May 8, 2009

MultiWell Program Suite User Manual

(MultiWell-2009.2)

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

University of Michigan
Ann Arbor, MI 48109-2143

J. R. Barker contact: jrbarker@umich.edu
(734) 763 6239

Preface

MultiWell Literature Citation

Please use the following citations to acknowledge results obtained using this version of the MultiWell Program Suite:

a) MultiWell-2009.2 Software, 2009, designed and maintained by John R. Barker with contributors Nicholas F. Ortiz, Jack M. Preses, Lawrence L. Lohr, Andrea Maranzana, Philip J. Stimac, and Lam T. Nguyen; University of Michigan, Ann Arbor, MI; http://aoss.engin.umich.edu/multiwell/.

b) John R. Barker, Int. J. Chem. Kinetics, 33, 232-45 (2001).

About the Authors

JOHN BARKER wrote most of the code, originally based on his 1983 paper on stochastic methods and subsequent developments.[1-4] 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 2009.0).

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
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.

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).[5,6] 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[7] of the Beyer-Swinehart algorithm.[8] 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) symmetrical and unsymmetrical hindered rotations (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. It also calculates canonical transition state theory rate constants, when provided with parameters for reactants and the transition state.

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, 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.)

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
          /densum                                  [source code for densum]
          /mominert                                [source code for mominert]
          /multiwell                               [source code for multiwell]
               /test                               [test files for multiwell]
          /thermo                                  [source code for thermo]
               /test                               [test files for thermo]
          /gauss2multi                             [source code for gauss2multi]
               /test                               [test files for gauss2multi]
```

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.

1.4 Starting Up Windows Version

In this version, binary executables (application, or .EXE files) have already been compiled and are found in directory **bin**. Programs in the MultiWell Suite are executed as 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 five codes: MultiWell, DenSum, Thermo, and Gauss2Multi. The Thermo examples include a small database of reactive atoms, free radicals, and other species. 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**, a code for calculating thermodynamic properties. Data for several dozen molecules are included in the file.

2. MultiWell Master Equation Code

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

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[9-11] 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,[12-14] as modified and implemented in our laboratory.[1-4] It has been described in considerable detail in a recent publication.[15] An example calculation has also been published.[16]

In the example,[16] 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.[16]

2.2 Terminology

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

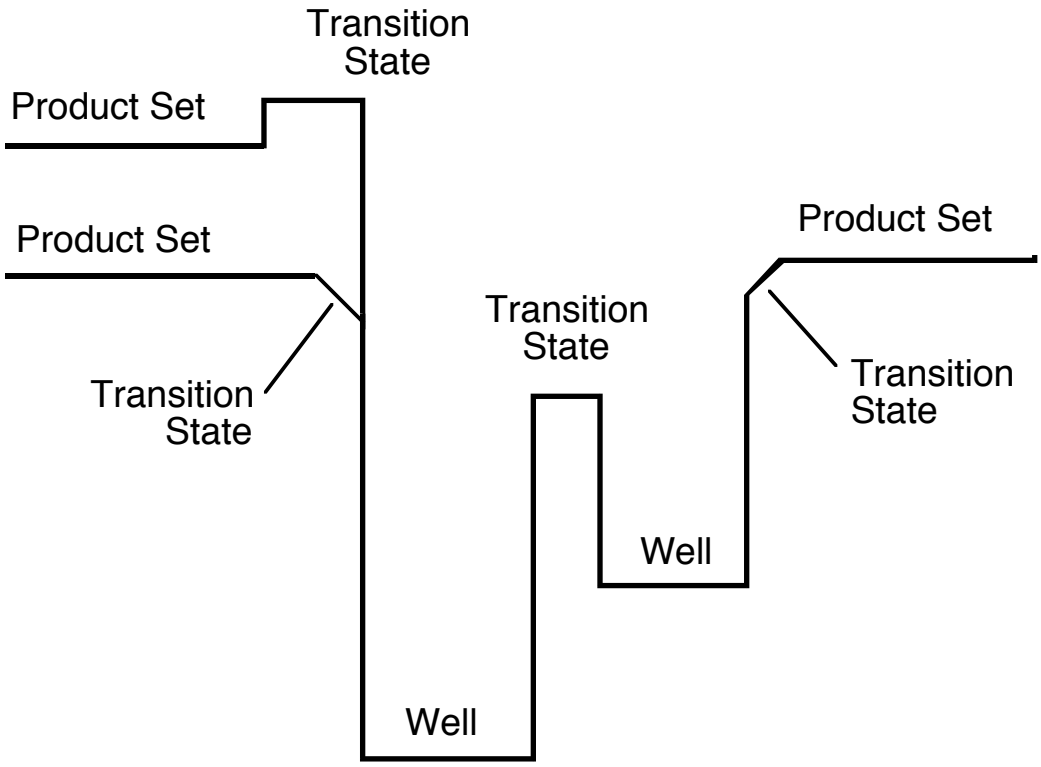
- "Wells" are chemical species corresponding to local minima on the potential energy surface.

- "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:

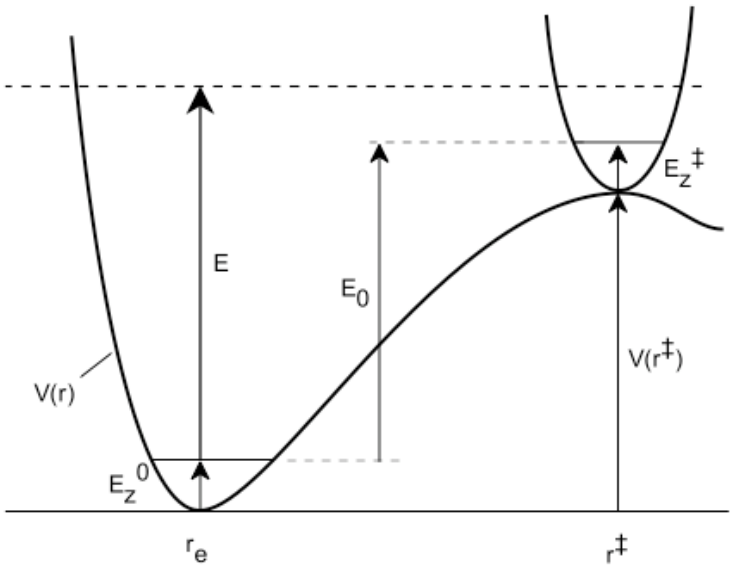


Figure 0. Energy schematic. The active energy E is measured from the zero point energy of the reactant well. The critical energy E_0 is the zero point energy difference between the transition state and the reactant well.

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. It 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" is a very convenient word processor for text files (share-ware available at http://www.nearside.com/trans-tex/), 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 file 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 words be entered in this exact order!*]

`Punits`	one of the following pressure units keywords: 'BAR', `'ATM'`, `'TOR'` [also will accept `'TORR'`], or `'MCC'`[for molecules/cc].
`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 contants 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: amu•Å^2)
`Molsym` external symmetry number for well (see Technical Note on Symmetry Numbers)
`Molele` electronic partition function for well (REAL number)
`Molopt` number of optical isomers for well

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: kcal/mole) [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[17]

(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 (amu Å^2)
j	external symmetry number for TS (see Technical Note on symmetry Numbers)
Qel	electronic partition function for TS (REAL number)
l	number of optical isomers for TS
AA	A-factor for reaction (units: s^{-1}); only used for ILT method, but ALWAYS read in
EE	reaction critical energy, relative to ZPE of reactant (Mol);
KEYWORDS	ALWAYS READ FIVE KEYWORDS, IN ANY ORDER. Select one from each of the Five Groups below. **See Section 2.8 (FATAL INPUT ERRORS) for a list of incompatible choices.**

Group 1

'NOREV'	for neglecting the reverse reaction
'REV'	for calculating reverse reaction rate

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

'CENT'	for quasi-diatomic centrifugal correction
'NOCENT'	for no centrifugal correction [**Note:** the calculated k_∞ should be numerically the same for both 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. ***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. 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 "Vibrational Energy Transfer Modeling of Non-Equilibrium Polyatomic Reaction Systems", John R. Barker, Laurie M. Yoder, and Keith D. King, J. Phys. Chem. A, 105, 796-809 (2001).

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

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)

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

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 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`

Execution is stopped and FATAL INPUT ERRORS are reported for the following incompatible combinations of keywords and input data:

`ILT` and **`CENT`**
`ILT` and **`TUN`**
`RKE` and **`TUN`**
`CENT` and **`TUN`**
`CENT` and **`TSmom(Mol,nchann)`** ≤ 0.0 (moment of inertia of the transition state)

3. DenSum: Sums and Densities of States

This computer code carries out exact counts for sums and densities of states via the Stein-Rabinovitch[7] extension of the Beyer-Swinehart algorithm.[8] Optionally, the Whitten-Rabinovitch approximation[18,19] 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)

Note the sign convention for anharmonicity.

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

where 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 (not including anharmonicity).

Classical Rotation (rot)

DenSum uses the method of Astholz et al.[20] For expressions, see Appendix 5 of Robinson and Holbrook,[21] or Section 4.5 of Holbrook et al.[22] Also see Appendix (Theoretical Basis), Section A.4.2 Note that starting with version 2.04, classical rotations are treated

purely classically; previously, a semi-classical approximation was imposed (that amounted to requiring that the sum of states equaled unity at E = 0).

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[23]:

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

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

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.

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

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

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. 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, $I_K \approx I_A$, where I_A is the unique moment of inertia. 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 $K \le |J|$, as in the usual approximation employed for the K-rotor, $E_r(J,K)$ can be <0. This is not serious a approximation for prolate tops, however, 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. For *almost* symmetric tops (where $I_B \approx I_C$), one can use either of two reasonable approximations for I_{2D}:

$$I_{2D} \approx \left[I_B I_C \right]^{1/2} \text{ [Ref. 9,11]} \qquad \text{or} \qquad I_{2D} \approx \left[I_B^{-1} + I_C^{-1} \right]^{-1} \text{ [Ref. 24]}$$

According to the correct treatment of the K-rotor, the 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. The ***kro*** degree of freedom type employs the correct treatment for a user-specified value of J.

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:

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

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

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

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

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

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})

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

Type **Ihrd1** (all coefficients in units of $amu.\text{Å}^2$)

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

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.,[20] 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

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

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:

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

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`	KEY WORD `'HAR'`: for vibrational frequencies input as harmonic frequencies. `'OBS'`: for vibrational frequencies input as 0-1 fundamental frequencies.
`VROT`	KEY WORD for molecular (internal and external) rotations `'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 (amu $Å^2$)
ANH = rotation symmetry number
NG = rotation dimension
'qro' (quantized rotation)
WE = rotation moment of inertia (amu $Å^2$)
ANH = rotation symmetry number
NG = rotation dimension
'kro' (K-rotor (1-dimensional); quantized rotation)
WE = rotation moment of inertia (amu $Å^2$)
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 (amu $Å^2$)

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 (amu $Å^2$)

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)

WE = total number of coefficients for potential energy function.

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

NG = dummy variable

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, SYMMV, 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

'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 are given in the examples directory: `multiwell/examples/mominert-examples`.

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

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

Line 3
(Repeat for every atom type [see Section 7.7])

`ATYPE, IA, X, Y, Z`

`ATYPE`	case sensitive atomic symbol, e.g. "C", "H", "Br79" (see Sec. 7.7)
`IA`	Index number of the atom (1 to `NATOMS`).
`X,Y,Z`	Cartesian coordinates (Å) of the atom.

Line 4

`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 5** and **Line 6** can be omitted.

Line 5

`NR`

Number of atoms in one of the two rotating moieties.

Line 6
List of the atom indices that comprise the rotating moiety containing `NR` atoms.
Line 7

For additional internal rotors, REPEAT lines 4-6. TO TERMINATE, enter two zeros: "0 , 0".

Line 8

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.[25]

5. Thermo: Thermochemistry

THERMO calculates **equilibrium constants**, **thermodynamic parameters**, and **canonical transition state theory rate constants** *via* standard statistical mechanics formulae.

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

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 cehmical species in this database are of interest to our group. We add entries from time to time, depending on our current reseaarch interests. The NIST WebBook[26] is a good source for input data, as are the NIST-JANAF Thermochemical Tables.[27]

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.[27] 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):

Note the sign convention for anharmonicity.

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

where 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 (not including anharmonicity).

Classical Rotation (rot):

DenSum uses the method of Astholz et al.[20] For expressions, see Robinson and Holbrook, Appendix 5.[21]

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]:

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

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:

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

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

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

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

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

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

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})

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

Type **`Ihrd1`** (all coefficients in units of *amu•Å*^{2})

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

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.[28-35] 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 ω (`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)

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

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

`VARSHNI` (Varshni Oscillator[36])

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

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

`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.

5.4 Data File Format

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

Line 1

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)

Line 2

Nt number of temperatures

Line 3

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

Line 4

Ns number of chemical species

Line 5

REPROD, MOLNAME, DelH

REPROD	key word: "reac" = reactant "prod" = product "none" = not included in equilibrium constant "ctst" = transition state. 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 chemical species (up to 10 characters)
DelH	enthalpy of formation at 0 K

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 7.7. 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 KEY WORD
'HAR': for vibrational frequencies input as harmonic frequencies.
'OBS': for vibrational frequencies input as 0-1 observed frequencies.

VROT KEY WORD for molecular (internal and external) rotations
'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.

Line13

(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 (amu $Å^2$)
- ANH = rotation symmetry number
- NG = rotation dimension

'qro' (quantized rotation)
- WE = rotation moment of inertia (amu $Å^2$)
- ANH = rotation symmetry number
- NG = rotation dimension

'kro' (K-rotor (1-dimensional); quantized rotation)
- WE = rotation moment of inertia (amu $Å^2$)
- 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 (amu $Å^2$)
- 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 (amu Å^2)
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)
WE = total number of coefficients for potential energy function.
ANH = total number of coefficients for rotational constant or moment of inertia function.
NG = dummy variable
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 9**)
WE = rotational moment of inertia (amu Å^2) for the internal rotor
ANH = rotational symmetry number
NG = rotational dimension

'gor' (Gorin Model. **Requires LINES 10-14**)
WE = 2D adiabatic external rotor (amu Å^2); not used but always read.
ANH = rotational symmetry number
NG = 2; rotational dimension

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

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`	Equilibrium frequency (cm^{-1}) of the dissociating (or newly-formed) bond.
Line 17	`De`	Dissociation Energy De (units of `Eunit`)
Line 18	`re`	Equilibrium center-of-mass distance (Å) between the of the two fragments that will dissociate
Line 19	`c`	Stiff-Morse parameter (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[37] and GAUSSIAN 03.[38]

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. Technical Notes

7.1 Conversion Factors

Moment of Inertia (Im):

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

Rotational Constant

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

7.2 Vibrational Anharmonicities

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

E_{vib} = `WE`*(v+1/2) + `ANH`*(v+1/2)*(v+1/2), v=0,1,2,3,...

For a Morse oscillator, `ANH` < 0.

7.3 Vibrational Degeneracies in DenSum

In DenSum, vibrational degeneracies are treated as accidental. Identical results are obtained if each vibrational degree of freedom with degeneracy `NG` is entered in the DenSum 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]

7.4 Symmetry Numbers

For use in DenSum, MultiWell, and Thermo, we recommend the following convention for symmetry numbers (described by Gilbert and Smith[39]):

The EXTERNAL SYMMETRY NUMBER does 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), as discussed below.

All polyatomics are treated by DenSum and MultiWell as SYMMETRIC TOPS with moments of inertia I_x, $I_y \approx I_z$ and rotational constants $A = \hbar^2/2I_x$ and $B == \hbar^2/2I_y$. The rotational energy of a symmetric top is given by

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

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). For a given value of J, quantum number K can take values from $-J$ to $+J$, inclusive; all are doubly degenerate except for K=0, which is singly degenerate. *In the present version of DenSum, the range of K is* ***not*** *restricted to ±J and the quantity (A-B) is assumed to be >1 (prolate symmetric top, which corresponds to a dissociating diatomic molecule).*

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**).

K-ROTOR properties are is listed in **densum.dat** and included in density and sums of states calculations. Its moment of inertia is $I_K = 2(A-B)/\hbar^2 = 2I_AI_B/(I_B-I_A)\hbar^2$, which is sometimes approximated as $I_K \approx I_A$.

2-D ADIABATIC ROTOR moment of inertia is listed in **multiwell.dat** on Line 8 (for wells) or Line 14 (for transition states). Its moment of inertia is $I_{2D} = I_B = 2B/\hbar^2$. For *almost* symmetric tops (where $I_B \approx I_C$), one can use either of two reasonable approximations for I_{2D}:

$$I_{2D} \approx \left[I_B I_C\right]^{1/2} \text{ [Ref. 9,11]} \qquad \text{or} \qquad I_{2D} \approx \left[I_B^{-1} + I_C^{-1}\right]^{-1} \text{ [Ref. 24]}$$

According to this convention, the EXTERNAL SYMMETRY NUMBER for Ethane is 3x2=6, and the internal rotor symmetry number is 3. The symmetry numbers for the external K-rotor and adiabatic 2-D-rotor are set equal to unity in DenSum and Thermo.

7.5 Internal Rotor Symmetries

Internal rotors are characterized by `NG`, the symmetry of the potential energy (i.e. the foldedness, or number of minima per 2π rotation), and `NSIG`, the symmetry number of the internal rotation, which depends on the substituents. For MultiWell calculations, `NG` and `NSIG` are given in the DenSum data file and the external symmetry number is given in the MultiWell data file.

Examples:

Data file:	**densum.dat**		**multiwell.dat**
	`NG`	**`NSIG`**	**Ext.Symm**
CH_3-CD_3	3	3	3
CH_3-CD_2H	3	3	1
CH_3-CDH_2	3	3	1
CH_3-CH_3	3	3	6
CH_2D-CH_2D	3	1	2
CH_2D-CD_2H	3	1	1
CD_3-CD_3	3	3	6

7.6 A Handy List of Lennard-Jones Parameters

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

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

Species	σ / Å	(ε/k$_B$) / K	Reference
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
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 (ethylene oxide)	4.08	421	1

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).

7.7 Recognized Elements and Isotopes

The following atomic symbols (atom types) are recognized. 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.[27] Isotopic masses were taken from N. E. Holden, "Table of the Isotopes (Revised 1998)".[40]

Atomic Symbol	Atomic Mass (g/mol)
Ar	39.948
B	10.81
Br	79.904
Br79	78.918338
Br81	80.916291
C	12.011
C12	12.0
C13	13.003355
C14	14.003242
C16	16.0147
Cl	35.453
Cl35	34.96885271
Cl37	36.9659
F	18.998403
H	1.0079
H1	1.007825
D	2.014102
T	3.016049
He	4.00260
Hg	200.59
I	126.9045
K	39.0983
Kr	83.80
Li	6.941
O	15.9994
N	14.00674
N14	14.003074
N15	15.00010897
Na	22.98977
Ne	20.179
O16	15.994915
O17	16.9991315
O18	17.999160
P	30.97376
Rb	85.4678
S	32.06
S32	31.9720707
S34	33.9678668
Se	78.96
Si	28.0855
Si29	28.98
Si30	29.97
Sn	118.69
Te	127.60
Xe	131.29

8. 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: I can only find output for rates at infinite pressure. Does the code calculate it for finite pressure?

ANSWER: The code is not designed to calculate conventional unimolecular reaction rate constants. One of the outputs gives k(infinity), but to find k(uni) in the fall-off, 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 their steady-state values. MultiWell is not designed for steady-state calculations, but the relative populations will approach steady-state values if the reaction rates are slow, compared with energy relaxation rates. Here are the two methods:

a) the 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, a large number of stochastic trials are needed in a given 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 as a function of time.

5. 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.

6. 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.

7. QUESTION: For chemical activation systems, how can one tell whether a simulation has been run for a long enough simulated time?

ANSWER: Pragmatically, one should run the simulations long enough to determine if two criteria are satisfied:

a) the average energy of 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 essentially 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.

8. QUESTION: How can one determine chemical activation rate constants from MultiWell outputs?

ANSWER: 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. To use MultiWell for

calculating a chemical activation rate constant, one must often rely on separation of time scales. The same thing is necessary in nature. In the case of the A+B=C reaction, there are usually additional channels that slow. Pragmatically, this means that if one uses MultiWell to simulate the A+B=C reaction over some time interval and the yield of C asymptotically approaches a constant value, then that asymptotic yield can be used with k(infinity) to calculate a rate constant for chemical activation [see Eq. (31) in reference [15]]. If one runs a much longer simulation, one will often find that product C hasn't really become constant, but is decaying with a longer time constant. If this decay is slow enough, it can be neglected when assessing the much faster chemical activation process. This is an example of reliance on separation of time scales. This is typically a good method as long as the time scales are different by an order of magnitude or more. If they are separated by less than that, it may not be possible to define a simple "rate constant" for chemical activation.

9. 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).

10. 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[35] (see Benson's discussion[41] of the Hindered Gorin Model), which is based on a modified version of the Gorin Model.[42] 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[35] 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[43] 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[44,45] 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[46] have used VariFlex software[47] to implement an approximate version of the Marcus-Wardlaw approach.

Appendix. Theoretical Basis

This Appendix is an expanded and corrected version of the paper originally published in the International Journal of Chemical Kinetics (Wiley):

Barker, John R. (2001), Multiple-well, multiple-reaction-path unimolecular reaction systems. I. MultiWell computer program suite, Int. J. Chem. Kinetics, 33, 232-245.

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[48-57]). The master equation provides the fundamental theoretical basis for modeling systems in which both energy transfer and chemical reaction can occur.[1,9,11,21,39] 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[9,11] 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.[9,11,21,39,58] (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.)

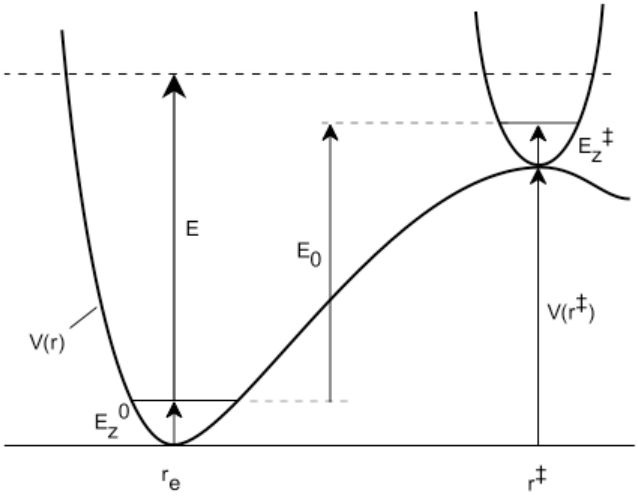


Figure 0. Energy schematic. The active energy E is measured from the zero point energy of the reactant well. The critical energy E_0 is the zero point energy difference between the transition state and the reactant well.

A.2.2 Sums and Densities of States

The MultiWell suite of computer codes includes DenSum, which utilizes the Stein-Rabinovitch[7] version of the Beyer-Swinehart algorithm[8] 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.,[20] 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

(This section contributed by Lam T. Nguyen.)

In the MultiWell Suite of programs, torsional modes of a chemical species 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 Schrodinger equation for such 1D-torsional motions can be written as follows[59]:

$$\left[-\frac{\partial}{\partial\chi}B_{hr}(\chi)\frac{\partial}{\partial\chi}+V(\chi)\right]\psi(\chi)=E\psi(\chi) \qquad (2.2\text{-}1)$$

where χ is the torsional (dihedral) angle ($0\le\chi<2\pi$), E and Ψ are energy eigenvalue and eigenvector, respectively, $V(\chi)$ is the torsional potential energy function, and B_{hr} is the rotational

constant, which can be a real constant (rigid rotor) or a function of torsional angle (non-rigid rotor). When B_{hr} is assumed to be a constant, which is realistic only for symmetrical rotors, equation (1) simplifies:

$$\left[-B_{hr}\frac{\partial^2}{\partial\chi^2}+V(\chi)\right]\psi(\chi)=E\psi(\chi) \tag{2.2-2}$$

Solutions of equation (2.2-2) are well-known,[59-61] but are not given here. for the purpose of calculating densities of states and computing partition functions in the MultiWell Suite, Eq. 1 is solved 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.[59] The Hamiltonian matrix of Eq. 2.2-1 is given by:

$$\mathbf{H}=-\mathbf{D}B_{hr}\mathbf{D}+\mathbf{V}=\mathbf{D}^T B_{hr}\mathbf{D}+\mathbf{V} \tag{2.2-3}$$

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** matrix (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 of the rotational constant. For 2N +1 grid points, elements of **D** are expressed:

$$D(i,j)=(-1)^{i-j}\left\{2\sin\left[(i-j)\pi/(2N-1)\right]\right\}^{-1} \quad \text{for } i\neq j \text{ and } D(i,i)=0. \tag{2.2-4}$$

To construct the symmetrical matrix **H**, one requires both $V(\chi)$ and $B_{hr}(\chi)$. Several types of common functions for $V(\chi)$ and $B(\chi)$ (or moment of inertia function $I(\chi)$) can be understood by MultiWell, as explained elsewhere in this manual. The H matrix is diagonalized in order to obtain a vector of energy eigenvalues, which are used to compute 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 using any of the many available quantum chemistry codes. The results for $V(\chi)$ can be fitted to a suitable truncated Fourier series, and one may use codes like the I_Eckart program[62] (written for use with MatLab) to compute $B_{hr}(\chi)$ or $I_{hr}(\chi)$.

A.2.3 Master Equation for the Vibrational Quasi-Continuum

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}.[63] 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.[63] In the vibrational quasicontinuum, individual quantum states cannot be resolved and the master equation can be written:

$$\frac{dy(E',t)}{dt} = f(E',t)\,dE' + \int_0^\infty dE\left[R(E',E)\; y(E,t)\right] - \int_0^\infty dE\left[R(E,E')\; y(E',t)\right] - \sum_{i=1}^{channels} k_i(E')\,y(E',t) \tag{2.3-1}$$

where $y(E',t)dE'$ is the concentration of species with vibrational energy in the range E' to $E'+dE'$, $R(E,E')$ is the (pseudo-first-order) rate coefficient for VET from energy E' to energy E, $f(E',t)dE'$ is a source term (e.g. chemical or photo activation), and $k_i(E')$ is a unimolecular reaction rate constant for molecules at energy E' reacting *via* the i^{th} channel. Terms involving radiative emission and absorption have been omitted.

In MultiWell, initial distributions are posited and the master equation is integrated to obtain time-dependent population distributions, reaction yields, etc. Initial distributions appropriate for several common phenomena are discussed below. In the current version of MultiWell, the source term f(E',t)dE' on the right hand side of Equation (2.3-1) is not included.

If the rate coefficients R(E,E') do not depend on the initial quantum states of the collider bath molecules, they can be written 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:

$$R(E,E')\,dE = \int_0^\infty R(E,E')\,dE \left\{ \frac{R(E,E')\,dE}{\int_0^\infty R(E,E')\,dE} \right\}, \tag{2.3-2a}$$

$$= \omega\, P(E,E')\,dE \tag{2.3-2b}$$

The first factor on the right hand side of Equation (2a), the integral over the rates of all inelastic transitions from initial energy E', is the frequency of inelastic collisions, ω. Usually, the collision frequency is calculated from the expression $\omega = k_c[M]$, where k_c is the bimolecular rate constant for inelastic collisions (which in general may depend on E) and $[M]$ is bath gas concentration. The second factor (in curly brackets) on the right hand side of Equation (2.3-2a) is $P(E,E')dE$. It is important to emphasize that the factorization of $R(E,E')$ in Equation (2.3-2) 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:

$$\int_0^\infty P(E,E')\,dE = 1. \tag{2.3-3}$$

Note that collision step-size distributions for activating and deactivating collisions are connected *via* detailed balance:

$$\frac{P(E,E')}{P(E',E)} = \frac{\rho(E)}{\rho(E')} \exp\left\{-\frac{E-E'}{k_B T_{trans}}\right\}, \qquad (2.3\text{-}4)$$

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.

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.[64] 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[9,11,21,22,39] 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 $\rho(E)$ 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 $\rho(E)$ 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 ($\rho(E)$, 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 ($\rho(E) = 0$) 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[65] + 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 [16].

Typical convergence tests[16] show that 250 bins with ΔE_{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[39] 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.

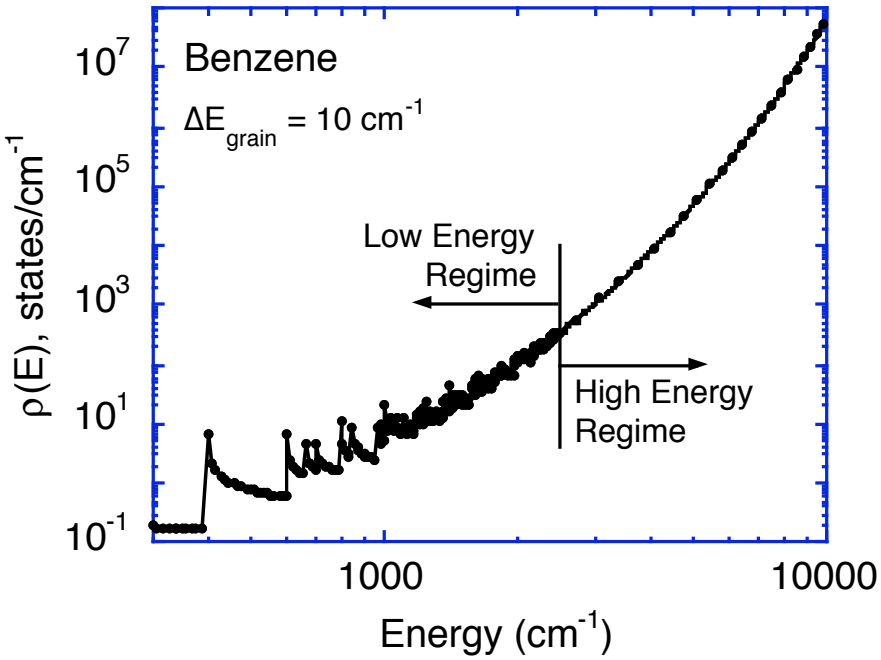


Figure 1. Density of states for benzene (including vibrations and one active external rotation). Solid line: density of states from exact count (ΔE_{grain} = 10 cm^{-1}); solid dots: elements of double array (see text for details).

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.[13,14] The algorithm has been described in the context of chemical kinetics.[1,12,66] 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.[12] 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.[57,67] 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:

$$\sum_{j=1}^{n-1} A_j < r_2 A_T \leq \sum_{j=n}^{k} A_j \qquad (3.1\text{-}2)$$

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,[53,68,69] 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 ≈ 2.1×10^9 equally-spaced numbers. Thus if the ratio of minimum to maximum values of the rate constants is less than ~0.5×10^{-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:

$$1 = f_1 + f_2 + \ldots = \sum_{i=1}^{species} f_i \qquad (3.2\text{-}1)$$

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[70]:

$$\sigma_i = \sqrt{\frac{1}{N_{trials}} f_i(1-f_i)} \tag{3.2-2}$$

where f_i is the fractional population of the i^{th} 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. σ_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[10,11,21,22,39]:

$$k(E) = \left\lfloor \frac{m^\ddagger}{m} \frac{\sigma_{ext}}{\sigma_{ext}^\ddagger} \right\rfloor \frac{g_e^\ddagger}{g_e} \frac{1}{h} \frac{G^\ddagger(E-E_0)}{\rho(E)} \tag{4.1-1}$$

where $m^\ddagger$ and m are the number of optical isomers,[39] $\sigma_{ext}^\ddagger$ 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 $\rho(E)$ 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.[39] 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*.[39]

At low energies, where densities of states may be very sparse, $\rho(E)$ 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.

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.[9,11,39] 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

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

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 $K \le |J|$, 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 is 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}:

$$I_{2D} \approx \left[I_B I_C \right]^{1/2} \text{ [Ref. 9,11]} \qquad \text{or} \qquad I_{2D} \approx \left[I_B{}^{-1} + I_C{}^{-1} \right]^{-1} \text{ [Ref. 24]}$$

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** (Keyword `'CENT'`) 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[21] (or Section 3.10 of Holbrook et al.[22]). Essentially, *k(E,J)* can be written

$$k(E,J) = \left[\frac{m^{\ddagger}}{m} \frac{\sigma_{ext}}{\sigma_{ext}^{\ddagger}} \right] \frac{g_e^{\ddagger}}{g_e} \frac{1}{h} \frac{G^{\ddagger}(E^+)}{\rho(E^+ + E_0 + \Delta E_r(J))} \tag{4.1-1b}$$

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:

$$\Delta E_{rot}(J) = \left\{ B_{\ddagger} - B_e \right\} J(J+1) \tag{4.1-1c}$$

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:

$$k(E) = \frac{1}{Q_{2D}} \sum_{J=0}^{\infty} k(E,J) \cdot (2J+1) \cdot \exp\left[\frac{-B_e \cdot J \cdot (J+1)}{k_B T} \right] \tag{4.1-1d}$$

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 somewhat overcome if a semi-empirical approach like the Hindered Gorin Model is applied, where $B_{\ddagger}$ is assumed to vary with temperature.

For loose transition states, more elaborate techniques are needed for calculating *k(E)* accurately. Such techniques include Variational Transition State Theory,[39,58] Adiabatic Channel Model,[71] and Flexible Transition State Theory.[45] Computer codes have been published for some of these theories.[47,72] These methods can be used in the current version of MultiWell by calculating $k(E-E_0)$ externally and then reading in the values in a data file. When the rate constant is known from experiments, however, it is often convenient to use a "restricted" Gorin Model with a "hindrance parameter" selected to reproduce the known rate constant for the corresponding reverse (recombination) reaction.[35,39,41]

According to the Gorin model,[42] 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,[39] 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.[39]

In the restricted, or Hindered Gorin Model,[35,39,41] 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,[35] 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.[31,73]

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.

Another selectable option in MultiWell is to calculate k(E) using the Inverse Laplace Transform method described by Forst:[9,11,74]

$$k(E)=\left\lceil \frac{m^{\ddagger}}{m}\frac{\sigma_{ext}}{\sigma_{ext}^{\ddagger}} \right\rceil \frac{g_e^{\ddagger}}{g_e} A_{\infty} \frac{\rho(E-E_{\infty})}{\rho(E)} \tag{4.1-2}$$

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.

When *k(E)* is calculated according to the Inverse Laplace Transform (ILT) Method, the centrifugal corrections correspond to Equations (4.30) and (4.31), respectively, in Robinson and Holbrook[21], or Equations (3.30) and (3.31) in Holbrook et al.[22] Essentially, this amounts to adopting a temperature-corrected threshold energy E_0^T:

$$E_0^T = E_0 - k_B T_{trans} \left\{ \frac{I_{2D}^{\ddagger}}{I_{2D}} - 1 \right\} \tag{4.1-3}$$

where I_{2D} and $I_{2D}^{\ddagger}$ are the moments of inertia for the external two-dimensional adiabatic (inactive) rotations of the reactant and of the transition state, respectively. The threshold energy including the centrifugal correction for each reaction is listed in output file multiwell.array above the column of k(E) values for each reaction: "Eo+Rot: ...".

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):

$$k_{\infty}(T_{trans}) = \frac{1}{Q(T_{trans})} \int_{E_0}^{\infty} k(E)\,\rho(E) \exp(-E/k_B T_{trans})\,dE \tag{4.1-4a}$$

$$k_0^{SC}(T_{trans}) = \frac{k_{coll}}{Q(T_{trans})} \int_{E_0}^{\infty} \rho(E) \exp(-E/k_B T_{trans})\,dE \tag{4.1-4b}$$

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}:

$$Q(T_{trans}) = \int_0^\infty \rho(E)\exp(-E/k_B T_{trans})dE \tag{4.1-5}$$

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}.[16] The activation energy is obtained by calculating $k_\infty(T_{trans})$ at two closely-spaced temperatures:

$$E_\infty = -R\frac{\ln\left[k_\infty(T_2)/k_\infty(T_1)\right]}{\left[T_2^{-1} - T_1^{-1}\right]} \tag{4.1-6}$$

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 `'CENT'` 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,[22,75-79] but perhaps the most successful of these is the local random matrix model of Wolynes and coworkers.[5,6,80-82]

According to most of the IVR models,[5,79] the effects of slow IVR can be accounted for with an IVR transmission coefficient:

$$\kappa_{IVR}(U,[M]) = \frac{k^q_{IVR}(U) + k^c_{IVR}[M]}{k^q_{IVR}(U) + k^c_{IVR}[M] + \nu_{ivr}} \tag{4.1-7}$$

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[5] has been used with MultiWell.[83,84] 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[5] 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:

$$k\left(E,[M]\right) = \kappa_{IVR}\left(E,[M]\right) \cdot k_{RRKM}\left(E\right) \tag{4.1-9}$$

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.[5,84] 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[17] (see the next section).

Tunneling

(This section 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.[85] 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[86]:

$$N(E) = \int_{-V_o}^{E-V_o} P(E_1) N^{'}(E - E_1) dE_1 \tag{4.1-10}$$

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:

$$P(E_1) = \frac{\sinh(a)\sinh(b)}{\sinh^2((a+b)/2)\cosh^2(c)} \tag{4.1-11a}$$

$$a = \frac{4\pi}{h\nu_i} \frac{\sqrt{E_1 + V_o}}{(V_o^{-0.5} + V_1^{-0.5})} \tag{4.1-11b}$$

$$b = \frac{4\pi}{h\nu_i} \frac{\sqrt{E_1 + V_1}}{(V_o^{-0.5} + V_1^{-0.5})} \tag{4.1-11c}$$

$$c = 2\pi\sqrt{\frac{V_o V_1}{(h\nu_i)^2} - \frac{1}{16}}. \qquad (4.1\text{-}11d)$$

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

$$N(E) = \int_0^E P(E - V_o - E^\dagger) N'(E^\dagger) dE^\dagger . \qquad (4.1\text{-}12)$$

$E^\dagger$ 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.[86]

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.[87] 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[88] 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^J can be calculated using the canonical bimolecular rate constant: $k^J = k_{bim}[B]$. This value for k^J 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.[89,90]

Theory

The discussion in this section was adapted from our recent paper on reactions of acetyl radical with O_2.[91]

Bimolecular reactions can be treated by using the bimolecular pseudo-first-order microcanonical approach. The microcanonical bimolecular rate constant is given by

$$k_{bim}(E^+) = \left[\frac{m^{\neq}_{AB}\ \sigma_A \sigma_B}{m_A m_B\ \sigma^{\neq}_{AB}} \right] \frac{g^{\neq}_{e(AB)}}{g_{e(A)} g_{e(B)}} \quad \frac{1}{h} \frac{G^{\neq}_{AB}(E^+ - E_0)}{\rho_{AB}(E^+)} \tag{4.2-1}$$

$$\rho_{AB}\left(E^+\right) = \int_0^{E^+} \rho_A\left(E^+ - x\right) \rho_B\left(x\right)\, dx \tag{4.2-2}$$

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 $G^{\ddagger}_{AB}(E^+\text{-}E_0)$ 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 $G^{\ddagger}_{AB}(E^+\text{-}E_0)$. The canonical bimolecular rate constant can be calculated by averaging over the canonical energy distribution of the A+B supermolecule:

$$\left\langle k_{bim}\left(T\right)\right\rangle = \frac{\int_{E_0}^{\infty} k_{bim}(E^+)\ \rho_{AB}(E^+)\ e^{-E^+/RT} dE^+}{Q_{tot}(T)} \tag{4.2-3}$$

where $Q_{tot}(T)$ is the total partition function of the supermolecule at temperature T:

$$Q_{tot}\left(T\right) = \int_0^{\infty} \rho_{AB}(E^+)\ e^{-E^+/RT} dE^+ \tag{4.2-4}$$

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:

$$\left\langle k_{bim}(E_1,T)\right\rangle_2 = \frac{\int_{E_{low}}^{\infty} k_{bim}(E_1+E_2)\ \rho_2(E_2)\ \exp\left(\frac{-E_2}{kT}\right)dE_2}{Q_2(T)} \tag{4.2-5}$$

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:

$$Q_2(T) = \int_0^{\infty} \rho_2(E_2)\ \exp\left(\frac{-E_2}{kT}\right) dE_2 \tag{4.2-6}$$

The canonical bimolecular rate constant can be obtained from $<k_{bim}(E_v,T)>_2$ by averaging over E_1:

$$\left\langle k_{bim}(T)\right\rangle = \frac{\int_0^{\infty}\left\langle k_{bim}(E_1,T)\right\rangle_2\ \rho_1(E_1)\ \exp\left(\frac{-E_1}{kT}\right)dE_1}{Q_1(T)} \tag{4.2-7}$$

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:

$$Q_1(T) = \int_0^{\infty} \rho_1(E_1)\ \exp\left(\frac{-E_1}{kT}\right)dE_1 \tag{4.2-8}$$

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}.[91]

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:

$$k^I(E,T) = \left\langle k_{bim}(E_1,T)\right\rangle_2\ [B] \tag{4.2-9}$$

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.[91]

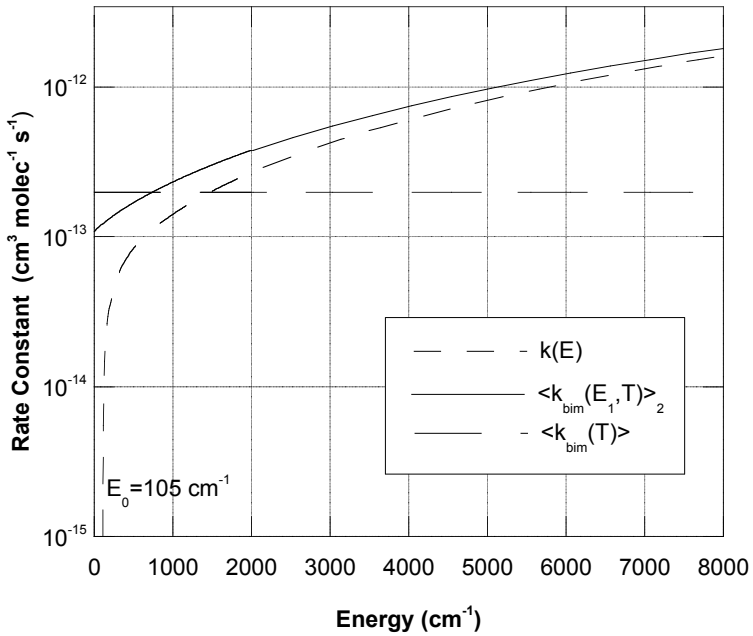


FIGURE 4.2: A comparison of k(E), $<k_{bim}(E_1,T)>_2$, and $<k_{bim}(T)>$ at T=298 K for the reaction **A2'** + O_2 → **B1** [see reference [91] for details].

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[15] have been coded only for unimolecular reactions. To include the reverse reaction would require using a theory for partitioning energy between unimolecular reaction products[58,92] 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:[9,11] $Q(\beta) = \mathfrak{L}[\rho(E)]$, where β is the Laplace transform parameter and can be identified as $\beta = (k_B T)^{-1}$. 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:[11]

$$G_{12}(E) = \int_0^E G_1(x)\rho_2(E-x)dx \tag{4.2-10}$$

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:[11]

$$G_{12}(E) = \mathfrak{L}^{-1}\left\{\frac{Q_1(\beta)Q_2(\beta)}{\beta}\right\} \tag{4.2-11}$$

where $Q_i(\beta)$ is the partition function of the i^{th} group of modes. For p independent classical rotations, the partition function can be written[11,22]

$$Q_{rot}(T) = \prod_{i=1}^{p}\left\{\left(\frac{8\pi^2 I_i k_B T}{h^2}\right)^{d_i/2}\Gamma\left(\frac{d_i}{2}\right)\right\} = a_r\left(k_B T\right)^{r/2} \tag{4.2-12a}$$

$$r = \sum_{i=1}^{p} d_i \tag{4.2-12b}$$

where d_i is the dimensionality of the i^{th} 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[11]

$$Q_{trans}(T) = \left\{\frac{2\pi\mu}{h^2}\right\}^{3/2} V\left(k_B T\right)^{3/2} = a_t\left(k_B T\right)^{3/2} \tag{4.2-13}$$

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:[11,90]

$$G_t(E) = \frac{4\pi(2\mu E)^{3/2}}{3h^3} V \qquad (4.2\text{-}15)$$

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.[20] modification of the Stein-Rabinovitch state-counting method.[7]

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:[93]

$$k_c = \pi\sigma^2 \langle v \rangle \Omega^{(2,2)*} \qquad (4.3\text{-}1)$$

where <v> is the average speed at the translational temperature, σ is the Lennard-Jones diameter, and $\Omega^{(2,2)*}$ is the collision integral,[94] which depends on the Lennard-Jones parameters. Since only the product $k_c P(E,E')$ 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.[95] 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[96] 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,[97,98] 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.[99] 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[100] 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.[17]

In a new development,[101] 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.[9,22,39]

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

$$\frac{P(x,y)}{P(y,x)} = \frac{k_c(x)}{k_c(y)} \frac{\rho(x)}{\rho(y)} \exp\left[-(x-y)/k_B T\right] \tag{4.3-4}$$

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:

$$N(y) = \frac{\int_0^y f_d(x,y)dx}{1 - \int_y^\infty \frac{f_d(y,x)}{N(x)}\frac{k_c(x)}{k_c(y)}\frac{\rho(x)}{\rho(y)}\exp\left[-(x-y)/k_BT\right]dx} \tag{4.3-11}$$

Eq. 4.3-11 can be solved with the finite difference algorithm described by Gilbert and coworkers [39,102]. 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 [1,11,22,102]. This constant is conventionally identified with k_{LJ}, the bimolecular rate constant for collisions between particles governed by a Lennard-Jones intermolecular potential.[11,22,39]

Experience has shown that the assumption the k_c is independent of energy leads to problems with normalization.[101] 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.[39,102] 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.[101]

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 $N(y)$.[101] 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:

$$k_c(y) = \frac{k_{LJ}}{N(E_{ref})} N(y) \tag{4.3-13}$$

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.[101]

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.[53] 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:

$$\delta E_d = \left| \frac{d(\ln f_d(E,E'))}{dE} \right|^{-1}, \qquad \text{for E' > E,} \qquad (4.3\text{-}14a)$$

$$\delta E_a = \left| \frac{d(\ln f_a(E',E))}{dE} \right|^{-1}, \qquad \text{for E' < E,} \qquad (4.3\text{-}14b)$$

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 $P(E,E')$.[93,103,104] 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 $N(E)$:

$$P_d(E,E') = \frac{1}{N(E')} f_d(E,E') \qquad \text{for } E' > E, \qquad (4.3\text{-}15)$$

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:[105]

$$f_d(E,E') = \exp\left\{ -\left| \frac{E'-E}{\alpha(E')} \right|^{\gamma} \right\}, \qquad \text{for } E' > E, \qquad (4.3\text{-}15)$$

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 $P_{up}(E)$:

$$P_{up}(E) = N_a(E) / N(E) \qquad \text{up-transition probability} \qquad (4.3\text{-}16)$$

$$0 \le r_3 < P_{up}(E), \qquad \text{activating} \qquad (4.3\text{-}16a)$$

$$P_{up}(E) \le r_3 < 1, \qquad \text{de-activating} \qquad (4.3\text{-}16b)$$

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]

$$r_4 = \frac{1}{N_a(E')} \int_{E'}^{E} f_a(x, E')dx , \qquad \text{activating} \qquad (4.3\text{-}17a)$$

$$r_4 = \frac{1}{N_d(E')} \int_{E'}^{E} f_d(x, E')dx , \qquad \text{de-activating} \qquad (4.3\text{-}17b)$$

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[106] 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,[107,108] 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 $y_0(E')$:

$$r_5 = Y_0(E) = \int_0^E y_0(E')dE' \qquad (5.1\text{-}1)$$

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[57,67,109] and energy distributions that are the result of bond fission.[110]

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

$$y_0^{(therm)}(E)dE = \frac{\rho(E)e^{-\frac{E}{k_B T_{vib}}}dE}{\int_0^{\infty}\rho(E')e^{-\frac{E'}{k_B T_{vib}}}dE'} \tag{5.2-1}$$

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.

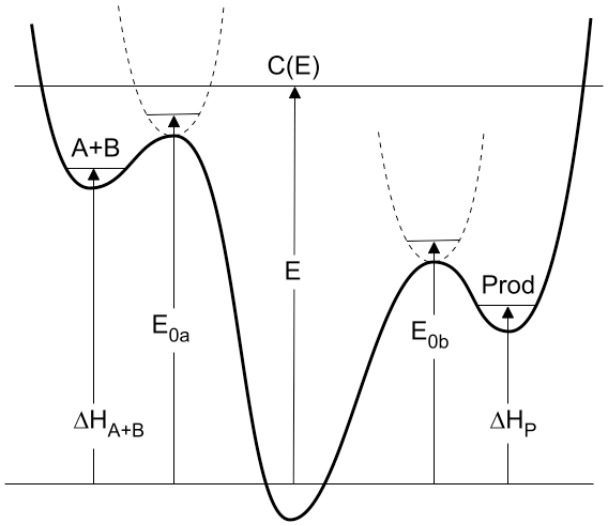


Figure 5.1. Energy diagram for chemical activation.

The chemical activation distribution function is obtained from the reverse reaction by using detailed balance.[9,11,21,22,39] 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:

$$y_0^{(ca,i)}(E)\,dE = \frac{k_a(E)\,\rho(E)\,e^{-\frac{E}{k_B T_{vib}}}\,dE}{\int_{E_{0a}}^{\infty} k_a(E')\,\rho(E')\,e^{-\frac{E'}{k_B T_{vib}}}\,dE'}, \quad \text{for } E \geq E_{0a} \tag{5.2-3}$$

$$r_5 = \int_{E_{0a}}^{E'} y_0^{(ca,i)}(E)\,dE \tag{5.2-4}$$

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.[16]

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[111,112] 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,[18,19] or according to the Stein-Rabinovitch method[7] 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 [113-115], 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 $G^{\ddagger}(E\text{-}E_0)$ or $k(E\text{-}E_0)$ 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:

$$< F(t) >= f_{react} < k(t) > \qquad (7.2\text{-}1)$$

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 D. M. Leitner, B. Levine, J. Quenneville, T. J. Martinez, and P. G. Wolynes, J. Phys. Chem. A **107** (49), 10706 (2003).
6 D. M. Leitner and P. G. Wolynes, Chem. Phys. Lett. **280** (5-6), 411 (1997).
7 S. E. Stein and B. S. Rabinovitch, J. Chem. Phys. **58** (6), 2438 (1973).
8 T. Beyer and D. F. Swinehart, Comm. Assoc. Comput. Machines **16** (6), 379 (1973).
9 W. Forst, *Theory of Unimolecular Reactions*. (Academic Press, New York, 1973).
10 W. Forst, J. Phys. Chem. **76** (3), 342 (1972).
11 W. Forst, *Unimolecular Reactions. A Concise Introduction*. (Cambridge University Press, Cambridge, 2003).
12 D. T. Gillespie, J. Phys. Chem. **81** (25), 2340 (1977).
13 D. T. Gillespie, J. Comp. Phys. **22** (4), 403 (1976).
14 D. T. Gillespie, J. Comp. Phys. **28** (3), 395 (1978).
15 J. R. Barker, Int. J. Chem. Kinetics **33** (4), 232 (2001).
16 J. R. Barker and N. F. Ortiz, Int. J. Chem. Kinetics **33** (4), 246 (2001).
17 J. L. Durant and F. Kaufman, Chem. Phys. Lett. **142** (3-4), 246 (1987).
18 G. Z. Whitten and B. S. Rabinovitch, J. Chem. Phys. **38**, 2466 (1963).
19 G. Z. Whitten and B. S. Rabinovitch, J. Chem. Phys. **41**, 1883 (1964).
20 D. C. Astholz, J. Troe, and W. Wieters, J. Chem. Phys. **70** (11), 5107 (1979).
21 P. J. Robinson and K. A. Holbrook, *Unimolecular Reactions*. (Wiley-Interscience, London; New York, 1972).
22 K. A. Holbrook, M. J. Pilling, and S. H. Robertson, *Unimolecular Reactions*, 2 ed. (Wiley, Chichester, 1996).
23 J. L. McHale, *Molecular Spectroscopy*. (Prentice Hall, Upper Saddle River, 1999).
24 C. H. Townes and A. L. Schlalow, *Microwave Spectroscopy*. (McGraw-Hill, New York, 1955).
25 A. L. L. East and L. Radom, J. Chem. Phys. **106** (16), 6655 (1997).
26 NIST, (National Institute of Science and Technology, 2005).
27 M. W. Chase, Jr., J. Phys. Chem. Ref. Data **Monograph No. 9**, 1 (1998).
28 D. M. Golden, J. Phys. Chem. A **111** (29), 6772 (2007).
29 D. M. Golden, J. Phys. Chem. A **110** (9), 2940 (2006).
30 D. M. Golden, Int. J. Chem. Kinet. **37** (10), 625 (2005).
31 D. M. Golden, J. R. Barker, and L. L. Lohr, J. Phys. Chem. A **107** (50), 11057 (2003).
32 D. M. Golden, Int. J. Chem. Kinet. **35**, 206 (2003).
33 D. M. Golden and G. P. Smith, J. Phys. Chem. A **104** (17), 3991 (2000).
34 R. Patrick and D. M. Golden, Int. J. Chem. Kinet. **15**, 1189 (1983).
35 G. P. Smith and D. M. Golden, Int. J. Chem. Kinet. **10** (5), 489 (1978).
36 V. P. Varshni, Rev. Mod. Phys. **29**, 664 (1957).

[37] 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).

[38] 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).

[39] R. G. Gilbert and S. C. Smith, *Theory of Unimolecular and Recombination Reactions*. (Blackwell Scientific, Oxford, 1990).

[40] *CRC Handbook of Physics and Chemistry, 81st Edition*, edited by D. R. Lide (CRC Press, Boca Raton, 2000).

[41] S. W. Benson, *Thermochemical Kinetics*, 2nd ed. (Wiley, New York, 1976).

[42] E. Gorin, Acta Physicochim., URSS **9**, 681 (1938).

[43] M. J. T. Jordan, S. C. Smith, and R. G. Gilbert, J. Phys. Chem. **95**, 8685 (1991).

[44] D. M. Wardlaw and R. A. Marcus, J. Phys. Chem. **90** (21), 5383 (1986).

[45] D. W. Wardlaw and R. A. Marcus, J. Chem. Phys. **110** (3), 230 (1984).

[46] J. A. Miller and S. J. Klippenstein, J. Phys. Chem. A **104** (10), 2061 (2000).

[47] S. J. Klippenstein, A. F. Wagner, S. H. Robertson, R. Dunbar, and D. M. Wardlaw, VariFlex Software (1999).

[48] A. P. Penner and W. Forst, CP **13**, 51 (1976).

[49] A. P. Penner and W. Forst, CP **11**, 243 (1976).

[50] S. C. Smith and R. G. Gilbert, Int. J. Chem. Kinet. **20**, 979 (1988).

[51] S. C. Smith and R. G. Gilbert, Int. J. Chem. Kinet. **20**, 307 (1988).

[52] J. Troe, J. Chem. Phys. **66**, 4745 (1977).

[53] 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).

[54] S. C. Smith, M. J. McEwan, and R. G. Gilbert, J. Chem. Phys. **90**, 1630 (1989).

[55] S. J. Jeffrey, K. E. Gates, and S. C. Smith, J. Phys. Chem. **100** (17), 7090 (1996).

[56] S. H. Robertson, A. I. Shushin, and D. M. Wardlaw, J. Chem. Phys. **98** (11), 8673 (1993).
[57] P. K. Venkatesh, A. M. Dean, M. H. Cohen, and R. W. Carr, J. Chem. Phys. **107** (21), 8904 (1997).
[58] T. Baer and W. L. Hase, *Unimolecular Reaction Dynamics. Theory and Experiments*. (Oxford University Press, New York, 1996).
[59] R. Meyer, J. Chem. Phys. **52**, 2053 (1970).
[60] C. C. Marston and G. G. Balint-Kurti, J. Chem. Phys. **91**, 3571 (1989).
[61] D. T. Colbert and W. H. Miller, J. Chem. Phys. **96**, 1982 (1992).
[62] B. M. Wong, R. L. Thom, and R. W. Field, J. Phys. Chem. A. **110**, 7406 (2006).
[63] 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.
[64] 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).
[65] L. Goodman, A. G. Ozkabak, and S. N. Thakur, J. Phys. Chem. **95**, 9044 (1991).
[66] D. T. Gillespie, Physica A (Amsterdam) **188**, 404 (1992).
[67] L. Vereecken, G. Huyberechts, and J. Peeters, J. Chem. Phys. **106** (16), 6564 (1997).
[68] D. E. Knuth, *Seminumerical Algorithms*, 2 ed. (Addison-Wesley, Reading, MA, 1981).
[69] W. H. Press and S. A. Teukolsky, Computers in Physics **6** (5), 522 (1992).
[70] C. P. Tsokos, *Probability Distributions: An Introduction to Probability Theory with Applications*. (Wadsworth Publishing Company Inc., Belmont CA, 1972).
[71] M. Quack and J. Troe, berichte **78** (3), 240 (1974).
[72] 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).
[73] J. R. Barker, L. L. Lohr, R. M. Shroll, and S. Reading, J. Phys. Chem. A. **107** (38), 7434 (2003).
[74] W. Forst, J. Phys. Chem. **83** (1), 100 (1983).
[75] I. Oref and B. S. Rabinovitch, Acc. Chem. Res. **12** (5), 166 (1979).
[76] K. Bolton and S. Nordholm, Chem. Phys. **207** (1), 63 (1996).
[77] K. Bolton and S. Nordholm, Chem. Phys. **206** (1-2), 103 (1996).
[78] K. Bolton and S. Nordholm, Chem. Phys. **203** (1), 101 (1996).
[79] S. Nordholm and A. Back, Phys. Chem. Chem. Phys. **3** (12), 2289 (2001).
[80] M. Gruebele and P. G. Wolynes, Acc. Chem. Res. **37** (4), 261 (2004).
[81] D. M. Leitner and P. G. Wolynes, Ach-Models in Chemistry **134** (5), 658 (1997).
[82] D. M. Leitner and P. G. Wolynes, Phys. Rev. Lett. **76** (2), 216 (1996).
[83] 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).
[84] R. E. Weston, Jr. and J. R. Barker, J. Phys. Chem. A. **110**, 7888 (2006).
[85] C. Eckart, Phys. Rev. **35** (11), 1303 (1930).
[86] W. H. Miller, J. Am. Chem. Soc **101** (23), 6810 (1979).
[87] M. Olzmann, Phys. Chem. Chem. Phys. **4**, 3614–3618 (2002).

[88] N. W. Moriarity and M. Frenklach, presented at the Twenty Eighth Symposium (International) on Combustion, University of Edinburgh, Edinburgh, Scotland, 2000 (unpublished).
[89] R. A. Marcus, J. Chem. Phys. **45**, 2138 (1966).
[90] J. I. Steinfeld, J. S. Francisco, and W. L. Hase, *Chemical Kinetics and Dynamics*, 2 ed. (Prentice-Hall, 1998).
[91] A. Maranzana, J. R. Barker, and G. Tonachini, Phys. Chem. Chem. Phys. **9** (31), 4129 (2007).
[92] C. Wittig, I. Nadler, H. Reisler, M. Noble, J. Catanzarite, and G. Radhakrishnan, J. Chem. Phys. **83** (11), 5581 (1985).
[93] D. C. Tardy and B. S. Rabinovitch, Chem. Rev. **77**, 369 (1977).
[94] J. O. Hirschfelder, C. F. Curtiss, and R. B. Bird, *Molecular Theory of Gases and Liquids*. (Wiley, New york, 1964).
[95] G. Lendvay and G. C. Schatz, J. Phys. Chem. **96** (9), 3752 (1992).
[96] W. D. Lawrance and A. E. W. Knight, J. Chem. Phys. **79** (12), 6030 (1983).
[97] A. J. Stace and J. N. Murrell, J. Chem. Phys. **68** (7), 3028 (1978).
[98] L. M. Yoder and J. R. Barker, J. Phys. Chem. A **104** (45), 10184 (2000).
[99] B. Xue, J. Han, and H.-L. Dai, Phys. Rev. Lett. **84** (12), 2606 (2000).
[100] J. T. Yardley, *Introduction to Molecular Energy Transfer*. (Academic Press, New York, 1980).
[101] J. R. Barker, Int. J. Chem. Kinetics, submitted for publication (copies available from the author at jrbarker@umich.edu) (2009).
[102] R. G. Gilbert and K. D. King, Chem. Phys. **49**, 367 (1980).
[103] J. R. Barker, L. M. Yoder, and K. D. King, J. Phys. Chem. A **105** (5), 796 (2001).
[104] I. Oref and D. C. Tardy, Chem. Rev. **90**, 1407 (1990).
[105] U. Hold, T. Lenzer, K. Luther, K. Reihs, and A. C. Symonds, J. Chem. Phys. **112** (9), 4076 (2000).
[106] J. F. Durana and J. D. McDonald, J. Chem. Phys. **64**, 2518 (1977).
[107] J. R. Barker, J. Chem. Phys. **72** (6), 3686 (1980).
[108] D. M. Golden, M. J. Rossi, A. C. Baldwin, and J. R. Barker, Acc. Chem. Res. **14**, 56 (1981).
[109] P. Urbain, B. Leyh, F. Remacle, and A. J. Lorquet, J. Chem. Phys. **110** (6), 2911 (1999).
[110] W. Tsang, V. Bedanov, and M. R. Zachariah, (1996).
[111] N. Davidson, *Statistical Mechanics*. (McGraw-Hill Book Company, Inc., New York, 1962).
[112] J. E. Mayer and M. G. Mayer, *Statistical Mechanics*. (John Wiley & Sons, Inc., New York, 1940).
[113] J. R. Barker, J. Phys. Chem. **91**, 3849 (1987).
[114] B. M. Toselli and J. R. Barker, J. Phys. Chem. **93**, 6578 (1989).
[115] B. M. Toselli and J. R. Barker, Chem. Phys. Lett. **159**, 499 (1989).

Index