).

[105] L. Goodman, A. G. Ozkabak, and S. N. Thakur, J. Phys. Chem. **95**, 9044 (1991).

[106] D. T. Gillespie, Physica A: Statistical and Theoretical Physics (Amsterdam) **188**, 404 (1992).

[107] L. Vereecken, G. Huyberechts, and J. Peeters, J. Chem. Phys. **106** (16), 6564 (1997).

[108] D. E. Knuth, *Seminumerical Algorithms*, 2 ed. (Addison-Wesley, Reading, MA, 1981).

[109] W. H. Press and S. A. Teukolsky, ComPh **6** (5), 522 (1992).

[110] C. P. Tsokos, *Probability Distributions: An Introduction to Probability Theory with Applications*. (Wadsworth Publishing Company Inc., Belmont CA, 1972).

[111] M. Quack and J. Troe, Ber. Bunsen-Ges. Phys. Chem. **78** (3), 240 (1974).

[112] Y.-Y. Chuang, J. C. Corchado, P. L. Fast, J. Villà, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, M. Z. Gu, I. Rossi, E. L. Coitiño, S. Clayton, V. S. Melissas, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar, POLYRATE-version 8.1 (University of Minnesota,, Minneapolis, MN, 1999).

[113] W. Forst, J. Phys. Chem. **83** (1), 100 (1983).

[114] J. R. Barker, L. L. Lohr, R. M. Shroll, and S. Reading, J. Phys. Chem. A. **107** (38), 7434 (2003).

[115] I. Oref and B. S. Rabinovitch, Acc. Chem. Res. **12** (5), 166 (1979).

[116] K. Bolton and S. Nordholm, Chem. Phys. **207** (1), 63 (1996).

[117] K. Bolton and S. Nordholm, Chem. Phys. **206** (1-2), 103 (1996).

[118] K. Bolton and S. Nordholm, Chem. Phys. **203** (1), 101 (1996).

[119] S. Nordholm and A. Back, Phys. Chem. Chem. Phys. **3** (12), 2289 (2001).

[120] M. Gruebele and P. G. Wolynes, Acc. Chem. Res. **37** (4), 261 (2004).

[121] D. M. Leitner and P. G. Wolynes, ACH Models Chem. **134** (5), 658 (1997).

[122] D. M. Leitner and P. G. Wolynes, Phys. Rev. Lett. **76** (2), 216 (1996).

[123] J. R. Barker, N. F. Ortiz, J. M. Preses, L. L. Lohr, A. Maranzana, and P. J. Stimac, MultiWell-1.5.1 Software (Development version) (Ann Arbor, Michigan, USA, 2005).

[124] R. E. Weston, Jr. and J. R. Barker, J. Phys. Chem. A. **110**, 7888 (2006).

[125] C. Eckart, Phys. Rev. **35** (11), 1303 (1930).

[126] W. H. Miller, J. Am. Chem. Soc. **101** (23), 6810 (1979).

[127] M. Olzmann, Phys. Chem. Chem. Phys. **4**, 3614 (2002).

[128] N. W. Moriarity and M. Frenklach, presented at the Twenty Eighth Symposium (International) on Combustion, University of Edinburgh, Edinburgh, Scotland, 2000 (unpublished).

[129] R. A. Marcus, J. Chem. Phys. **45**, 2138 (1966).

[130] J. I. Steinfeld, J. S. Francisco, and W. L. Hase, *Chemical Kinetics and Dynamics*, 2 ed. (Prentice-Hall, 1998).

[131] A. Maranzana, J. R. Barker, and G. Tonachini, Phys. Chem. Chem. Phys. **9** (31), 4129 (2007).

[132] C. Wittig, I. Nadler, H. Reisler, M. Noble, J. Catanzarite, and G. Radhakrishnan, J. Chem. Phys. **83** (11), 5581 (1985).

[133] D. C. Tardy and B. S. Rabinovitch, Chem. Rev. **77**, 369 (1977).

[134] J. O. Hirschfelder, C. F. Curtiss, and R. B. Bird, *Molecular Theory of Gases and Liquids*. (Wiley, New york, 1964).

[135] G. Lendvay and G. C. Schatz, J. Phys. Chem. **96** (9), 3752 (1992).

[136] W. D. Lawrance and A. E. W. Knight, J. Chem. Phys. **79** (12), 6030 (1983).

[137] A. J. Stace and J. N. Murrell, J. Chem. Phys. **68** (7), 3028 (1978).

[138] L. M. Yoder and J. R. Barker, J. Phys. Chem. A **104** (45), 10184 (2000).

[139] B. Xue, J. Han, and H.-L. Dai, Phys. Rev. Lett. **84** (12), 2606 (2000).

[140] J. T. Yardley, *Introduction to Molecular Energy Transfer*. (Academic Press, New York, 1980).

[141] R. G. Gilbert and K. D. King, Chem. Phys. **49**, 367 (1980).

[142] I. Oref and D. C. Tardy, Chem. Rev. **90**, 1407 (1990).

[143] U. Hold, T. Lenzer, K. Luther, K. Reihs, and A. C. Symonds, J. Chem. Phys. **112** (9), 4076 (2000).

[144] J. F. Durana and J. D. McDonald, J. Chem. Phys. **64**, 2518 (1977).

[145] J. R. Barker, J. Chem. Phys. **72** (6), 3686 (1980).

[146] D. M. Golden, M. J. Rossi, A. C. Baldwin, and J. R. Barker, Acc. Chem. Res. **14**, 56 (1981).

[147] P. Urbain, B. Leyh, F. Remacle, and A. J. Lorquet, J. Chem. Phys. **110** (6), 2911 (1999).

[148] W. Tsang, V. Bedanov, and M. R. Zachariah, (1996).

[149] N. Davidson, *Statistical Mechanics*. (McGraw-Hill Book Company, Inc., New York, 1962).

[150] J. E. Mayer and M. G. Mayer, *Statistical Mechanics*. (John Wiley & Sons, Inc., New York, 1940).

[151] B. M. Toselli and J. R. Barker, J. Phys. Chem. **93**, 6578 (1989).

[152] B. M. Toselli and J. R. Barker, Chem. Phys. Lett. **159**, 499 (1989).

Index