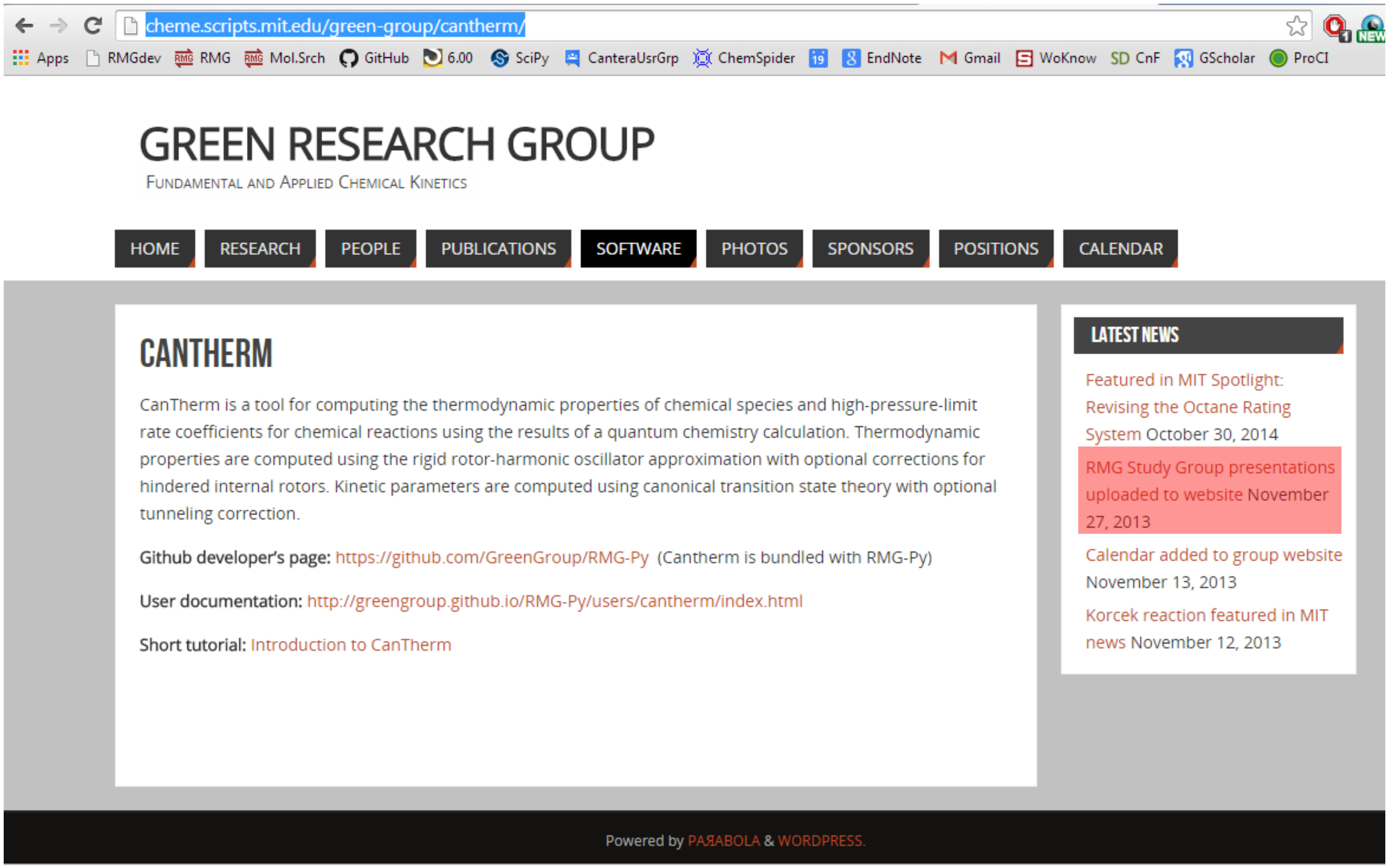
CanTherm Refresher/Overview

Enoch Dames

RMG Study Group Meeting

Jan. 12, 2015

Online Resources: http://cheme.scripts.mit.edu/green-group/cantherm/
http://greengroup.github.io/RMG-Py/theory/measure/index.html
http://cccbdb.nist.gov/ - tables of force constant scaling factors, lots of explanations and tutorials


cheme.scripts.mit.edu/green-group/cantherm/
Apps RMGdev RMG Mol.Srch GitHub 6.00 SciPy CanteraUsrGrp ChemSpider EndNote Gmail WoKnow CnF GScholar ProCI
HOME RESEARCH PEOPLE PUBLICATIONS SOFTWARE PHOTOS SPONSORS POSITIONS CALENDAR

GREEN RESEARCH GROUP

Fundamental and Applied Chemical Kinetics

CANTHERM

CanTherm is a tool for computing the thermodynamic properties of chemical species and high-pressure-limit rate coefficients for chemical reactions using the results of a quantum chemistry calculation. Thermodynamic properties are computed using the rigid rotor-harmonic oscillator approximation with optional corrections for hindered internal rotors. Kinetic parameters are computed using canonical transition state theory with optional tunneling correction.

Github developer's page: https://github.com/GreenGroup/RMG-Py (Cantherm is bundled with RMG-Py)

User documentation: http://greengroup.github.io/RMG-Py/users/cantherm/index.html

Short tutorial: Introduction to CanTherm

LATEST NEWS

Featured in MIT Spotlight: Revising the Octane Rating System October 30, 2014

RMG Study Group presentations uploaded to website November 27, 2013

Calendar added to group website November 13, 2013

Korcek reaction featured in MIT news November 12, 2013

Powered by PARABOLA & WORDPRESS.

Outline of this RMG Study Group

- What is CanTherm? How is it used?
- The world’s most compact overview of the theory behind rate theory packages (with emphasis on kinetics)
- Running CanTherm
- Complex Pdep Example Calculation, I/O components

Objective of this RMG Study Group

Provide basic information and conduct a brief overview of topics necessary for computing pressure dependent rates using CanTherm

What is CanTherm?

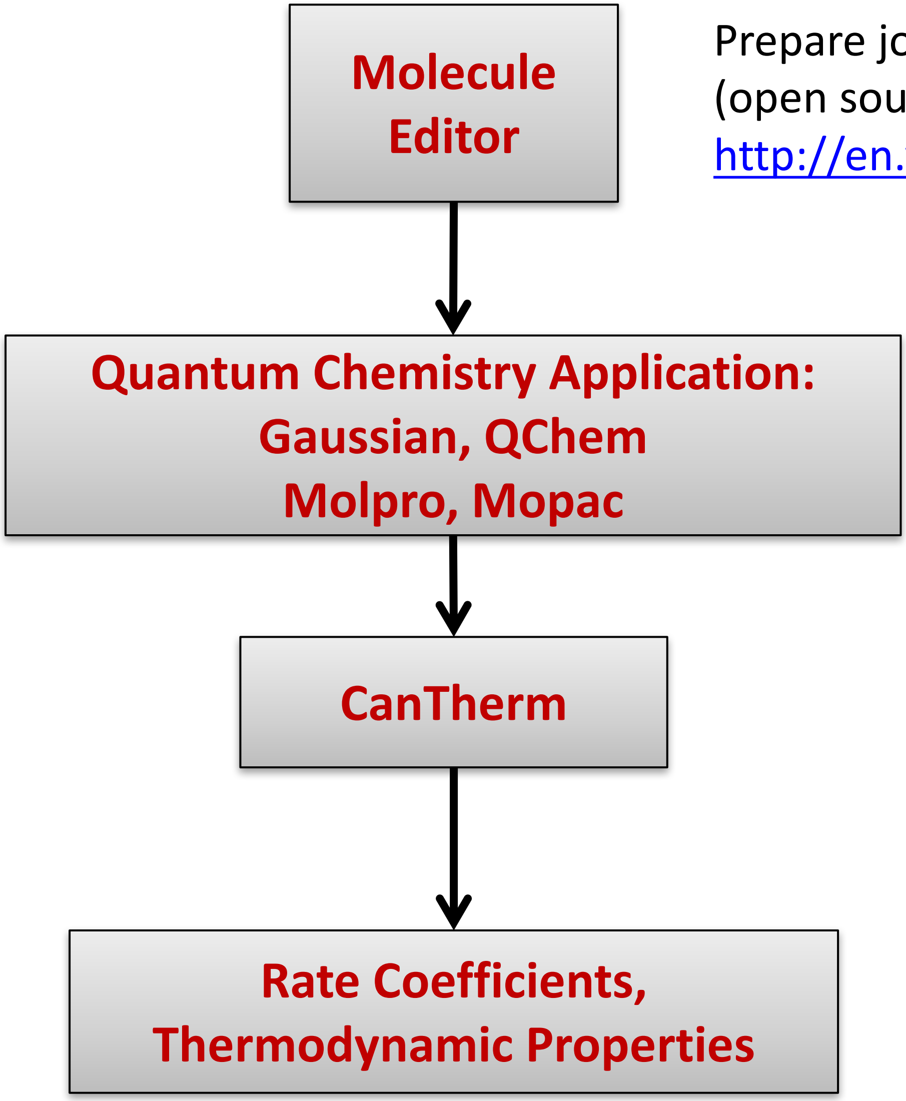
CanTherm is an open source python package of utilities for the computation of the following:

1. Thermodynamic properties of stable molecules (H_{298}, S , $C_p(T)$)
   (see Shamel’s study group presentation #5 for more)
2. High pressure limit rate coefficients, k_∞
3. Pressure dependent rate coefficients, *k(T,P),* for arbitrarily large multiple-well reaction networks using either Modified Strong Collision, Reservoir State or Chemically Significant Eigenvalue (CSE) approximations

Notes:

- CanTherm does not have a GUI
- There are numerous other similar codes out there, but CanTherm has the nice feature that many molecular properties can be automatically read in from outputs of quantum chemistry jobs
- If you forked over a copy of RMG-Py from Github, you have CanTherm

How CanTherm Is Used


Molecule Editor
Quantum Chemistry Application: Gaussian, QChem Molpro, Mopac
CanTherm
Rate Coefficients, Thermodynamic Properties

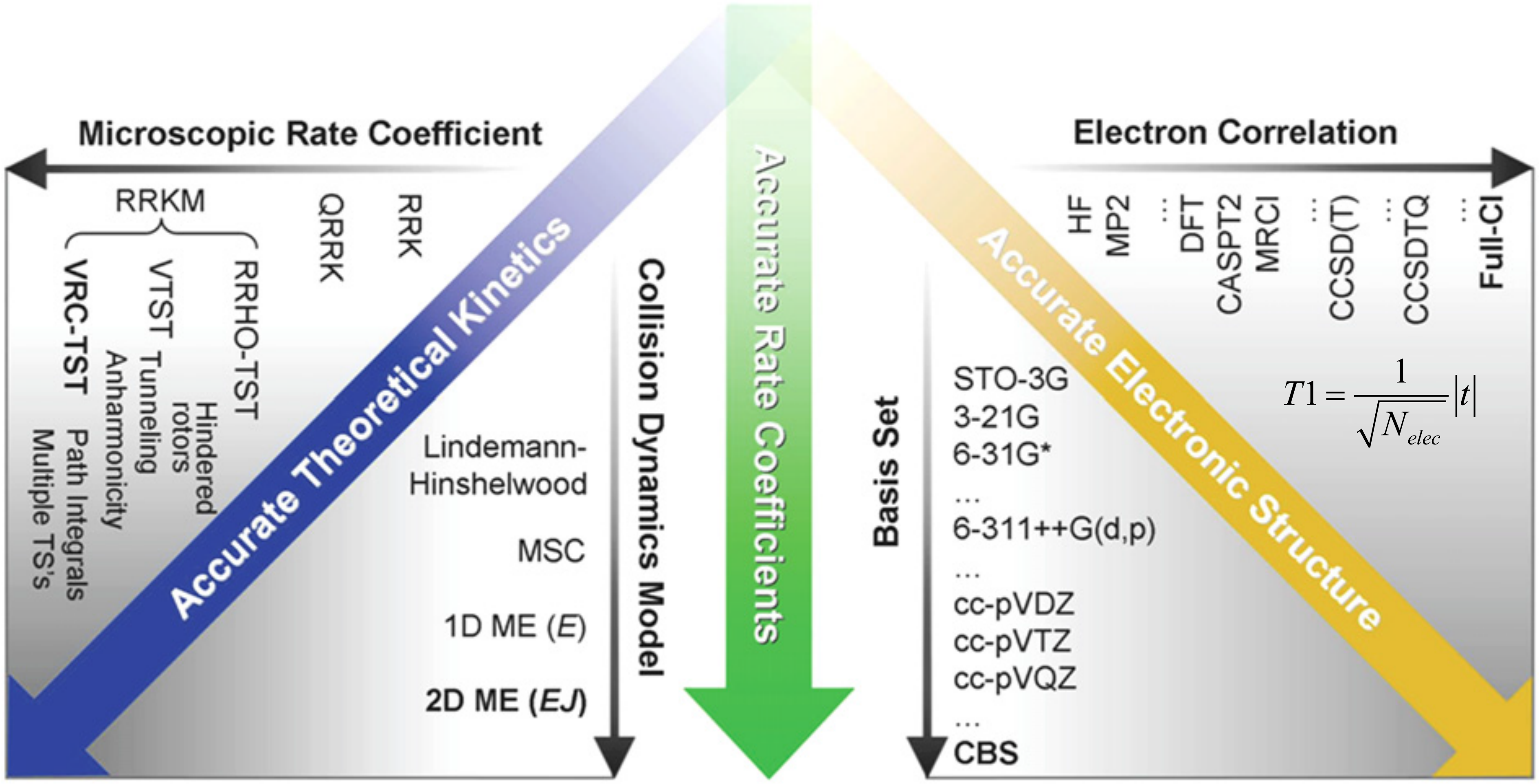
Prepare jobs *via* GaussView, WebMO, Avagadro (open source), etc. See: http://en.wikipedia.org/wiki/Molecule_editor

Run jobs to obtain energies, frequencies

Compute *k(T,P)*, thermo parameters

Use *k(T,P)*, thermo parameters for science

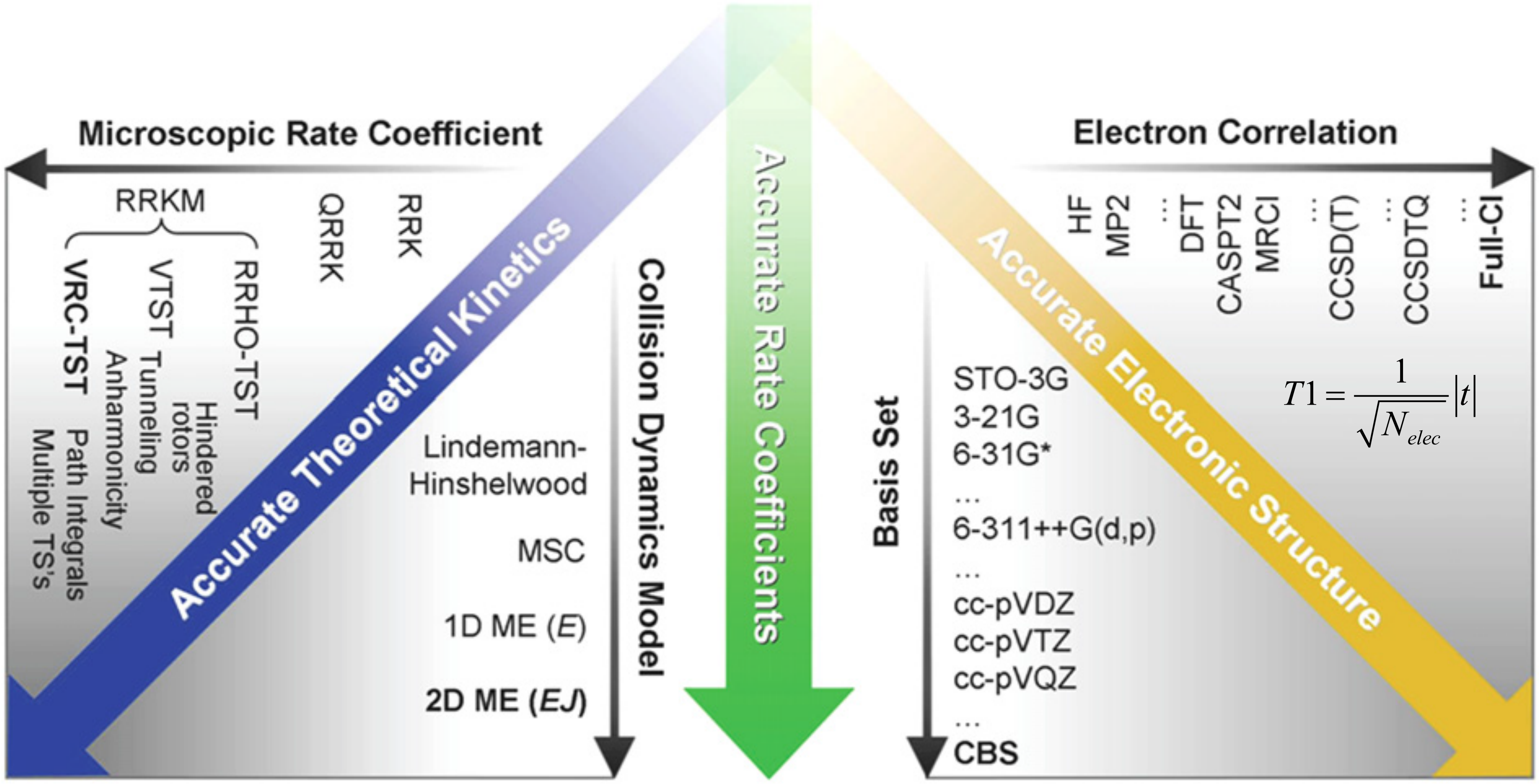
Electronic Structure and Rates: varying levels of theory



Zador et al 2010 Prog. Energy. Combust. Sci.

Best practices: always make an attempt to validate or verify the accuracy of your methods, either through comparison with experiments or benchmark calculations

Electronic Structure and Rates: varying levels of theory



$$T1 = \frac{1}{\sqrt{N_{elec}}}|t|$$

Zador et al 2010 Prog. Energy. Combust. Sci.

Q: Which model chemistry is right for you?
A: depends on the level of accuracy you require, computational resources (time)

Sub-orbital space view: differences between HF, post-HF, and DFT

- Hartree Fock (HF) theory is a way to variationally estimate the energy of a system of electrons and nuclei, but neglects electron correlation (mean field apprx).
- post HF methods are advancements of HF that add electron correlation as opposed to simply averaging it out
- Density Functional Theory (DFT):
  - Computationally faster, scales better with size
  - Focus is on electron density rather than wavefunction
  - Molecular energy is a function of electron density is a function of spacial coordinates (position), hence the name D**F**T
  - Many DFT methods are semi-empirical (i.e., trained against a experimentally derived dataset)
- Hybrid or Composite methods: model chemistries involving both HF and DFT components, designed to yield accurate energies at reduced computational costs (e.g., CBS-QB3)

$$H\Psi = E\Psi$$

Electronic structure calculations only provide geometries, *relative* energies, force constants, and sometimes, correct point groups necessary for calculation of rates and thermo properties

Symmetry Numbers, Point Groups: Important for A-factors and thermo

Things to know:

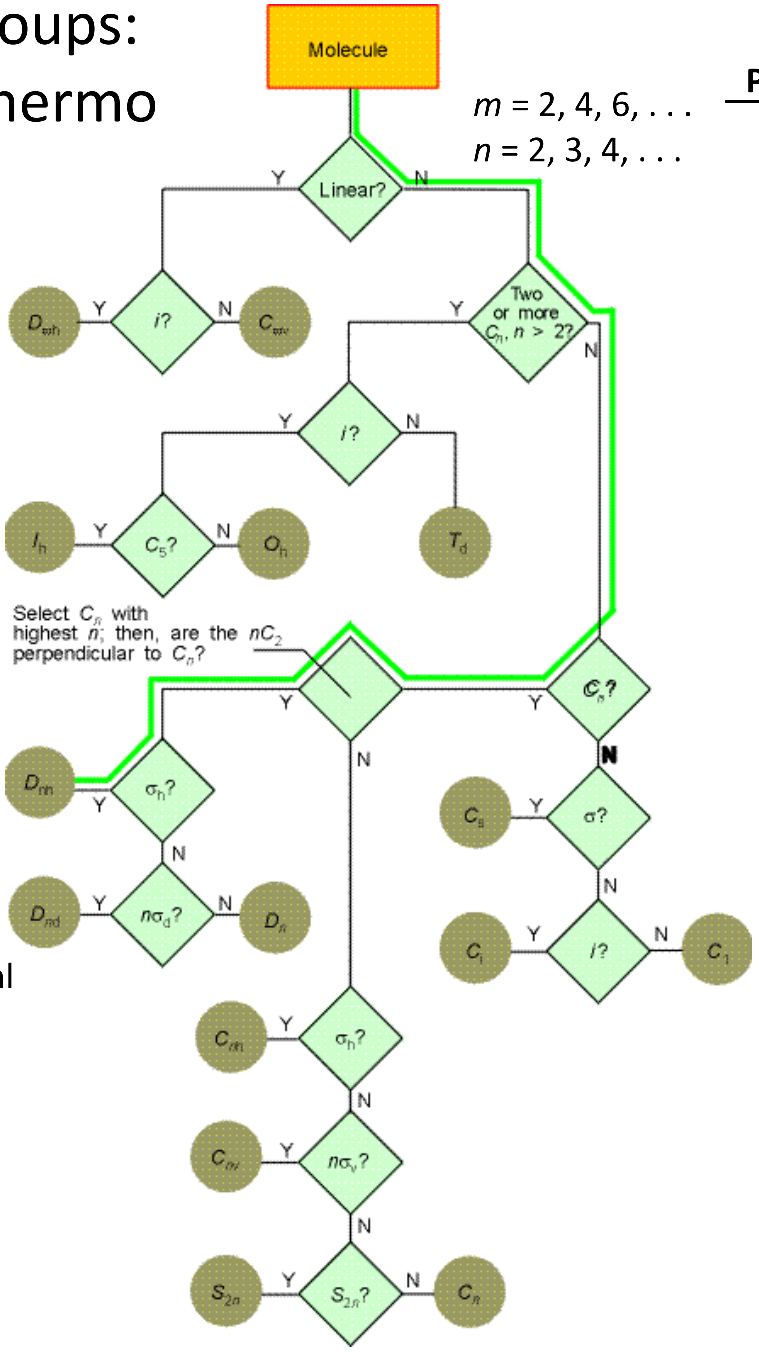
1. Symmetry operations
2. How to identify point groups
3. The rotational symmetry corresponding to various point groups

Tips:

- Flowcharts help. If you can perform basic symmetry operations, you can use a flowchart.
- Many online resources/tutorials

Rotational symmetry reduces a molecule's entropy by a factor of $R\ln(\sigma)$, where σ is the rotational symmetry number and R the gas constant. Example: a C_{60} Buckminsterfullerene belongs to the I_h point group and has a rotational symmetry of 60. Neglecting the rotational contribution to entropy results in an error of over 8 cal/mol-K in an estimation of its standard state entropy.

Question. How does the rotational symmetry of cyclohexane change with temperature?



$m = 2, 4, 6, \ldots$

$n = 2, 3, 4, \ldots$

Point Group	**σ**
C_1	1
C_i	1
C_s	1
C_{2v}	2
$C_{\infty v}$	1
$D_{\infty h}$	2
S_m	$m/2$
C_n	n
C_{nv}	n
D_n	$2n$
D_{nh}	$2n$
D_{nd}	$2n$
T	12
T_d	12
O_h	24
I_h	60

An effort in futility: statistical mechanics in one slide

$$Q(N,V,T) = \sum_i e^{-E_i(N,V)/k_B T}$$

The canonical partition function (e.g., macroscopic), *Q*, is summed over all energy levels of a 'system'

$$Q(N,V,T) = \frac{\left[q(V,T)\right]^N}{N!}$$

Under the ideal gas assumption, we can rewrite the canonical Partition function as a function of the molecular partition function

$$q_{tot}(V,T) = \sum_i e^{-E_i/k_B T}$$

We typically assume that molecular degrees of freedom may be uncoupled:

$$q_{tot}(V,T) = q_{elec}(T)\,q_{trans}(V,T)\,q_{rot}(T)\,q_{vib}(T)$$

$$q_{elec}(T) = g_1 + g_2 e^{-E_2/k_B T} + \cdots$$

$$q_{rot,3D}(V,T) = \frac{\pi^{1/2}}{\sigma}\sqrt{\frac{k_B T}{B_x}}\sqrt{\frac{k_B T}{B_y}}\sqrt{\frac{k_B T}{B_z}}$$

$$q_{trans}(V,T) = \left(\frac{2\pi M k_B T}{h^2}\right)^{3/2} V$$

$$q_{vib}(T) = \frac{e^{h\nu/2k_B T}}{1 - e^{-h\nu/2k_B T}}$$

We use these relations to derive standard thermodynamic properties:

$$U = k_B T^2 \left(\frac{\partial \ln Q}{\partial T}\right)_{N,V} \qquad S = \ln Q + k_B T \left(\frac{\partial \ln Q}{\partial T}\right)_{N,V}$$

Transition state theory gives only the high-pressure limit rate, for most reactions

$$k_{\infty}\left(T\right) = l\,\kappa \frac{k_B T}{h} \frac{Q_{tot}^{\dagger}}{Q_{tot}} \exp\left({}^{-E_0}\!\Big/_{k_B T} \right)$$

Conventional TST fails for some systems:

- Barrierless reactions. Must use variational or other methods
- Systems with many possible transition states

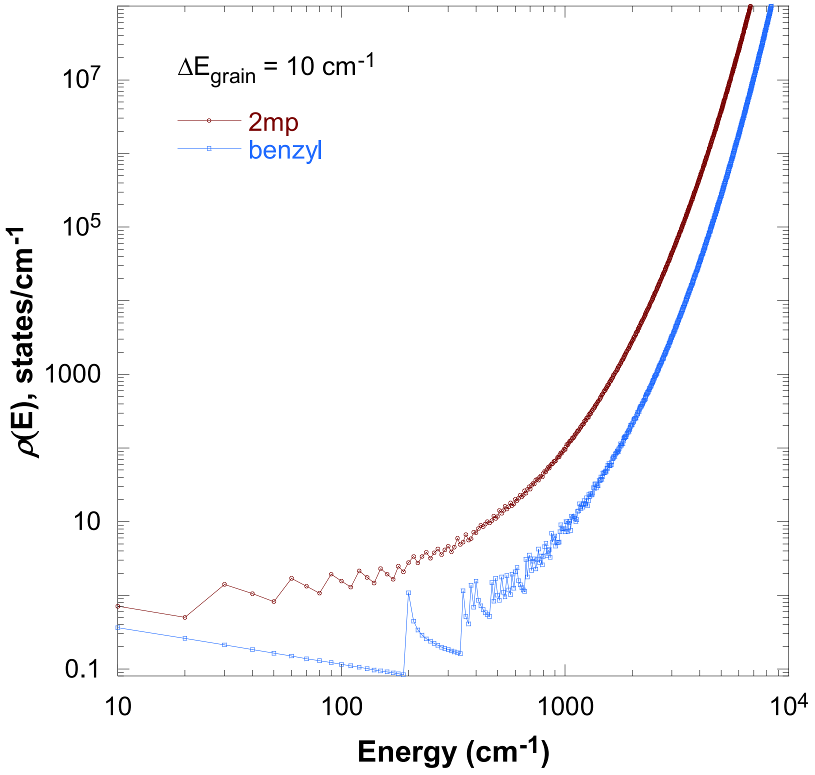
RRKM theory is used in the context of the master equation for energy transfer to compute pressure dependence

$$N(E) = \sum_i \partial(E - E_i)$$

$$\rho(E) = {dN(E)}\Big/{dE}$$

RRKM rate: $$k(E) = \frac{N^\dagger(E)}{h\rho(E)}$$

$$Q = \int_0^\infty \rho(E)\exp\left(-E\Big/{k_B T}\right)dE$$



CanTherm counts the density of states using the method of steepest decents, which has been shown to be accurate and faster than direct counting.

Pressure Dependence – a unimolecular perspective

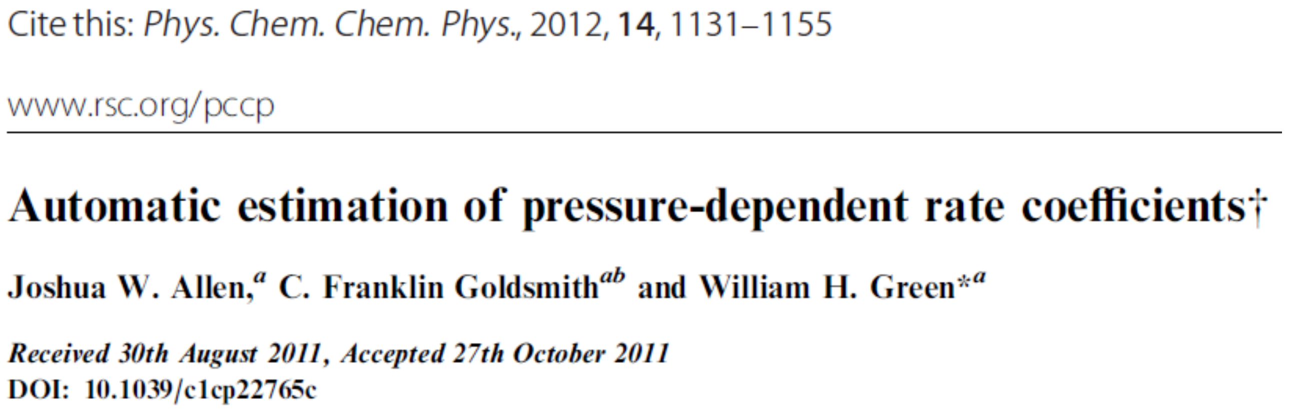
The unimolecular dissociation process is captured by the well-known Lindemann-Hinshelwood mechanism:

$$A + M \xrightarrow{k_f} A^* + M$$

$$A^* + M \xrightarrow{k_b} A + M$$

$$A^* \xrightarrow{k_2} products$$

Read Josh Allen's Pdep paper for an in depth discussion:



Cite this: *Phys. Chem. Chem. Phys.*, 2012, **14**, 1131–1155

www.rsc.org/pccp

Automatic estimation of pressure-dependent rate coefficients†

Joshua W. Allen,[a] C. Franklin Goldsmith[ab] and William H. Green*[a]

Received 30th August 2011, Accepted 27th October 2011
DOI: 10.1039/c1cp22765c

The master equation

- The master equation in chemical kinetics describes the time evolution of a reaction network
- Consider a reactant, A, with 3N degrees of freedom, depending on the surrounding T and bath gas
- A is more accurately envisioned as A(E*ᵢ*)

$$\frac{d\left[A(E_i)\right]}{dt} = \begin{bmatrix}\text{rate of collisional } \textit{production} \\ \text{of } A \text{ at energy level } j\end{bmatrix} - \begin{bmatrix}\text{collisional rate } \textit{loss} \text{ of} \\ A \text{ at energy level } i\end{bmatrix} - \begin{bmatrix}\text{rate } \textit{loss} \text{ of } A_i \\ \text{due to reaction}\end{bmatrix}$$

$$\frac{d\left[A(E_i)\right]}{dt} = Z[M]\sum_j \left\{P_{ij}\left[A(E_j)\right] - P_{ji}\left[A(E_i)\right]\right\} - \sum_m k_m(E_i)\left[A(E_i)\right]$$

Collision rate and frequency:

$$Z = \sigma_{ij}^2 \sqrt{\frac{8k_B T}{\mu}}\,\Omega_{ij}^{(2,2)} N_a \quad cm^3 mol^{-1} s^{-1}$$

Microcanonical rate constant:

$$k(E) = l_a \frac{Q_{r,in}^{\dagger}}{Q_{r,in}} \frac{W'\left(E^{\dagger}\right)}{h\rho(E)} \qquad s^{-1}$$

The master equation (2)

- The probability of energy transfer is related to the energy transfer upon collision with bath gas

$$P\alpha\left\langle \Delta E_d \right\rangle$$

- The average downward energy transferred is bath gas (and reactant) dependent and typically a function of temperature

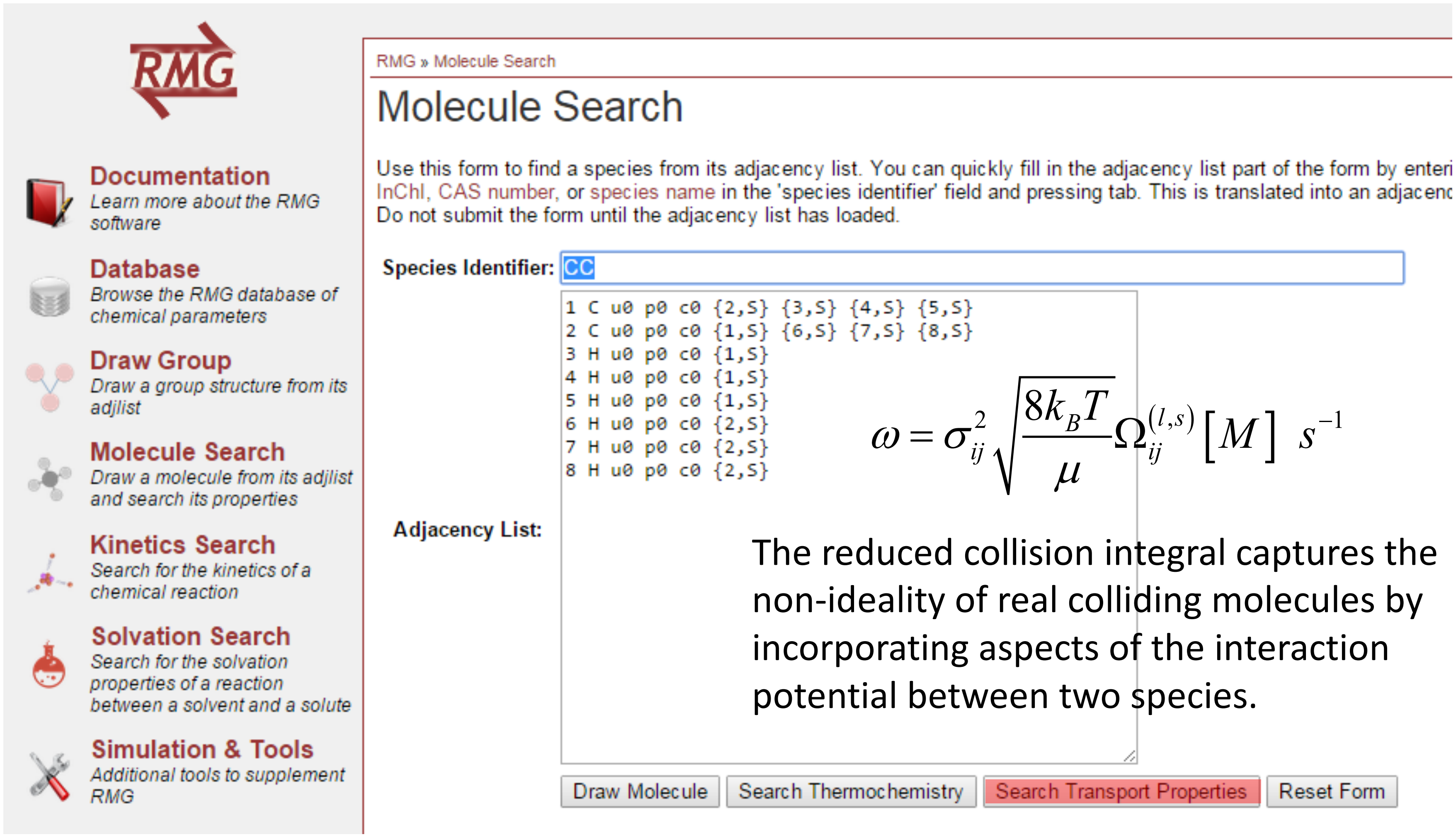
$$\left\langle \Delta E_d \right\rangle = \left\langle \Delta E_d \right\rangle_{300} \left(T/_{300} \right)^n cm^{-1}$$

Sources:

- Empirically derived
- Computed
- Tuned

Collision Frequency, Lennard Jones Parameters

Gas-Kinetic theory is used to compute the collision frequency. Species' 6-12 Lennard-Jones parameters are needed to compute the reduced collision integral.



RMG » Molecule Search

Molecule Search

Use this form to find a species from its adjacency list. You can quickly fill in the adjacency list part of the form by enteri
InChI, CAS number, or species name in the 'species identifier' field and pressing tab. This is translated into an adjacenc
Do not submit the form until the adjacency list has loaded.

Species Identifier: CC

Adjacency List:

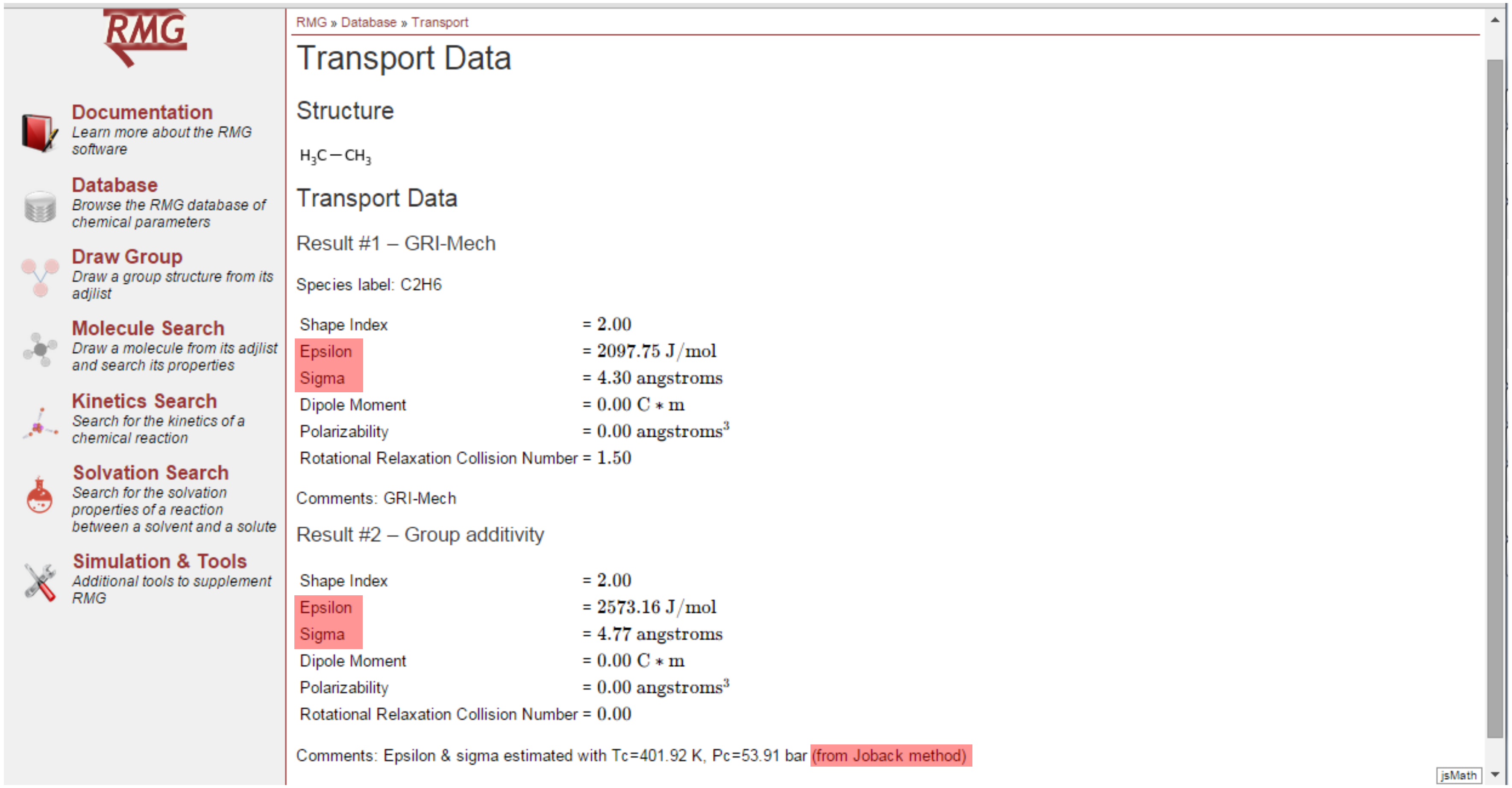
```
1 C u0 p0 c0 {2,S} {3,S} {4,S} {5,S}
2 C u0 p0 c0 {1,S} {6,S} {7,S} {8,S}
3 H u0 p0 c0 {1,S}
4 H u0 p0 c0 {1,S}
5 H u0 p0 c0 {1,S}
6 H u0 p0 c0 {2,S}
7 H u0 p0 c0 {2,S}
8 H u0 p0 c0 {2,S}
```

RMG
Documentation
Learn more about the RMG software
Database
Browse the RMG database of chemical parameters
Draw Group
Draw a group structure from its adjlist
Molecule Search
Draw a molecule from its adjlist and search its properties
Kinetics Search
Search for the kinetics of a chemical reaction
Solvation Search
Search for the solvation properties of a reaction between a solvent and a solute
Simulation & Tools
Additional tools to supplement RMG
Draw Molecule
Search Thermochemistry
Search Transport Properties
Reset Form

$$\omega = \sigma_{ij}^{2}\sqrt{\frac{8k_BT}{\mu}}\Omega_{ij}^{(l,s)}\left[M\right]\ s^{-1}$$

The reduced collision integral captures the non-ideality of real colliding molecules by incorporating aspects of the interaction potential between two species.

Online RMG resources make life easier


RMG
Documentation
Learn more about the RMG software
Database
Browse the RMG database of chemical parameters
Draw Group
Draw a group structure from its adjlist
Molecule Search
Draw a molecule from its adjlist and search its properties
Kinetics Search
Search for the kinetics of a chemical reaction
Solvation Search
Search for the solvation properties of a reaction between a solvent and a solute
Simulation & Tools
Additional tools to supplement RMG
RMG » Database » Transport
jsMath

Transport Data

Structure

H_3C-CH_3

Transport Data

Result #1 – GRI-Mech

Species label: C2H6

Shape Index	= 2.00
Epsilon	= 2097.75 J/mol
Sigma	= 4.30 angstroms
Dipole Moment	= 0.00 C ∗ m
Polarizability	= 0.00 angstroms^3
Rotational Relaxation Collision Number	= 1.50

Comments: GRI-Mech

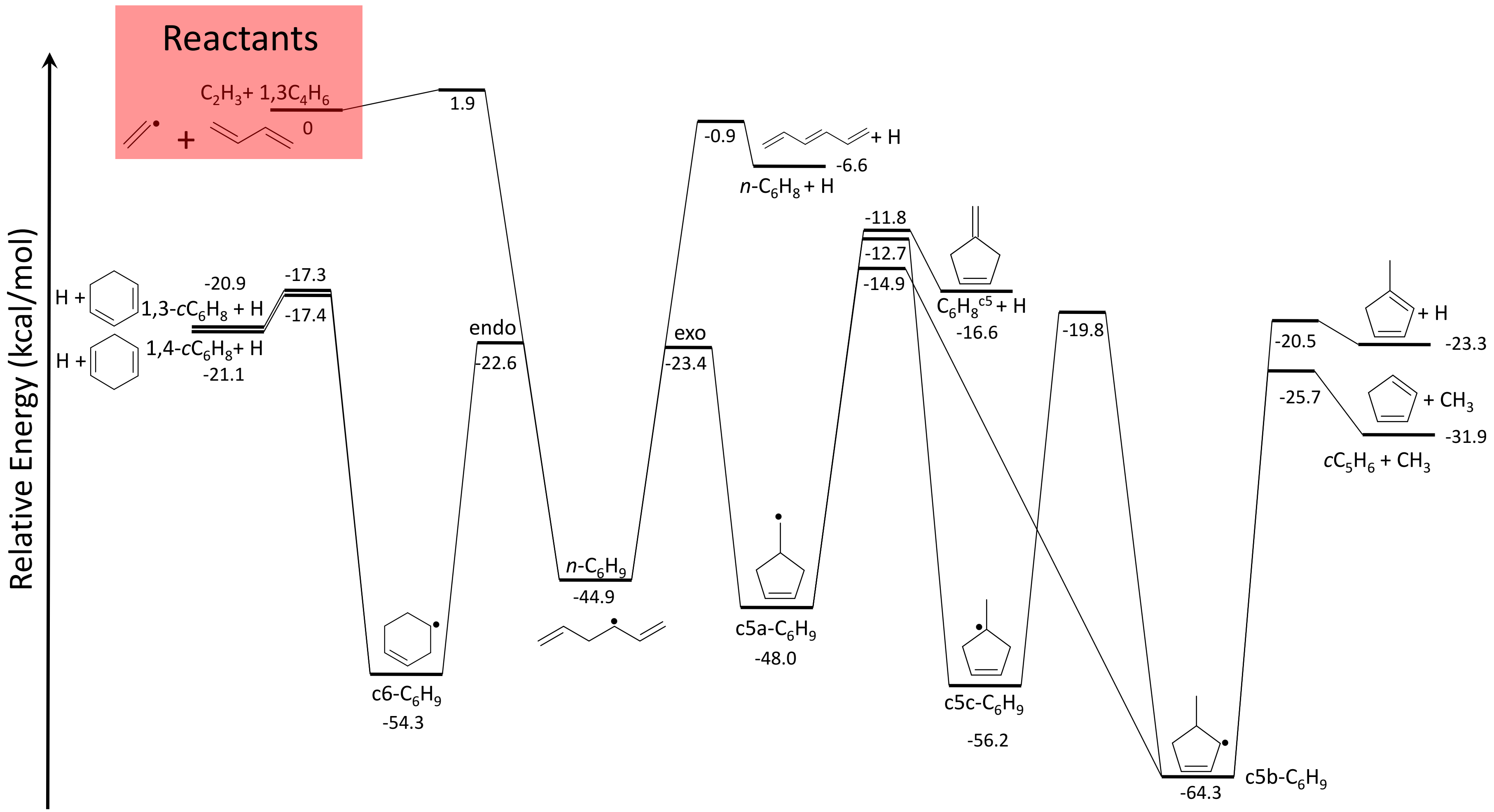
Result #2 – Group additivity

Shape Index	= 2.00
Epsilon	= 2573.16 J/mol
Sigma	= 4.77 angstroms
Dipole Moment	= 0.00 C ∗ m
Polarizability	= 0.00 angstroms^3
Rotational Relaxation Collision Number	= 0.00

Comments: Epsilon & sigma estimated with Tc=401.92 K, Pc=53.91 bar (from Joback method)

The Joback method is one of corresponding states that relates the critical temperature and pressure of molecules to their LJ-parameters

Example – large multi-well system: vinyl + butadiene



5 wells, 6 product channels, 12 transition states → 47+ separate input and Gaussian/Qchem files needed (not inlcuding HRs)!

Cantherm input file components – piece by piece

The first few lines:

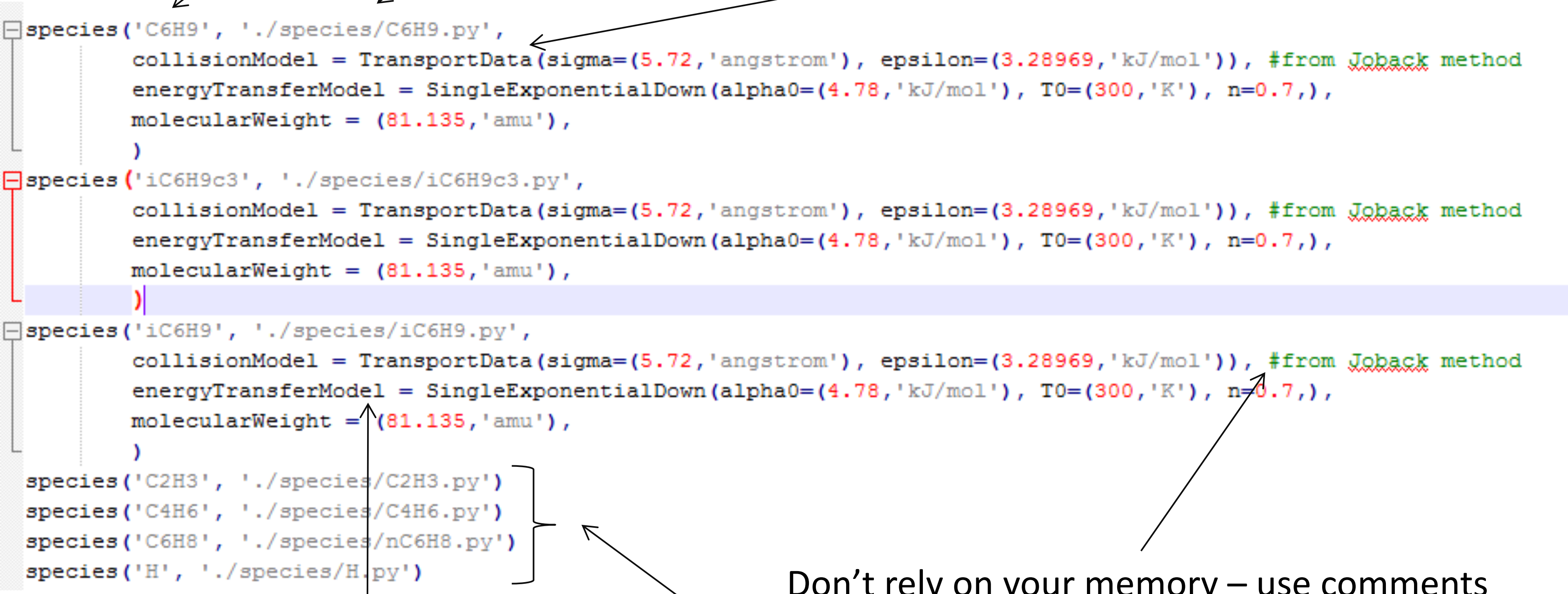
```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

modelChemistry = "M08SO/MG3S*“
frequencyScaleFactor = 0.985
useHinderedRotors = True
useBondCorrections = False
```

Cantherm input file components – species cards

label

species file name and location

Lennard-Jones 6-12 parameters



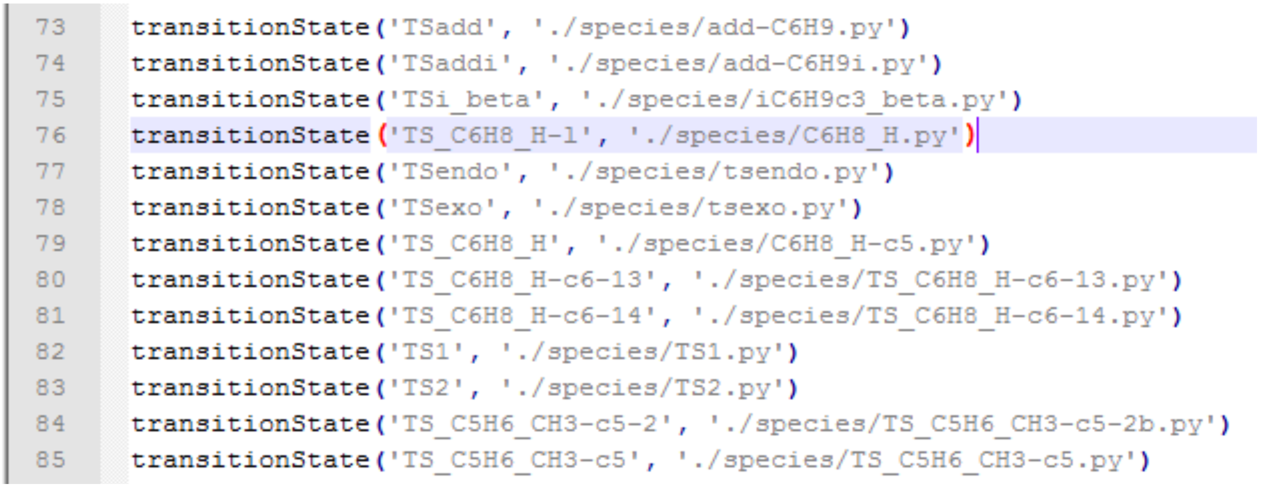
```
species('C6H9', './species/C6H9.py',
        collisionModel = TransportData(sigma=(5.72,'angstrom'), epsilon=(3.28969,'kJ/mol')), #from Joback method
        energyTransferModel = SingleExponentialDown(alpha0=(4.78,'kJ/mol'), T0=(300,'K'), n=0.7,),
        molecularWeight = (81.135,'amu'),
        )
species('iC6H9c3', './species/iC6H9c3.py',
        collisionModel = TransportData(sigma=(5.72,'angstrom'), epsilon=(3.28969,'kJ/mol')), #from Joback method
        energyTransferModel = SingleExponentialDown(alpha0=(4.78,'kJ/mol'), T0=(300,'K'), n=0.7,),
        molecularWeight = (81.135,'amu'),
        )
species('iC6H9', './species/iC6H9.py',
        collisionModel = TransportData(sigma=(5.72,'angstrom'), epsilon=(3.28969,'kJ/mol')), #from Joback method
        energyTransferModel = SingleExponentialDown(alpha0=(4.78,'kJ/mol'), T0=(300,'K'), n=0.7,),
        molecularWeight = (81.135,'amu'),
        )
species('C2H3', './species/C2H3.py')
species('C4H6', './species/C4H6.py')
species('C6H8', './species/nC6H8.py')
species('H', './species/H.py')
```

Don't rely on your memory – use comments

bimolecular products don't need energy transfer components

$$\langle \Delta E_d \rangle = \langle \Delta E_d \rangle_{300} \left(\frac{T}{300\ \mathrm{K}} \right)^n \mathrm{cm}^{-1}$$

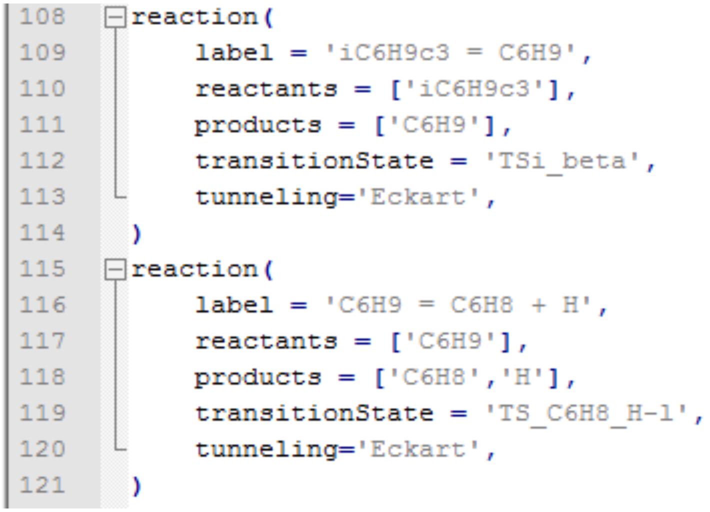
Cantherm input file components – transition states



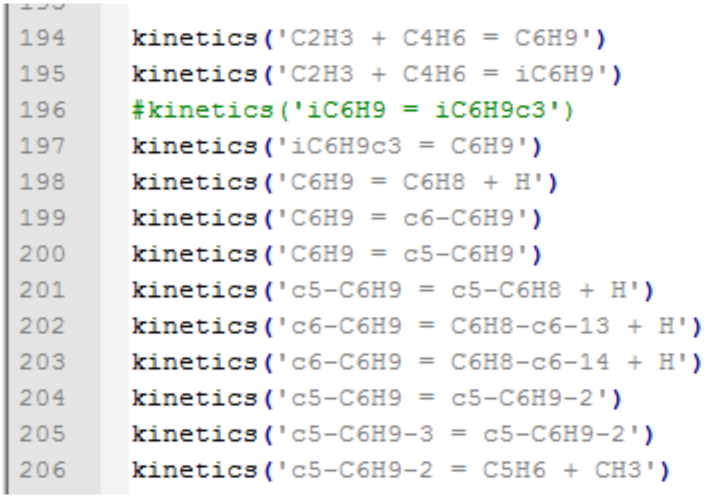
```
transitionState('TSadd', './species/add-C6H9.py')
transitionState('TSaddi', './species/add-C6H9i.py')
transitionState('TSi_beta', './species/iC6H9c3_beta.py')
transitionState('TS_C6H8_H-1', './species/C6H8_H.py')
transitionState('TSendo', './species/tsendo.py')
transitionState('TSexo', './species/tsexo.py')
transitionState('TS_C6H8_H', './species/C6H8_H-c5.py')
transitionState('TS_C6H8_H-c6-13', './species/TS_C6H8_H-c6-13.py')
transitionState('TS_C6H8_H-c6-14', './species/TS_C6H8_H-c6-14.py')
transitionState('TS1', './species/TS1.py')
transitionState('TS2', './species/TS2.py')
transitionState('TS_C5H6_CH3-c5-2', './species/TS_C5H6_CH3-c5-2b.py')
transitionState('TS_C5H6_CH3-c5', './species/TS_C5H6_CH3-c5.py')
```

Label ID and *location* of TS files. Note: no collisional information needed.

Reaction cards are needed for each reaction you want to compute the kinetics (one for each TS in your system):

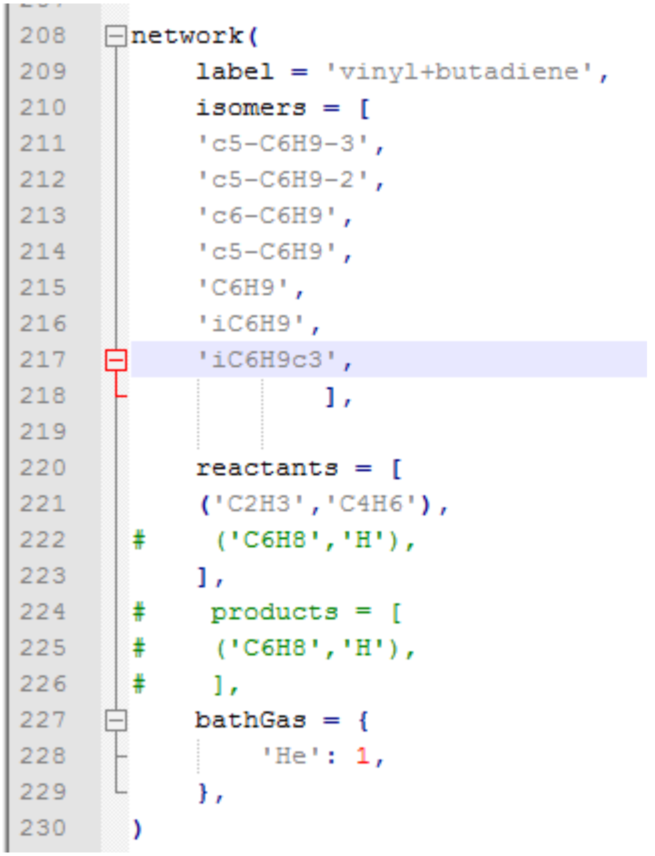


```
reaction(
    label = 'iC6H9c3 = C6H9',
    reactants = ['iC6H9c3'],
    products = ['C6H9'],
    transitionState = 'TSi_beta',
    tunneling='Eckart',
)
reaction(
    label = 'C6H9 = C6H8 + H',
    reactants = ['C6H9'],
    products = ['C6H8','H'],
    transitionState = 'TS_C6H8_H-1',
    tunneling='Eckart',
)
```



```
kinetics('C2H3 + C4H6 = C6H9')
kinetics('C2H3 + C4H6 = iC6H9')
#kinetics('iC6H9 = iC6H9c3')
kinetics('iC6H9c3 = C6H9')
kinetics('C6H9 = C6H8 + H')
kinetics('C6H9 = c6-C6H9')
kinetics('C6H9 = c5-C6H9')
kinetics('c5-C6H9 = c5-C6H8 + H')
kinetics('c6-C6H9 = C6H8-c6-13 + H')
kinetics('c6-C6H9 = C6H8-c6-14 + H')
kinetics('c5-C6H9 = c5-C6H9-2')
kinetics('c5-C6H9-3 = c5-C6H9-2')
kinetics('c5-C6H9-2 = C5H6 + CH3')
```

kinetics('reaction label'): Indicates to CanTherm that you want to compute k_∞ for each of these reactions, which are identified according to labels in the corresponding *reaction* cards



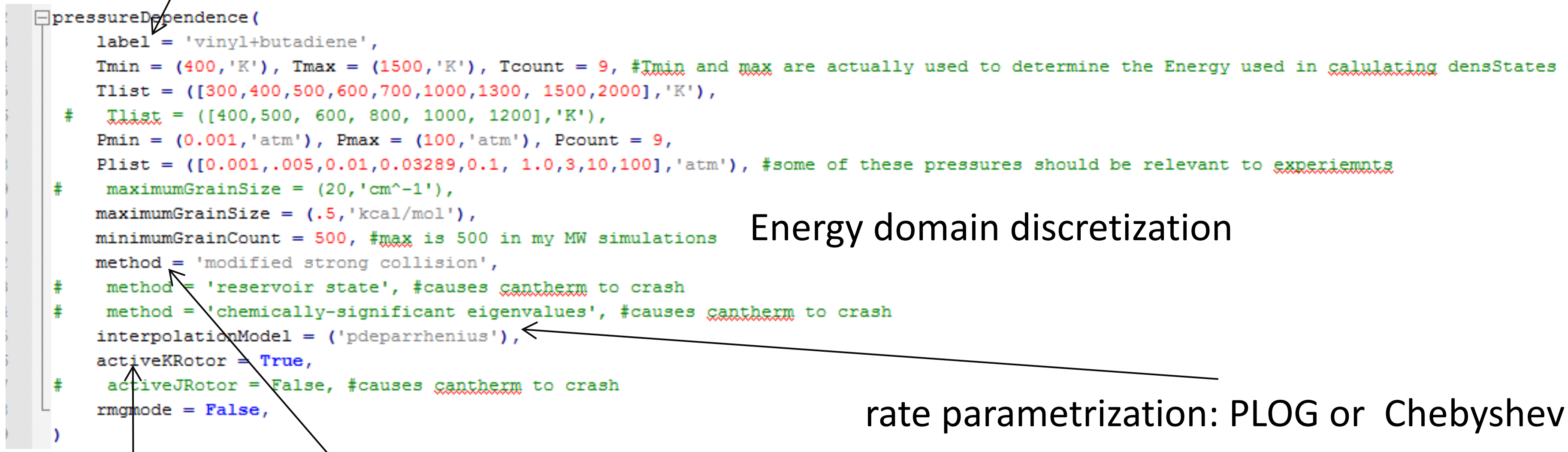
```
network(
    label = 'vinyl+butadiene',
    isomers = [
    'c5-C6H9-3',
    'c5-C6H9-2',
    'c6-C6H9',
    'c5-C6H9',
    'C6H9',
    'iC6H9',
    'iC6H9c3',
            ],

    reactants = [
    ('C2H3','C4H6'),
#    ('C6H8','H'),
    ],
#    products = [
#    ('C6H8','H'),
#    ],
    bathGas = {
        'He': 1,
    },
)
```

For Pdep reactions, this section is necessary and defines the multiple well reaction network. Include all relevant isomers/wells.

The reactant[s] and bath must be included.

Cantherm input file components – pdep

network label



```
pressureDependence(
    label = 'vinyl+butadiene',
    Tmin = (400,'K'), Tmax = (1500,'K'), Tcount = 9, #Tmin and max are actually used to determine the Energy used in calulating densStates
    Tlist = ([300,400,500,600,700,1000,1300, 1500,2000],'K'),
 #   Tlist = ([400,500, 600, 800, 1000, 1200],'K'),
    Pmin = (0.001,'atm'), Pmax = (100,'atm'), Pcount = 9,
    Plist = ([0.001,.005,0.01,0.03289,0.1, 1.0,3,10,100],'atm'), #some of these pressures should be relevant to experiemnts
#    maximumGrainSize = (20,'cm^-1'),
    maximumGrainSize = (.5,'kcal/mol'),
    minimumGrainCount = 500, #max is 500 in my MW simulations
    method = 'modified strong collision',
#    method = 'reservoir state', #causes cantherm to crash
#    method = 'chemically-significant eigenvalues', #causes cantherm to crash
    interpolationModel = ('pdeparrhenius'),
    activeKRotor = True,
#    activeJRotor = False, #causes cantherm to crash
    rmgmode = False,
)
```

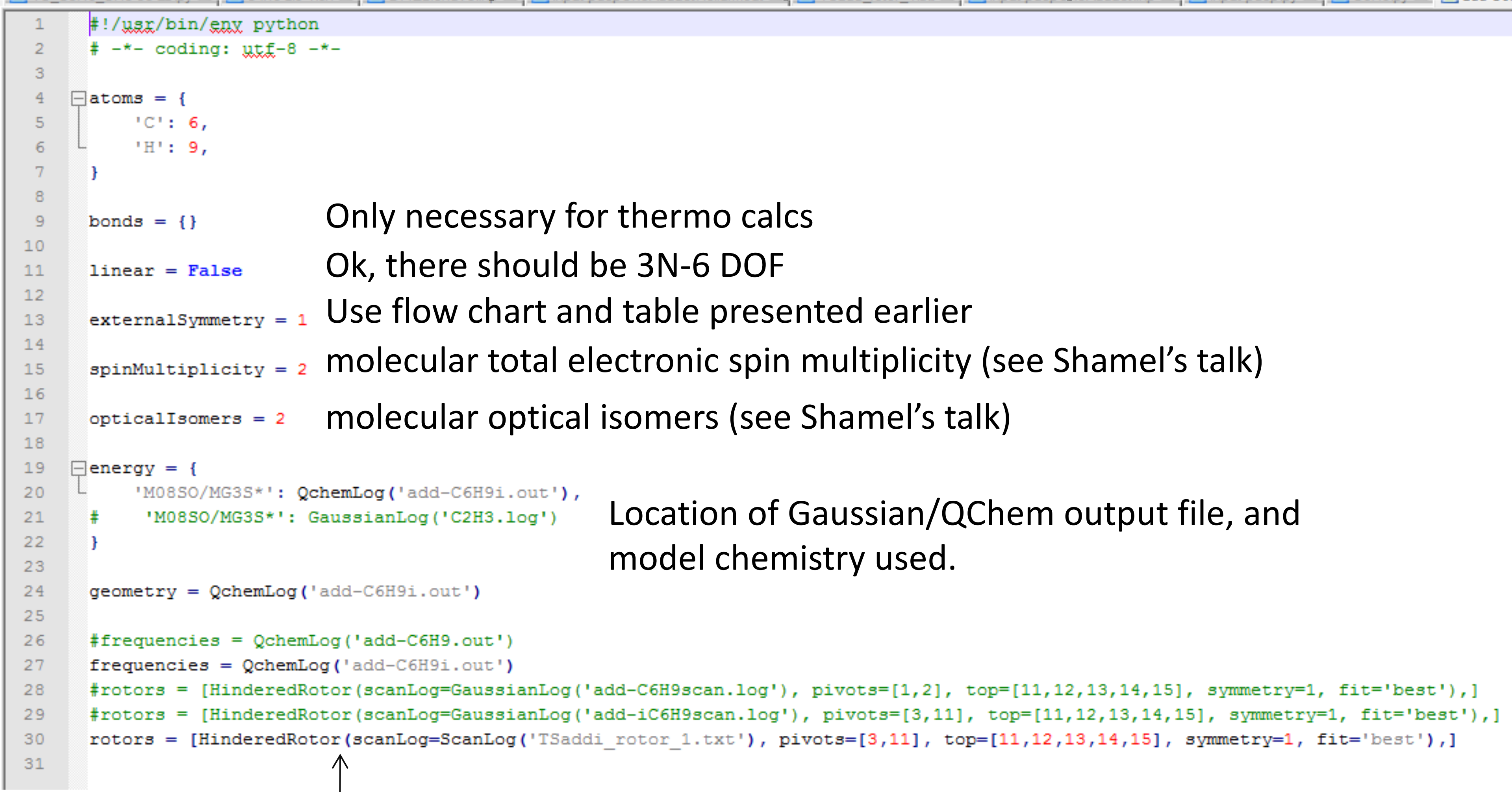
Energy domain discretization

rate parametrization: PLOG or Chebyshev

Master equation solution method

Include External 1D rotor as an active degree of freedom.
Specific to assuming that the molecule is a symmetric top with $I_a \neq I_b \sim I_c$
By treating it as active, it exchanges energy with other molecular degrees of freedom, convoluted into density of states

Cantherm input file components – species files



```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

atoms = {
    'C': 6,
    'H': 9,
}

bonds = {}

linear = False

externalSymmetry = 1

spinMultiplicity = 2

opticalIsomers = 2

energy = {
    'M08SO/MG3S*': QchemLog('add-C6H9i.out'),
#    'M08SO/MG3S*': GaussianLog('C2H3.log')
}

geometry = QchemLog('add-C6H9i.out')

#frequencies = QchemLog('add-C6H9.out')
frequencies = QchemLog('add-C6H9i.out')
#rotors = [HinderedRotor(scanLog=GaussianLog('add-C6H9scan.log'), pivots=[1,2], top=[11,12,13,14,15], symmetry=1, fit='best'),]
#rotors = [HinderedRotor(scanLog=GaussianLog('add-iC6H9scan.log'), pivots=[3,11], top=[11,12,13,14,15], symmetry=1, fit='best'),]
rotors = [HinderedRotor(scanLog=ScanLog('TSaddi_rotor_1.txt'), pivots=[3,11], top=[11,12,13,14,15], symmetry=1, fit='best'),]
```

Only necessary for thermo calcs

Ok, there should be 3N-6 DOF

Use flow chart and table presented earlier

molecular total electronic spin multiplicity (see Shamel's talk)

molecular optical isomers (see Shamel's talk)

Location of Gaussian/QChem output file, and model chemistry used.

1D Hindered Rotor information, to follow

If all your input parameters are correct, and if CanTherm can accept the level of theory you computed your system at:

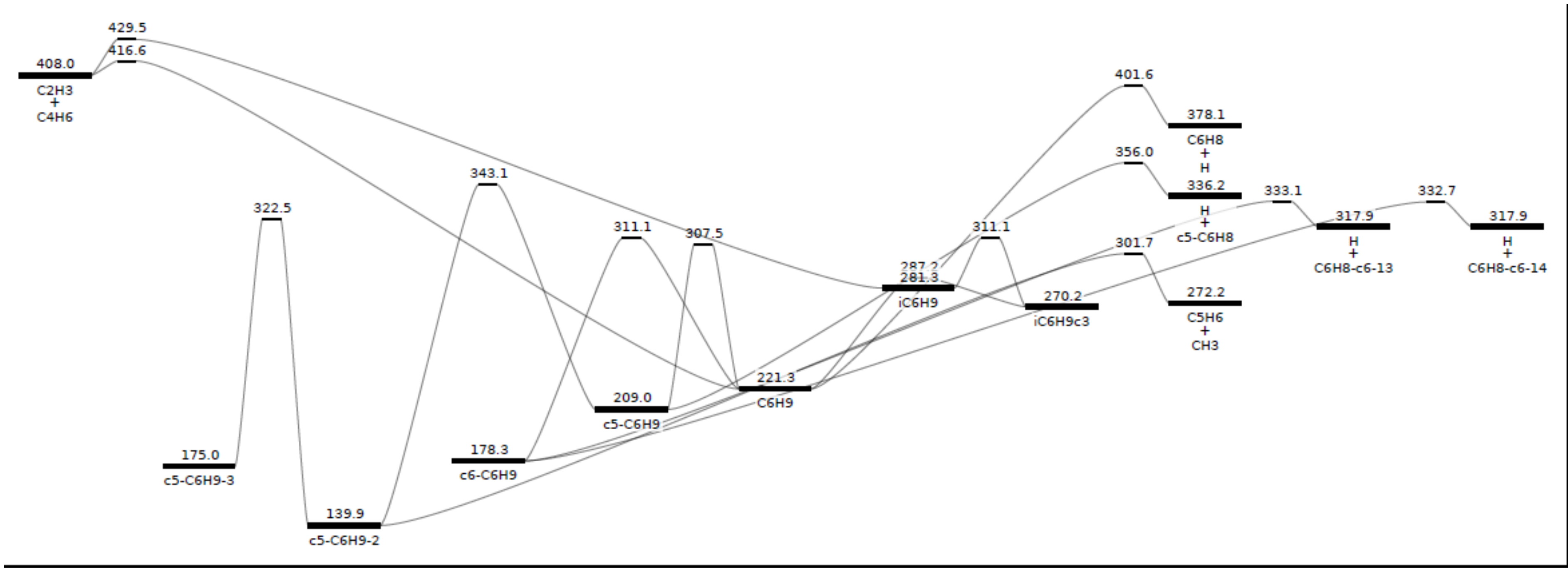
Run Cantherm. For example, at linux command line:

```
python ~edames/RMG-Py/cantherm.py anyFileName.py
```

Look at output files:

- pdf of reaction network
- anyFileName.out
- chem.inp
- pdfs of 1D rotor potentials and .txts of dihedral angle vs potential energy

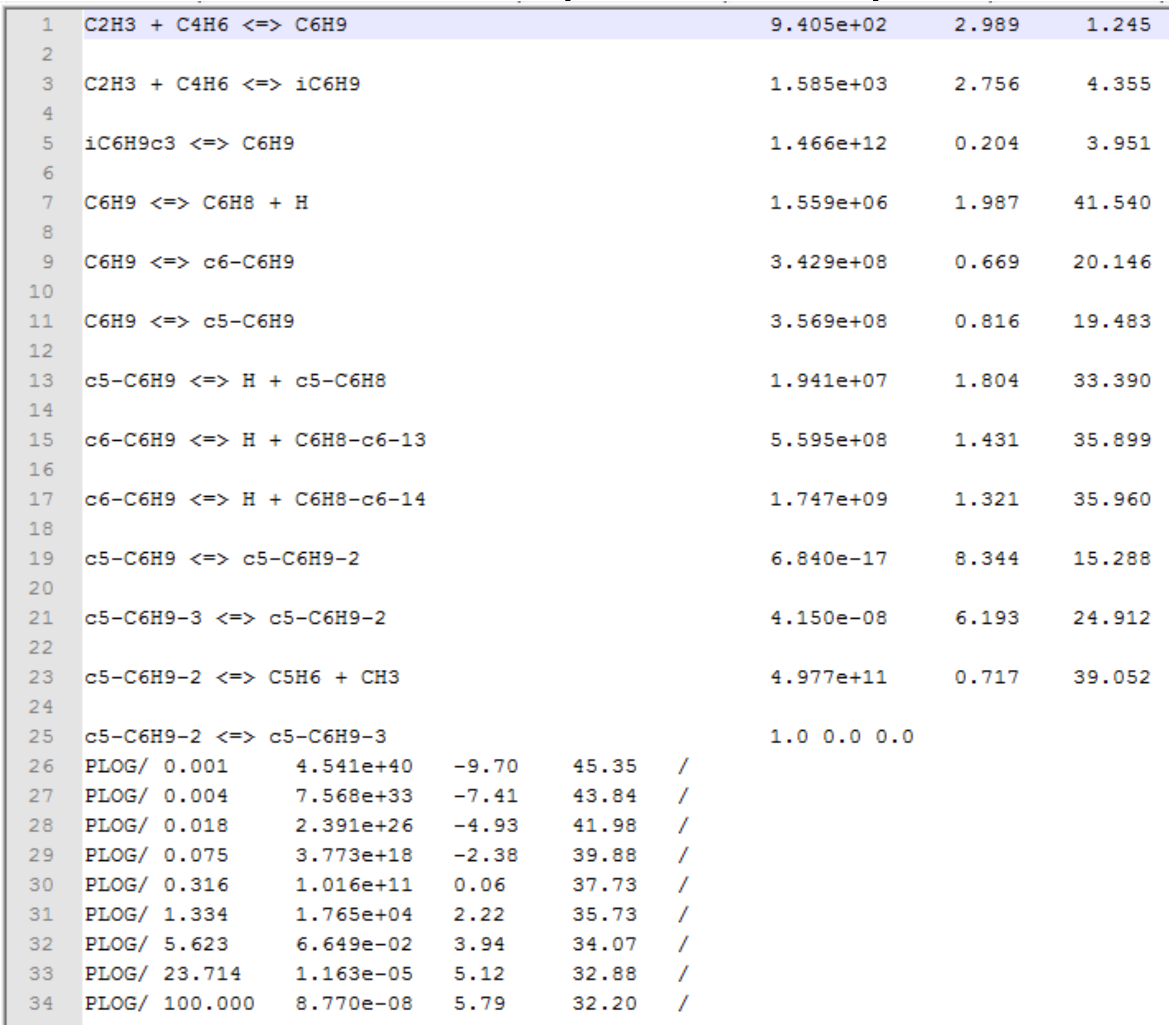
Cantherm generates a pdf of your network, which can serve as a good sanity check



Make sure your network looks good:

- No unreasonably large absolute energy values (default units are kJ/mol)
- All wells are connected as you expect and compare well with your independently created potential energy surface
- All barriers and relative energies look reasonable compared to your independently performed calculations

Cantherm output file components – chem.inp



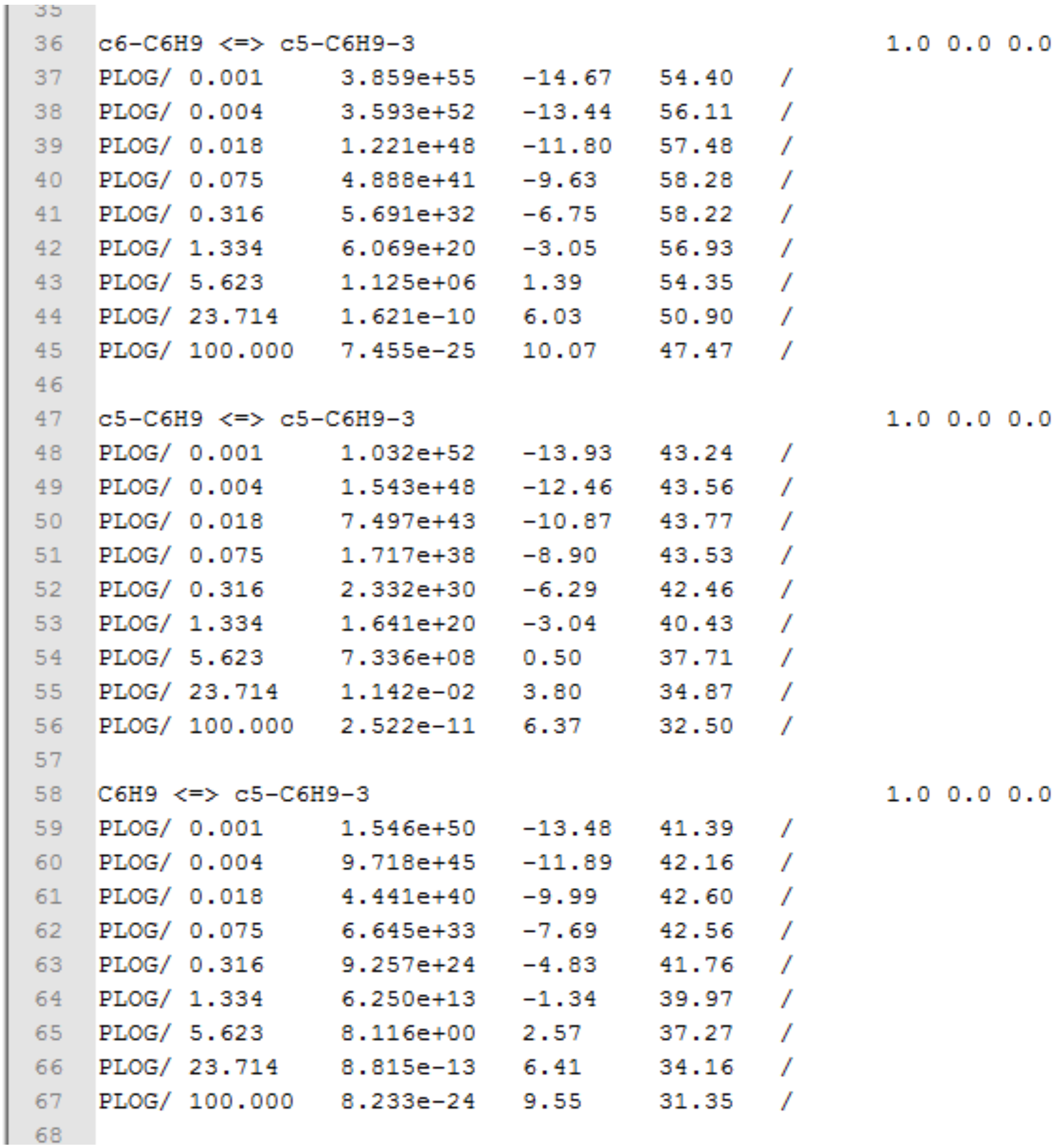
```
C2H3 + C4H6 <=> C6H9                                   9.405e+02     2.989     1.245

C2H3 + C4H6 <=> iC6H9                                  1.585e+03     2.756     4.355

iC6H9c3 <=> C6H9                                       1.466e+12     0.204     3.951

C6H9 <=> C6H8 + H                                      1.559e+06     1.987    41.540

C6H9 <=> c6-C6H9                                       3.429e+08     0.669    20.146

C6H9 <=> c5-C6H9                                       3.569e+08     0.816    19.483

c5-C6H9 <=> H + c5-C6H8                                1.941e+07     1.804    33.390

c6-C6H9 <=> H + C6H8-c6-13                             5.595e+08     1.431    35.899

c6-C6H9 <=> H + C6H8-c6-14                             1.747e+09     1.321    35.960

c5-C6H9 <=> c5-C6H9-2                                  6.840e-17     8.344    15.288

c5-C6H9-3 <=> c5-C6H9-2                                4.150e-08     6.193    24.912

c5-C6H9-2 <=> C5H6 + CH3                               4.977e+11     0.717    39.052

c5-C6H9-2 <=> c5-C6H9-3                                1.0 0.0 0.0
PLOG/ 0.001     4.541e+40   -9.70    45.35   /
PLOG/ 0.004     7.568e+33   -7.41    43.84   /
PLOG/ 0.018     2.391e+26   -4.93    41.98   /
PLOG/ 0.075     3.773e+18   -2.38    39.88   /
PLOG/ 0.316     1.016e+11   0.06     37.73   /
PLOG/ 1.334     1.765e+04   2.22     35.73   /
PLOG/ 5.623     6.649e-02   3.94     34.07   /
PLOG/ 23.714    1.163e-05   5.12     32.88   /
PLOG/ 100.000   8.770e-08   5.79     32.20   /
```

Fitted high-P limit rates requested in *kinetics* cards of input

Cantherm output file components – chem.inp

Pdep rates:

- either PLOG or Chebyshev (see documentation for definitions)
- *always* look at fitting errors in anyFileName.out

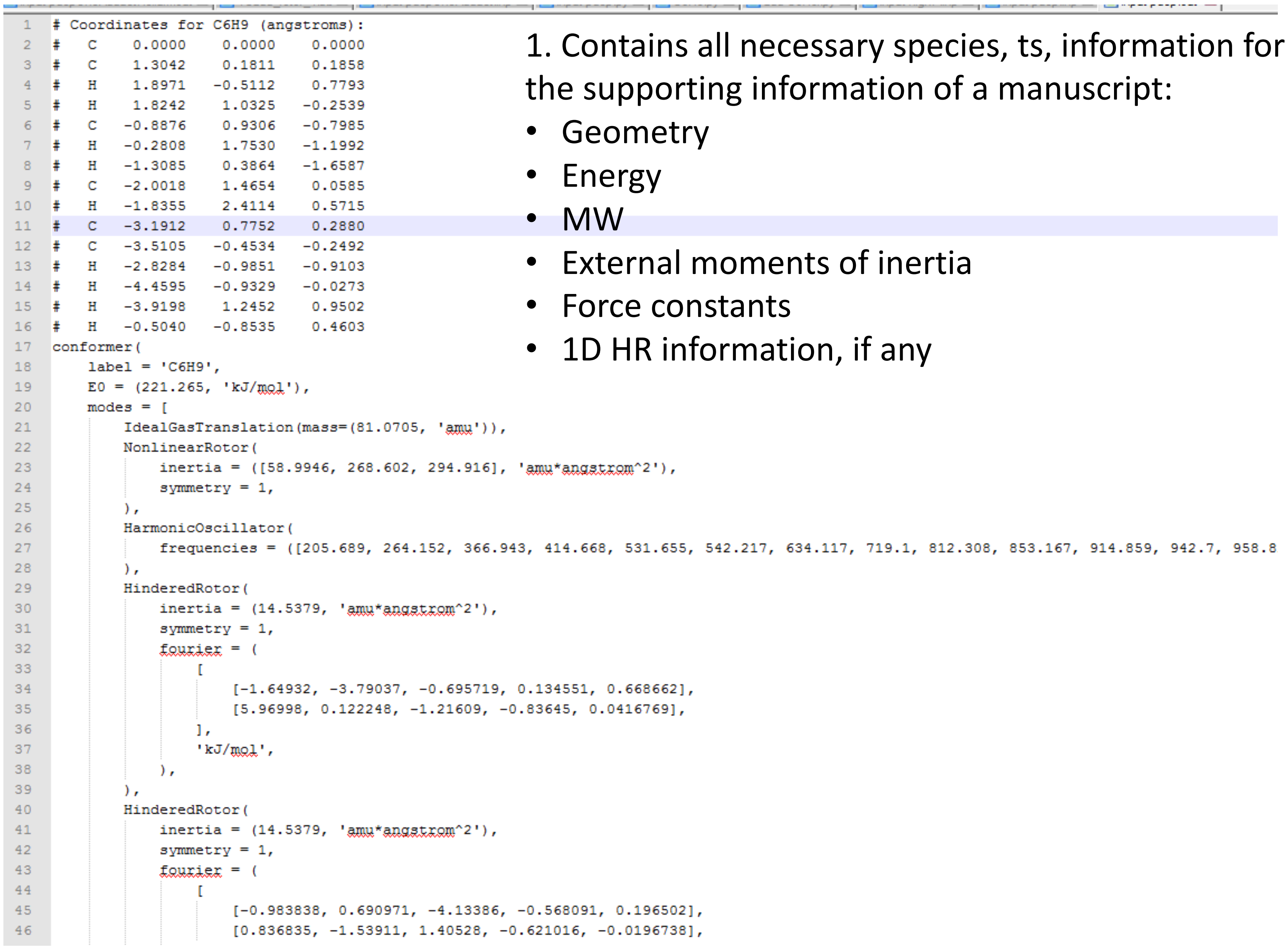


```
c6-C6H9 <=> c5-C6H9-3                                     1.0 0.0 0.0
PLOG/ 0.001      3.859e+55   -14.67   54.40   /
PLOG/ 0.004      3.593e+52   -13.44   56.11   /
PLOG/ 0.018      1.221e+48   -11.80   57.48   /
PLOG/ 0.075      4.888e+41   -9.63    58.28   /
PLOG/ 0.316      5.691e+32   -6.75    58.22   /
PLOG/ 1.334      6.069e+20   -3.05    56.93   /
PLOG/ 5.623      1.125e+06   1.39     54.35   /
PLOG/ 23.714     1.621e-10   6.03     50.90   /
PLOG/ 100.000    7.455e-25   10.07    47.47   /

c5-C6H9 <=> c5-C6H9-3                                     1.0 0.0 0.0
PLOG/ 0.001      1.032e+52   -13.93   43.24   /
PLOG/ 0.004      1.543e+48   -12.46   43.56   /
PLOG/ 0.018      7.497e+43   -10.87   43.77   /
PLOG/ 0.075      1.717e+38   -8.90    43.53   /
PLOG/ 0.316      2.332e+30   -6.29    42.46   /
PLOG/ 1.334      1.641e+20   -3.04    40.43   /
PLOG/ 5.623      7.336e+08   0.50     37.71   /
PLOG/ 23.714     1.142e-02   3.80     34.87   /
PLOG/ 100.000    2.522e-11   6.37     32.50   /

C6H9 <=> c5-C6H9-3                                        1.0 0.0 0.0
PLOG/ 0.001      1.546e+50   -13.48   41.39   /
PLOG/ 0.004      9.718e+45   -11.89   42.16   /
PLOG/ 0.018      4.441e+40   -9.99    42.60   /
PLOG/ 0.075      6.645e+33   -7.69    42.56   /
PLOG/ 0.316      9.257e+24   -4.83    41.76   /
PLOG/ 1.334      6.250e+13   -1.34    39.97   /
PLOG/ 5.623      8.116e+00   2.57     37.27   /
PLOG/ 23.714     8.815e-13   6.41     34.16   /
PLOG/ 100.000    8.233e-24   9.55     31.35   /
```

Cantherm output file components – anyFileName.out

1. Contains all necessary species, ts, information for the supporting information of a manuscript:

- Geometry
- Energy
- MW
- External moments of inertia
- Force constants
- 1D HR information, if any

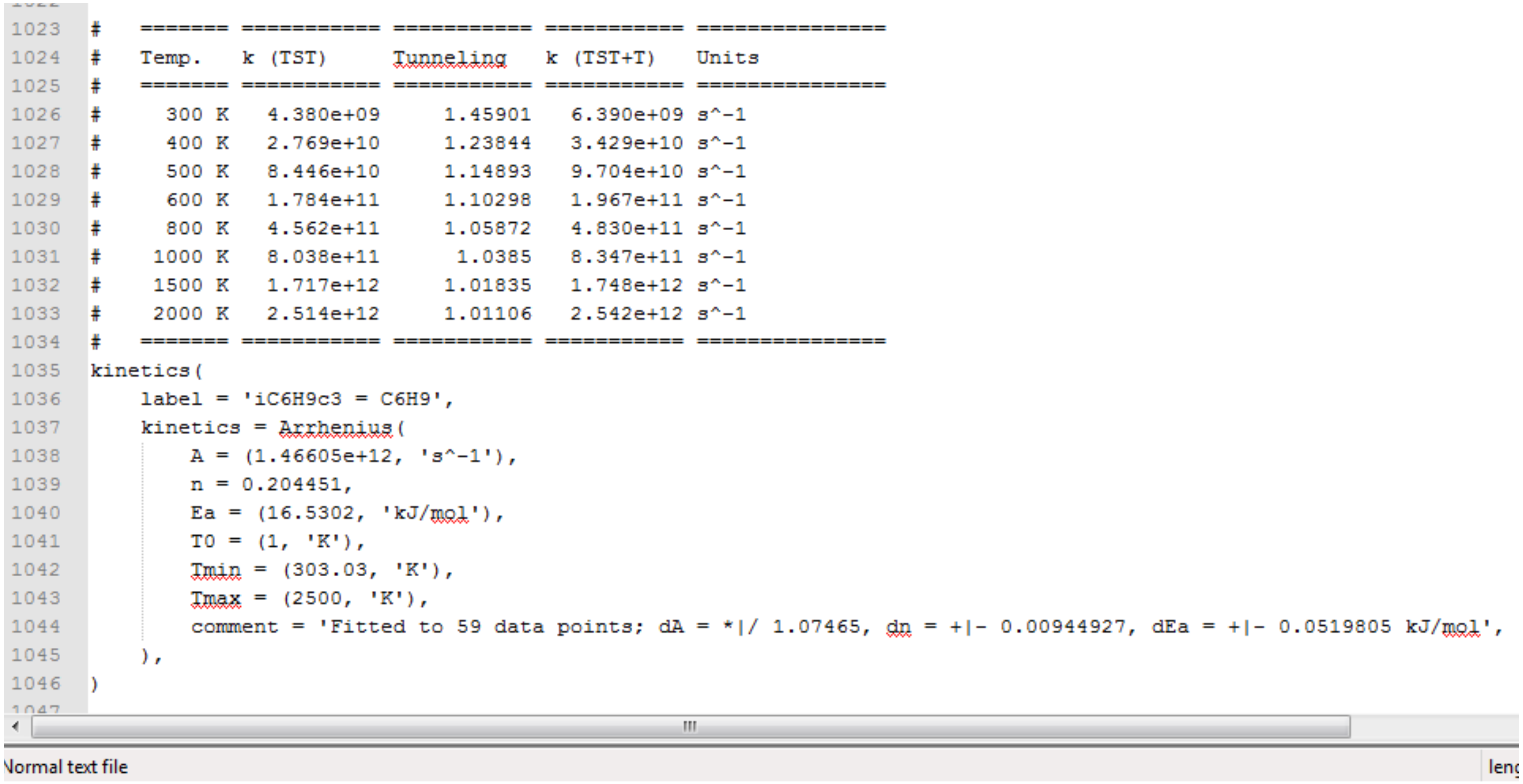


```
# Coordinates for C6H9 (angstroms):
#   C     0.0000    0.0000    0.0000
#   C     1.3042    0.1811    0.1858
#   H     1.8971   -0.5112    0.7793
#   H     1.8242    1.0325   -0.2539
#   C    -0.8876    0.9306   -0.7985
#   H    -0.2808    1.7530   -1.1992
#   H    -1.3085    0.3864   -1.6587
#   C    -2.0018    1.4654    0.0585
#   H    -1.8355    2.4114    0.5715
#   C    -3.1912    0.7752    0.2880
#   C    -3.5105   -0.4534   -0.2492
#   H    -2.8284   -0.9851   -0.9103
#   H    -4.4595   -0.9329   -0.0273
#   H    -3.9198    1.2452    0.9502
#   H    -0.5040   -0.8535    0.4603
conformer(
    label = 'C6H9',
    E0 = (221.265, 'kJ/mol'),
    modes = [
        IdealGasTranslation(mass=(81.0705, 'amu')),
        NonlinearRotor(
            inertia = ([58.9946, 268.602, 294.916], 'amu*angstrom^2'),
            symmetry = 1,
        ),
        HarmonicOscillator(
            frequencies = ([205.689, 264.152, 366.943, 414.668, 531.655, 542.217, 634.117, 719.1, 812.308, 853.167, 914.859, 942.7, 958.8
        ),
        HinderedRotor(
            inertia = (14.5379, 'amu*angstrom^2'),
            symmetry = 1,
            fourier = (
                [
                    [-1.64932, -3.79037, -0.695719, 0.134551, 0.668662],
                    [5.96998, 0.122248, -1.21609, -0.83645, 0.0416769],
                ],
                'kJ/mol',
            ),
        ),
        HinderedRotor(
            inertia = (14.5379, 'amu*angstrom^2'),
            symmetry = 1,
            fourier = (
                [
                    [-0.983838, 0.690971, -4.13386, -0.568091, 0.196502],
                    [0.836835, -1.53911, 1.40528, -0.621016, -0.0196738],
```

Cantherm output file components – anyFileName.out

2. Tabulated k_∞ for all reactions specified in 'kinetics' cards of input file:

- 3-parameter Arrhenius fits
- fitting errors
- units
- tunneling correction factors



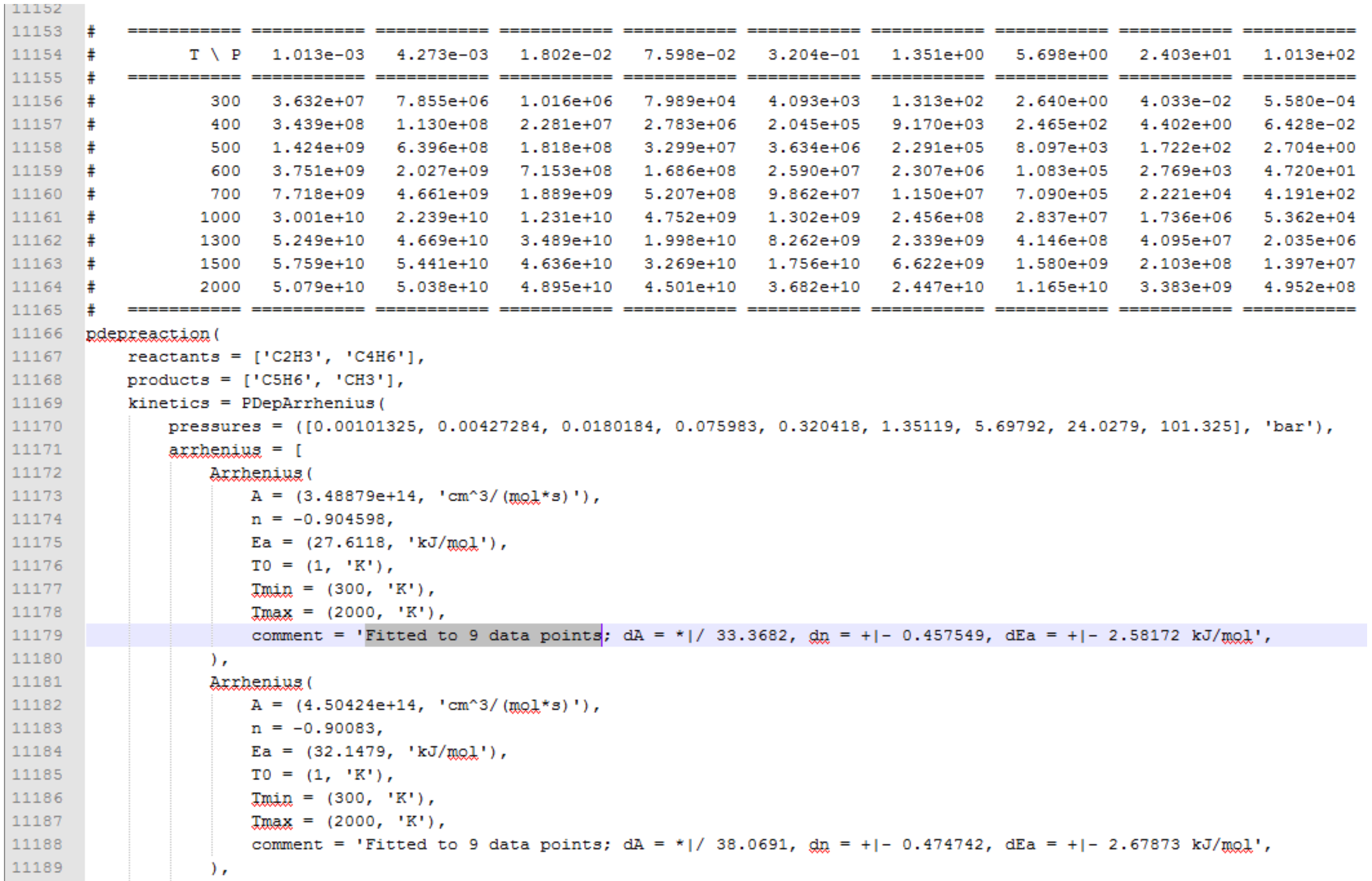
```
#   ======= =========== =========== =========== ===============
#   Temp.   k (TST)     Tunneling   k (TST+T)   Units
#   ======= =========== =========== =========== ===============
#     300 K   4.380e+09     1.45901   6.390e+09 s^-1
#     400 K   2.769e+10     1.23844   3.429e+10 s^-1
#     500 K   8.446e+10     1.14893   9.704e+10 s^-1
#     600 K   1.784e+11     1.10298   1.967e+11 s^-1
#     800 K   4.562e+11     1.05872   4.830e+11 s^-1
#    1000 K   8.038e+11      1.0385   8.347e+11 s^-1
#    1500 K   1.717e+12     1.01835   1.748e+12 s^-1
#    2000 K   2.514e+12     1.01106   2.542e+12 s^-1
#   ======= =========== =========== =========== ===============
kinetics(
    label = 'iC6H9c3 = C6H9',
    kinetics = Arrhenius(
        A = (1.46605e+12, 's^-1'),
        n = 0.204451,
        Ea = (16.5302, 'kJ/mol'),
        T0 = (1, 'K'),
        Tmin = (303.03, 'K'),
        Tmax = (2500, 'K'),
        comment = 'Fitted to 59 data points; dA = *|/ 1.07465, dn = +|- 0.00944927, dEa = +|- 0.0519805 kJ/mol',
    ),
)
```

Normal text file

Cantherm output file components – anyFileName.out

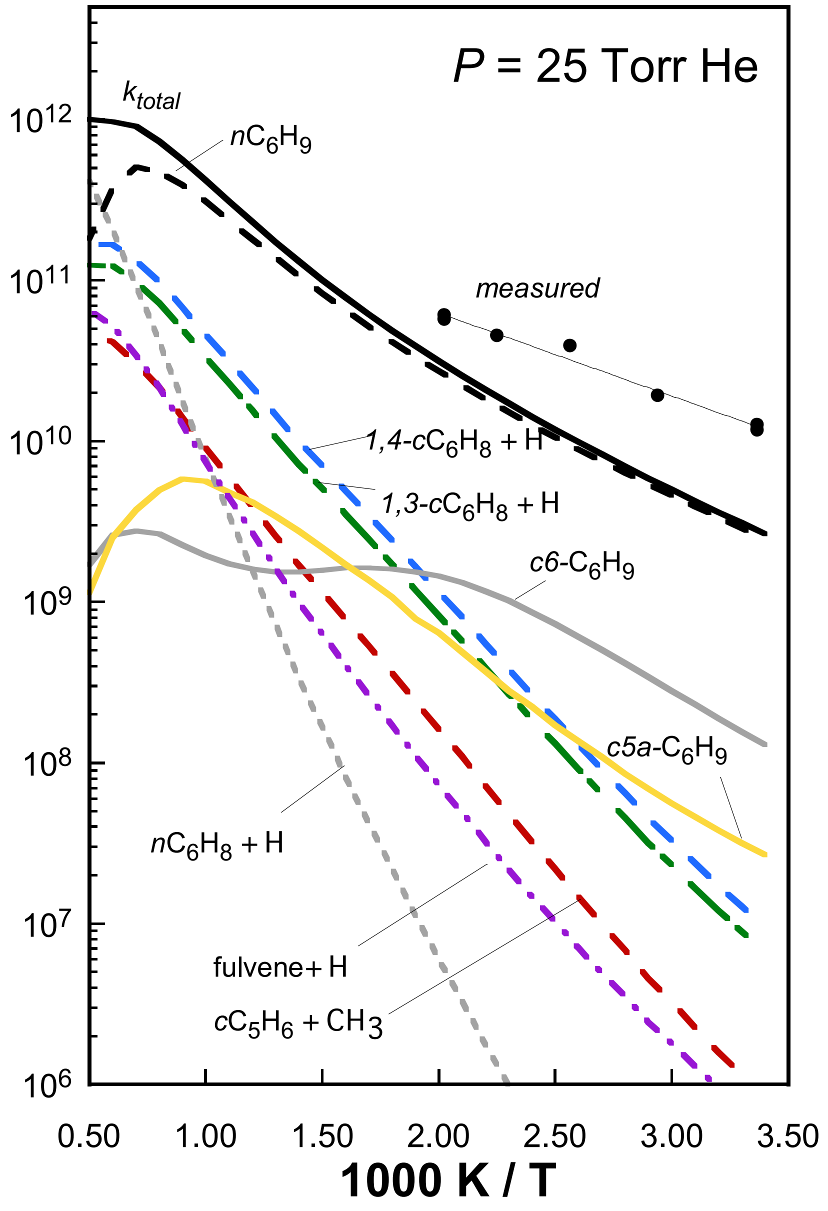
3. Tabulated *k(T,P)* for all possible direct and well-skipping reactions in your reaction network:

- tabulated values are raw ME soln. output
- PLOG/Chebyshev fitting errors
- units
- fitted to same no. of points as temperatures desired (increase for decreased fitting error)

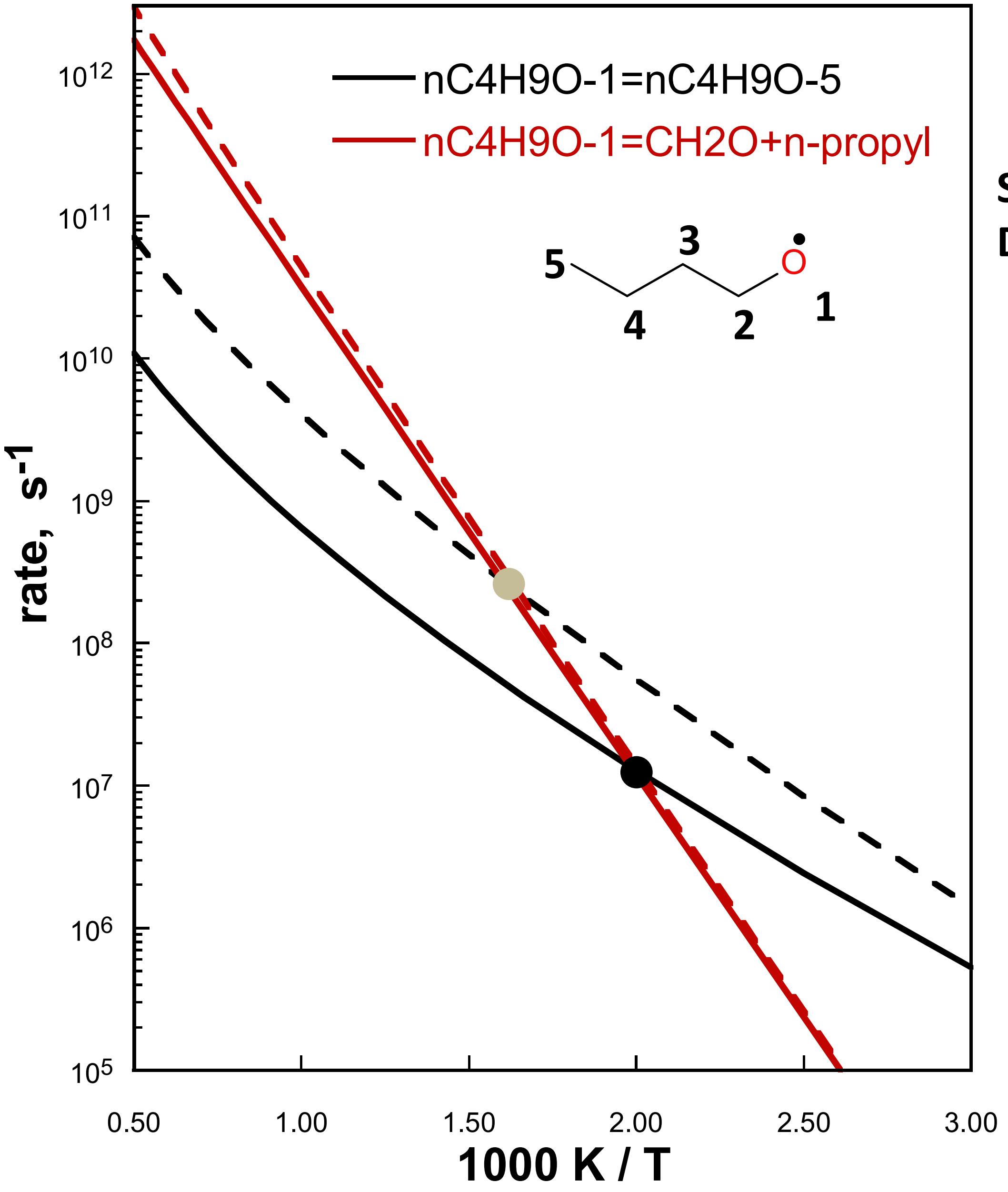


```
#   =========== =========== =========== =========== =========== =========== =========== =========== =========== ===========
#         T \ P   1.013e-03   4.273e-03   1.802e-02   7.598e-02   3.204e-01   1.351e+00   5.698e+00   2.403e+01   1.013e+02
#   =========== =========== =========== =========== =========== =========== =========== =========== =========== ===========
#           300   3.632e+07   7.855e+06   1.016e+06   7.989e+04   4.093e+03   1.313e+02   2.640e+00   4.033e-02   5.580e-04
#           400   3.439e+08   1.130e+08   2.281e+07   2.783e+06   2.045e+05   9.170e+03   2.465e+02   4.402e+00   6.428e-02
#           500   1.424e+09   6.396e+08   1.818e+08   3.299e+07   3.634e+06   2.291e+05   8.097e+03   1.722e+02   2.704e+00
#           600   3.751e+09   2.027e+09   7.153e+08   1.686e+08   2.590e+07   2.307e+06   1.083e+05   2.769e+03   4.720e+01
#           700   7.718e+09   4.661e+09   1.889e+09   5.207e+08   9.862e+07   1.150e+07   7.090e+05   2.221e+04   4.191e+02
#          1000   3.001e+10   2.239e+10   1.231e+10   4.752e+09   1.302e+09   2.456e+08   2.837e+07   1.736e+06   5.362e+04
#          1300   5.249e+10   4.669e+10   3.489e+10   1.998e+10   8.262e+09   2.339e+09   4.146e+08   4.095e+07   2.035e+06
#          1500   5.759e+10   5.441e+10   4.636e+10   3.269e+10   1.756e+10   6.622e+09   1.580e+09   2.103e+08   1.397e+07
#          2000   5.079e+10   5.038e+10   4.895e+10   4.501e+10   3.682e+10   2.447e+10   1.165e+10   3.383e+09   4.952e+08
#   =========== =========== =========== =========== =========== =========== =========== =========== =========== ===========
pdepreaction(
    reactants = ['C2H3', 'C4H6'],
    products = ['C5H6', 'CH3'],
    kinetics = PDepArrhenius(
        pressures = ([0.00101325, 0.00427284, 0.0180184, 0.075983, 0.320418, 1.35119, 5.69792, 24.0279, 101.325], 'bar'),
        arrhenius = [
            Arrhenius(
                A = (3.48879e+14, 'cm^3/(mol*s)'),
                n = -0.904598,
                Ea = (27.6118, 'kJ/mol'),
                T0 = (1, 'K'),
                Tmin = (300, 'K'),
                Tmax = (2000, 'K'),
                comment = 'Fitted to 9 data points; dA = *|/ 33.3682, dn = +|- 0.457549, dEa = +|- 2.58172 kJ/mol',
            ),
            Arrhenius(
                A = (4.50424e+14, 'cm^3/(mol*s)'),
                n = -0.90083,
                Ea = (32.1479, 'kJ/mol'),
                T0 = (1, 'K'),
                Tmin = (300, 'K'),
                Tmax = (2000, 'K'),
                comment = 'Fitted to 9 data points; dA = *|/ 38.0691, dn = +|- 0.474742, dEa = +|- 2.67873 kJ/mol',
            ),
```

Cantherm output file components – overall plotted rates



Consideration of hindered rotors important when they are tied up in transition states

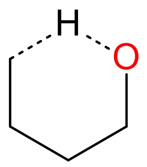


n-butoxy decomp./isom. comparisons(k_∞): HR vs RRHO

Solid lines: Hindered rotor treatment

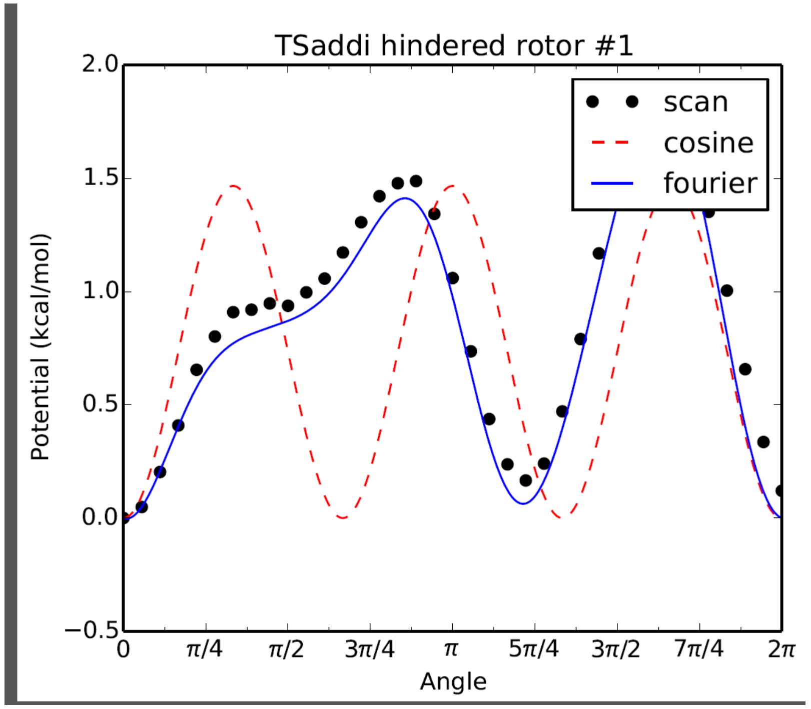
Dashed lines: RRHO treatment

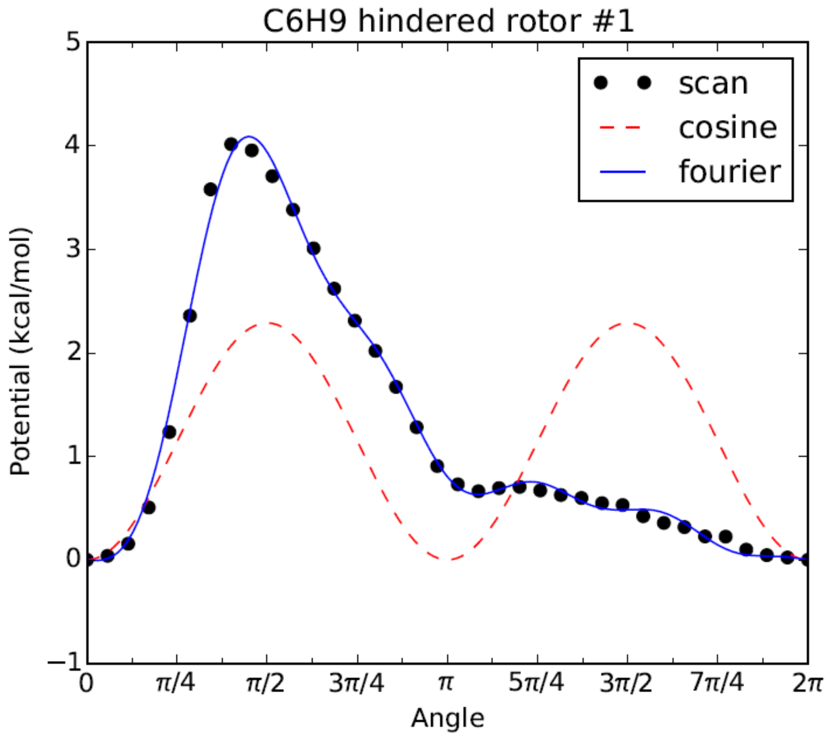
What the 1,5 H-shift transition state 'looks' like:



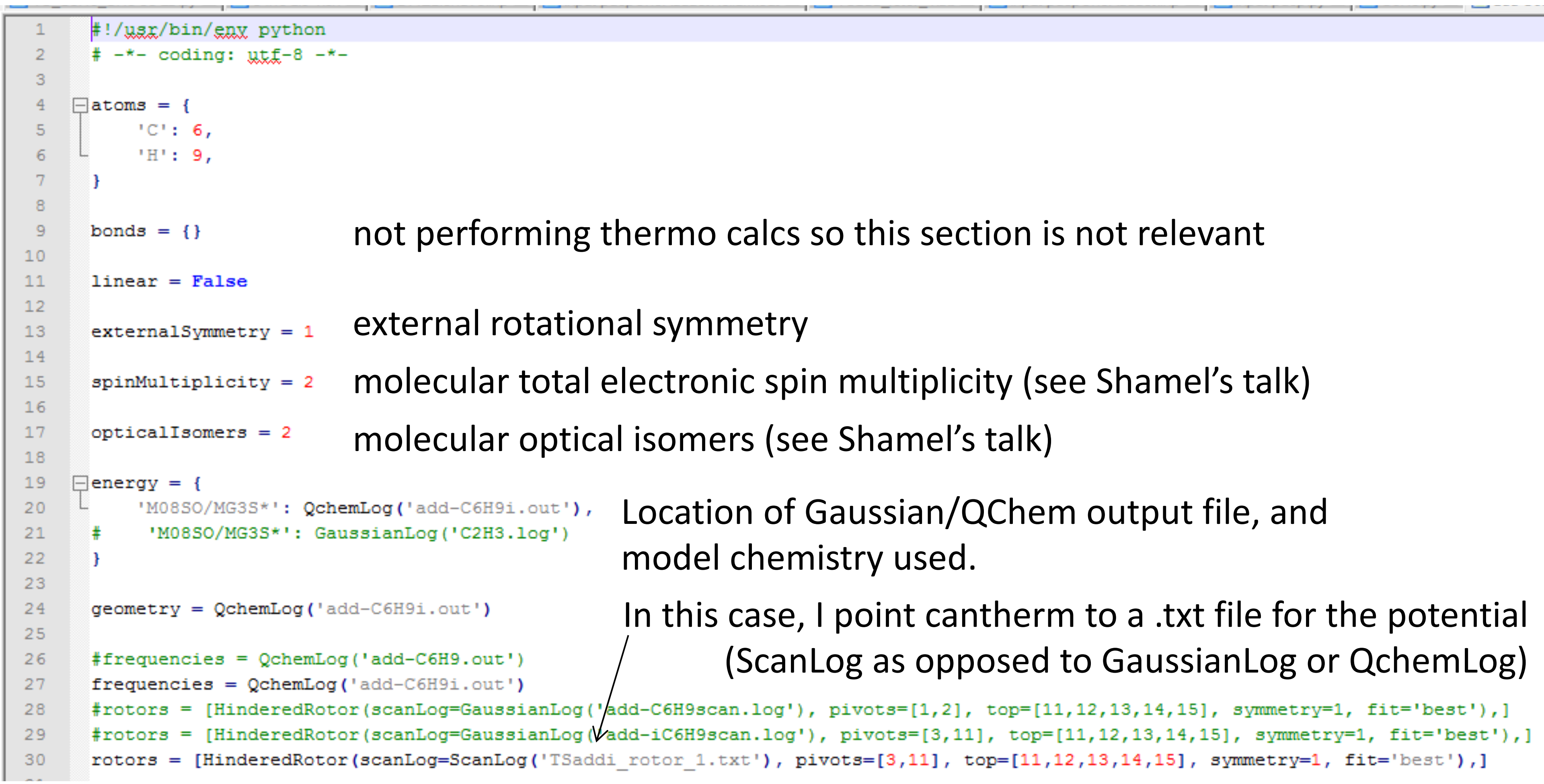
Hindered rotors

- Typically can be identified by a vibrational frequencies less than 150 cm^{-1}
- Know there are many ways to account for 1-D internal rotors. Cantherm projects out the degree of freedom corresponding to the rotor from the force constant matrix – a good compromise between accuracy and speed.
- 1-D potential scans typically performed in Gaussian or QChem
- Care must be taken when preparing cantherm input files
- If $V(\theta=0°) \neq 0$, fourier fit will be inaccurate, $\therefore$ user may 'shift' potential to fix this, rather than recompute scan from different starting geometry





Hindered rotors



```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

atoms = {
    'C': 6,
    'H': 9,
}

bonds = {}

linear = False

externalSymmetry = 1

spinMultiplicity = 2

opticalIsomers = 2

energy = {
    'M08SO/MG3S*': QchemLog('add-C6H9i.out'),
#     'M08SO/MG3S*': GaussianLog('C2H3.log')
}

geometry = QchemLog('add-C6H9i.out')

#frequencies = QchemLog('add-C6H9.out')
frequencies = QchemLog('add-C6H9i.out')
#rotors = [HinderedRotor(scanLog=GaussianLog('add-C6H9scan.log'), pivots=[1,2], top=[11,12,13,14,15], symmetry=1, fit='best'),]
#rotors = [HinderedRotor(scanLog=GaussianLog('add-iC6H9scan.log'), pivots=[3,11], top=[11,12,13,14,15], symmetry=1, fit='best'),]
rotors = [HinderedRotor(scanLog=ScanLog('TSaddi_rotor_1.txt'), pivots=[3,11], top=[11,12,13,14,15], symmetry=1, fit='best'),]
```

not performing thermo calcs so this section is not relevant

external rotational symmetry

molecular total electronic spin multiplicity (see Shamel's talk)

molecular optical isomers (see Shamel's talk)

Location of Gaussian/QChem output file, and model chemistry used.

In this case, I point cantherm to a .txt file for the potential (ScanLog as opposed to GaussianLog or QchemLog)

pivots: two atoms defining axis of rotation
top: atoms containing in one of two portions of rotating moiety
symmetry: 3 (•CH3), 2 (•CH2), 1 (potato)
fit: typically, use 'best'
Note: atom indices should correspond to those in the geometry file read in by cantherm

Recipe for Reliable Rate Theory Calculations

1. Define the reaction network and explore pathways – this can be done using RMG (e.g., via generate reactions); perform a literature search
2. Know what you want to calculate (i.e., relevant T, P) and what you are doing.
3. Conduct quantum chemistry calculations (Gaussian, Qchem, Molpro for CC) at a desired/appropriate level of theory
4. Confirm that your geometries have been optimized properly
   - look at each structure and ask yourself if the energy is at a minimum
   - does each saddle point (TS) have one and only one imaginary frequency?
   - visual inspection via a molecule editor (there are many: GaussView, Avagadro, etc. See http://en.wikipedia.org/wiki/Molecule_editor) Note: avagadro is nice because it can perform isomer searches for you.
5. [Very carefully] prepare your CanTherm input files, triple check everything
6. Run CanTherm.
7. Inspect output pdfs: network, 1D HRs
8. Before you use the parametrized rate coefficients in kinetic mechanisms, make sure the fitting errors are acceptable to you, or else consider other options (increase *nTemps*, use raw output, other fitting methods)

Questions?