Reaction Mechanism Generator: Automatic construction of chemical kinetic mechanisms☆

Connie W. Gao [a], Joshua W. Allen [a], William H. Green [a], Richard H. West [b],*

[a] Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, United States
[b] Department of Chemical Engineering, Northeastern University, Boston, MA 02115, United States

ARTICLE INFO

ABSTRACT

Reaction Mechanism Generator (RMG) constructs kinetic models composed of elementary chemical reaction steps using a general understanding of how molecules react. Species thermochemistry is estimated through Benson group additivity and reaction rate coefficients are estimated using a database of known rate rules and reaction templates. At its core, RMG relies on two fundamental data structures: graphs and trees. Graphs are used to represent chemical structures, and trees are used to represent thermodynamic and kinetic data. Models are generated using a rate-based algorithm which excludes species from the model based on reaction fluxes. RMG can generate reaction mechanisms for species involving carbon, hydrogen, oxygen, sulfur, and nitrogen. It also has capabilities for estimating transport and solvation properties, and it automatically computes pressure-dependent rate coefficients and identifies chemically-activated reaction paths. RMG is an object-oriented program written in Python, which provides a stable, robust programming architecture for developing an extensible and modular code base with a large suite of unit tests. Computationally intensive functions are cythonized for speed improvements.

Program summary

---

1. Introduction

Kinetic models are relevant to many chemical processes, including combustion, pyrolysis, and atmospheric oxidation [1]. These processes involve complex free-radical reactions between hundreds of reaction intermediates, coupled with thermodynamic and heat and mass transfer effects. In the past, models were often greatly simplified due to lack of computing power as well as poor understanding of underlying chemistry. Today, numerical solvers and computational chemistry have advanced to the point where detailed kinetic models can now be constructed and applied to complex systems.

Some detailed kinetic models are constructed by hand, through carefully keeping track of all species and reactions and incorporating relevant chemistry. This process is often tedious and error-prone, requiring expert and up-to-date understanding of chemistry. However, the challenges associated with hand-constructed models are the very things that are easily handled by computers. This insight has spawned several automatic reaction mechanism generation codes, some proprietary and some open-source, including MAMOX, NetGen, REACTION, and EXGAS. Broadbelt and Pfaendtner [2] provide an introduction to the general concepts and terminology of kinetic model generation, and several recent reviews describe the commonalities, differences, and histories of these software projects [1,3–5].

The open-source software package RMG (**R**eaction **M**echanism **G**enerator) was developed in the Green Group at MIT to help researchers model physical processes through automatic mechanism generation. All $60,000+$ lines of Python code are open-source and hosted on Github (https://github.com/ReactionMechanism Generator/RMG-Py). RMG was originally developed in Java by Jing Song [6] in 2004, following approaches pioneered by NetGen and the ExxonMobil Mechanism Generator (XMG) in the 1990s [7,8]. An object-oriented programming style was used prioritizing flexibility and extensibility of chemical rules and code re-usability. Over the years, several detailed kinetic models generated by RMG have been published in literature, including models for butanol [9], ketone biofuels [10], JP-10 jet fuel [11], and neopentane [12]. The source code for the Java version of RMG can be found both on Github (https://github.com/ReactionMechanismGenerator/RMG-Java) and Sourceforge (http://rmg.sourceforge.net/), with over 7000 downloads from Sourceforge alone over the last two years. In 2008, Joshua Allen and Richard West began writing a Python version of RMG, known as RMG-Py [13]. This was motivated by improved code readability, better error handling, and broader access to a variety of existing chem informatics libraries. This paper presents the features and usage of the current Python version of RMG.

2. Overview of RMG

RMG is an automatic reaction mechanism generator which uses known chemistry knowledge stored in a database along with parameter estimation methods to generate detailed chemical kinetic mechanisms. These mechanisms can be used as inputs to third party reactor software (e.g. CHEMKIN, Cantera, ANSYS Fluent) to simulate predictions for macrovariables of interest such as product composition, ignition behavior, or flame speed.

The four principal capabilities required for any automatic reaction mechanism generation code are [14]: (1) a way to uniquely and unambiguously represent chemical species, (2) a method to determine what reactions can occur between species, (3) a means to estimate the kinetic and thermodynamic parameters, and (4) a metric by which to include or exclude species and reactions in the model.

RMG uses a functional group based methodology to work with species and reactions. In this approach, reaction families are defined by templates that manipulate matching functional groups to convert molecules from reactants to products. Chemical graph theory is used to represent molecules and functional group substructures, with vertices representing atoms and edges representing bonds. This allows for graph isomorphism comparisons: to identify functional groups when estimating parameters, and to check the identity of species against one another. Thermodynamic parameters are estimated for chemical structures using the Benson group contribution method [15,16] or on-the-fly quantum chemistry calculations [17]. The species and reactions included in the final model are chosen by expanding the model using the rate-based algorithm of Susnow et al. [18].

2.1. Species and functional group representation

In RMG, molecules are described using "adjacency lists", a graph representation of the atoms and bonds that connect them. A set of molecule objects which are resonance isomers form a single species. This species contains its own thermochemical (i.e. enthalpy, entropy, and heat capacities) and statistical mechanical (i.e. frequencies and energies) information. The adjacency list for a methyl radical CH_3 is depicted in Fig. 1.

The syntax of a molecular adjacency list can be described as follows: the first column indicates the atom index, the second column indicates the atomic element, the 3rd column indicates the number of unpaired electrons associated each atom and is preceded by the lowercase letter u representing "unpaired", the 4th column indicates the number of lone pairs associated with each atom and is preceded by the lowercase letter p representing "pairs", and the 5th column indicates the formal charge on the atom preceded by the lowercase letter c representing "charge". The values in brackets indicate the presence of a bond, with the first value within a bracket indicating the atom index of the atom to which the current atom is bonded, and the second value indicating whether the bond order is single S, double D, triple T, or benzene B. Finally, the molecule has an overall spin multiplicity defined above the adjacency list. In the adjacency list depicted in Fig. 1, the carbon
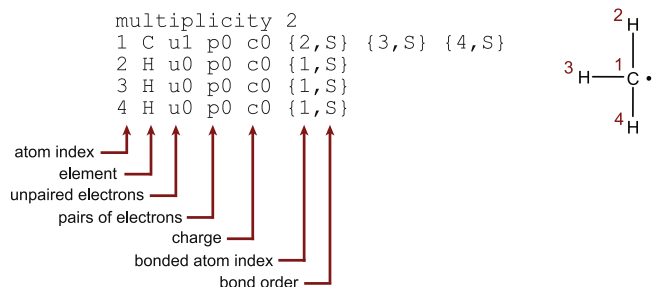
```
multiplicity 2
1 C u1 p0 c0 {2,S} {3,S} {4,S}
2 H u0 p0 c0 {1,S}
3 H u0 p0 c0 {1,S}
4 H u0 p0 c0 {1,S}
```

atom index
element
unpaired electrons
pairs of electrons
charge
bonded atom index
bond order

Fig. 1. Adjacency list (left) and graph (right) of a methyl radical.

Group

```
multiplicity [1,2,3]
1 R!H      ux      {2,S} {3,D}
2 H        u0      {1,S}
3 [Cd,Od]  u[0,1]  {1,D}
```
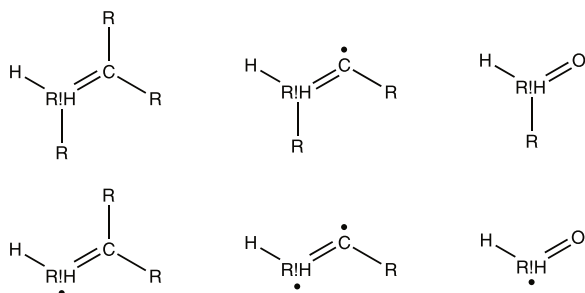
Possible Molecular Structures

Fig. 2. Adjacency list for a functional group (top) and its possible molecular structures (bottom).

atom has a single unpaired electron and 3 single bonds to hydrogen atoms, forming a methyl radical.

Similar to molecular adjacency lists, functional groups can also be described by adjacency lists, but group atom types are used in the adjacency list instead of atomic elements. These atom types can describe a more general set of elements and can sometimes provide additional local bond structure requirements. The use of atom types accelerates graph isomorphism, or equivalency, checks and helps define both broad and highly specific functional groups. The list of group atom types used in RMG are defined in Table 1. Shown in Fig. 2 is an example of a group adjacency list, utilizing the R!H, Cd, and Od atomtypes described in Table 1.

Note that in a group adjacency list, multiplicity, bonds, atom types, and even unpaired electrons can be a set of values. In order to distinguish from the notation used for bonds, square brackets are used to group a set of values. In a group, only the number of unpaired electrons and bond information are required. Values that are unspecified, such as the number of lone pairs and charges on each atom in the adjacency list in Fig. 2, are assumed to be wildcards. The notation x can also be used to represent a wildcard.

By using isomorphism checks between reacting molecules and group definitions in this form, RMG can quickly identify functional group-specific reaction sites. The VF2 algorithm [19] for graph and subgraph isomorphisms is currently implemented in RMG.

2.2. Thermodynamic parameter estimation

Benson-style group additivity [15,16] is used to estimate thermochemical parameters, including enthalpy ΔH_f°, entropy S°, and heat capacities C_p. For free radicals, we use the hydrogen bond increment (HBI) method of Lay et al. [20]. RMG uses hierarchical trees in its database for organizing functional group data in order to improve the speed of identifying group contributions. Trees are organized by placing general functional groups as top nodes, then creating more specific functional groups as children. Identifying the group contribution requires traversing down the tree from top to bottom to match the specific functional group.

The algorithm for estimating the thermodynamic parameters for a species is shown in Fig. 3. First, resonance isomers of the species are generated, including aromatic forms of the species. Then, the thermodynamic parameters for each individual isomer are calculated, first by checking whether the isomer is a free radical species in which HBI corrections are needed. After this step, group contributions to the enthalpy, entropy, and heat capacities are applied to the saturated compound. Then, symmetry algorithms are used to apply a total symmetry number σ correction to the entropy of formation:

$$S^\circ = S_{GA}^\circ - R \ln \sigma \tag{1}$$

where S° is the standard corrected entropy of formation at 298 K, S_{GA}° is the standard entropy at 298 K calculated by the group additivity method, and R is the gas constant. Finally, cyclic and polycyclic ring corrections and gauche corrections are made to the thermodynamic parameters. Once the algorithm finishes iterating through all the isomers, RMG chooses the thermochemistry of the isomer with the most stable enthalpy to represent the thermochemistry for the overall species.

2.2.1. On-the-fly quantum mechanics for cyclic species

Benson group additivity is known to poorly estimate the enthalpy and entropy of cyclic and fused cyclic compounds due to the lack of available ring strain corrections. RMG includes a **Q**uantum **M**echanics **T**hermodynamic **P**roperty (QMTP) interface [17], shown in Fig. 4, which allows it to perform on-the-fly quantum calculations to determine thermodynamic parameters for cyclic and polycyclic species. This interface uses three-dimensional molecular structures in force field or quantum mechanical calculations to obtain thermodynamic parameter estimates. First, RMG sends molecular connectivities derived from its internal molecular graph representations to RDKit [21], which converts them to 3D coordinates using a distance geometry algorithm. Then, an input file containing the 3D molecular structure is sent to an external quantum mechanics program such as MOPAC or Gaussian. RMG derives the thermodynamic properties from parsing the relevant frequencies and energies from the output files. Currently, RMG supports several semi-empirical methods, including PM3, PM6, and PM7.

2.3. Kinetic parameter estimation

RMG generates elementary reactions from chemical species using an extensible set of 45 reaction families, shown in Table 2. A reaction family consists of a template that describes the reactive sites, and a reaction recipe which dictates how the bond connectivity changes when the reaction proceeds to products. Associated with each reaction family is a hierarchical tree of rate estimation rules, assigning kinetics between reaction sites according to their closest-matching functional groups. The rate estimation trees can be modified and extended without editing or

Table 1
Atom types used in RMG group definitions.

Atom type	Description
General atom types	
R	any atom with any local bond structure
R!H	any non-hydrogen atom with any local bond structure
Val4	any atom containing 4 valence electrons with any local bond structure
Val5	any atom containing 5 valence electrons with any local bond structure
Val6	any atom containing 6 valence electrons with any local bond structure
Val7	any atom containing 7 valence electrons with any local bond structure
Specific atom types	
H	hydrogen atom with any local bond structure
Cl	chlorine atom with any local bond structure
He	helium atom with any local bond structure
Ne	neon atom with any local bond structure
Ar	argon atom with any local bond structure
Carbon atom types	
C	carbon atom with any local bond structure
Cs	carbon atom with only single bonds
Cd	carbon atom with one double bond
CO	carbon atom with one double bond, to oxygen
CS	carbon atom with one double bond, to sulfur
Cdd	carbon atom with two double bonds
Ct	carbon atom with one triple bond
Cb	carbon atom with two benzene bonds
Cbf	carbon atom with three benzene bonds
Silicon atom types	
Si	silicon atom with any local bond structure
Sis	silicon atom with only single bonds
Sid	silicon atom with one double bond
SiO	silicon atom with one double bond, to oxygen
Sidd	silicon atom with two double bonds
Sit	silicon atom with one triple bond
Sib	silicon atom with two benzene bonds
Sibf	silicon atom with three benzene bonds
Nitrogen atom types	
N	nitrogen atom with any local bond structure
N1d	nitrogen atom with one double bond and two lone pairs
N3s	nitrogen atom with up to three single bonds
N3d	nitrogen atom with one double bond and up to one single bond
N3t	nitrogen atom with one triple bond
N3b	nitrogen atom with two benzene bonds
N5s	nitrogen atom with four single bonds
N5d	nitrogen atom with one double bond two single bonds
N5dd	nitrogen atom with two double bonds
N5t	nitrogen atom with one triple bond and one single bond
N5b	nitrogen atom with two benzene bonds and one single bond
Oxygen atom types	
O	oxygen atom with any local bond structure
Os	oxygen atom with only single bonds
Od	oxygen atom with one double bond
Ot	oxygen atom with one triple bond
Oa	oxygen atom with no bonds
Sulfur atom types	
S	sulfur atom with any local bond structure
Ss	sulfur atom with only single bonds
Sd	sulfur atom with one double bond
Sa	sulfur atom with no bonds

recompiling the software, making it easier for chemists to add new information.

As an example, the reaction family H_abstraction describing the hydrogen abstraction from species XH by a radical species Y, is shown in Fig. 5 along with its reaction recipe.

For any reversible reaction $Reactant(s) \underset{k_r}{\overset{k_f}{\rightleftharpoons}} Product(s)$, thermodynamic consistency is maintained through the following relation between the forward and backwards reaction rate:

$$\frac{k_f}{k_r} = K_{eq} = \left(\frac{RT}{P^\circ}\right)^{-\Delta n} \exp\left(\frac{-\Delta G^\circ_{rxn}(T)}{RT}\right) \qquad (2)$$

where K_{eq} is the equilibrium constant of the reaction, T is the reaction temperature, R is the gas constant, P° is the standard pressure (1 bar), $\Delta G^\circ_{rxn}(T)$ is the standard reaction free energy, and Δn is the change in moles in the reaction.

For most reaction families in RMG, the rates are defined in the forward direction. The reverse kinetics are calculated through the relation $k_r = k_f/K_{eq}$ using the thermodynamic parameters estimated for the reaction species. The kinetic rate parameters are organized, like the thermodynamic parameters, using hierarchical trees based on the principle that reactions between similar reacting sites in a family will have similar rates. For each reactant site, a
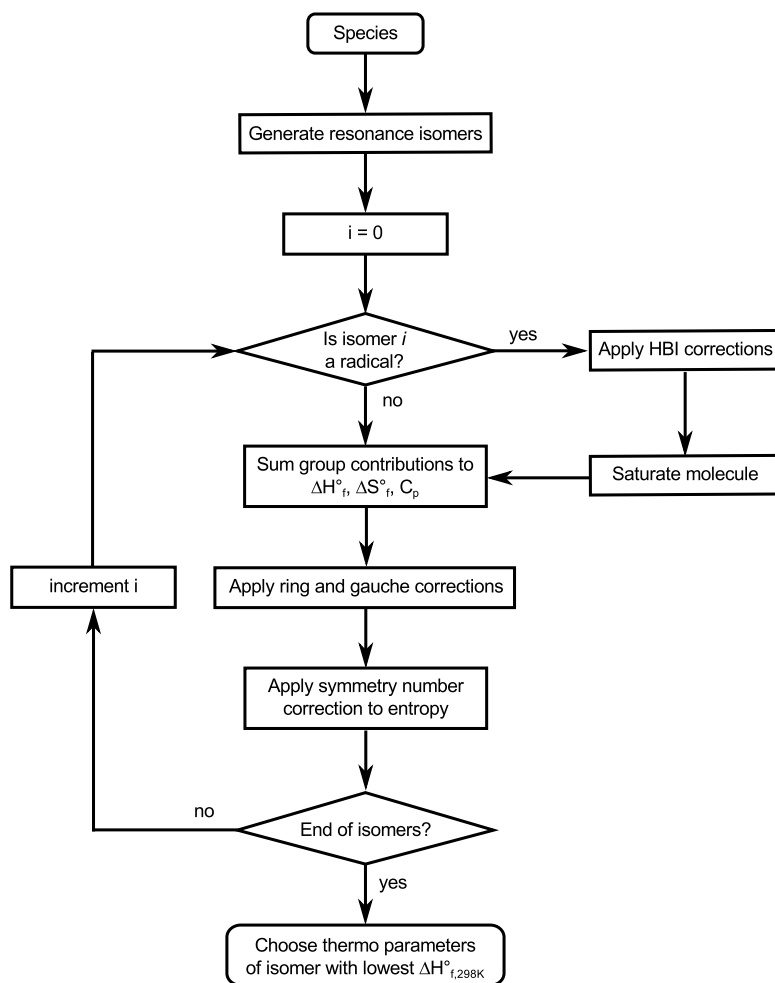
Fig. 3. Flowchart of the group additivity-based thermodynamic parameter estimation algorithm as implemented in RMG.
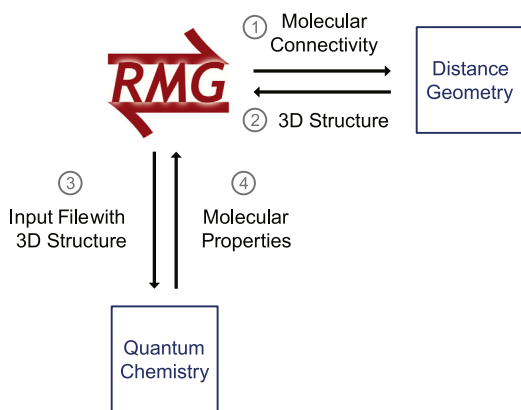

Fig. 4. Schematic depicting the QMTP interface for calculating thermodynamic parameters on-the-fly using quantum mechanics in RMG.
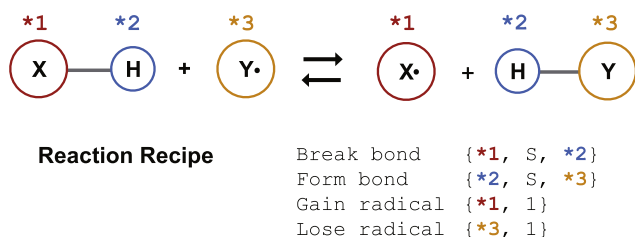*Source:* Adapted from Magoon et al. [17].

Fig. 5. Reaction template and recipe for the H_Abstraction family.

hierarchical tree is constructed with a general functional group that subdivides into children that are mutually exclusive and more specific than the parent. Partial representative trees of the two reactants from the H_abstraction family, XH and Y, are shown in Fig. 6. Individual "rate rules" are defined for a set of functional group-based reaction sites through a temperature dependent kinetic parameter $k(T)$ described by the modified Arrhenius expression:

$$k(T) = A \left(\frac{T}{1\,\text{K}} \right)^n \exp \left(-\frac{E_a}{RT} \right) \tag{3}$$

where A is the pre-exponential factor, n is the temperature exponential factor, E_a is the activation energy, R is the gas constant, and T is the temperature. Alternatively, the activation energy E_a can be related to the enthalpy of reaction ΔH_{rxn} through the constrained Evans–Polanyi relationship:

$$E_a = max(0, \alpha \Delta H_{rxn} + E_0) \tag{4}$$

where α and E_0 are constants. Finally, for endothermic reactions in RMG, the activation energy E_a is raised to at least the endothermicity of the reaction using the following relationship:
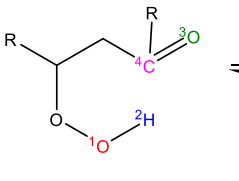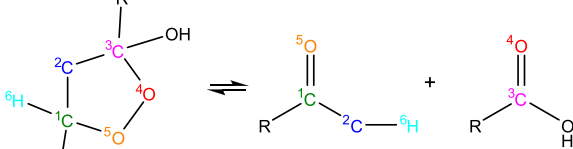
$$E_a = max(E_a, \Delta H^{\circ}_{rxn}). \tag{5}$$

Upon loading the database, RMG fills in data within the hierarchical tree through an averaging algorithm. It locates sets of parent functional groups which have children containing data, and

Table 2
RMG reaction families.

Reaction family	Reaction scheme
1+2_Cycloaddition	$^1R{=}^2R$ + $^3R\ddot{:}$ ⇌ cyclopropane (^1R–^2R–^3R ring)
1,2-Birad_to_alkene	$^1\dot{R}$–$^2\dot{R}$ ⇌ $^1R{=}^2R$
1,2_Insertion_carbene	$^1CH_2$ + ^2R–^3R ⇌ ^2R–^1C(H)(H)–^3R
1,2_Insertion_CO	$^1C^-{\equiv}^4O^+$ + ^2R–^3R ⇌ ^2R–$^1C({=}^4O)$–^3R
1,2_shiftS	^1C–^2S–$^3\dot{R}$ ⇌ $^2\dot{S}$–^3R–^1C
1,3_Insertion_CO2	$^2O{=}^1C{=}O$ + ^3R–^4R ⇌ ^3R–$^1C({=}O)$–^2O–^2R
1,3_Insertion_ROR	^3R–^4O–R + $^1R{=}^2R$ ⇌ ^3R–^1R–^2R–^4O–R
1,3_Insertion_RSR	^3R–^4S–R + $^1R{=}^2R$ ⇌ ^3R–^1R–^2R–^4S–R
1,4_Cyclic_birad_scission	^2R–$^1\dot{R}$⁓$^4\dot{R}$–^3R ⇌ $^2R{=}^1R$⁓$^4R{=}^3R$
1,4_Linear_birad_scission	$^1\dot{R}$–^2R–^3R–$^4\dot{R}$ ⇌ $^1R{=}^2R$ + $^3R{=}^4R$
2+2_cycloaddition_CCO	$^1C({=}^2C{=}O)$ + $^3R{=}^4R$ ⇌ four-membered ring ^1C–^3R/^2C–^4R
2+2_cycloaddition_Cd	$^1C{=}^2C$ + $^3R{=}^4R$ ⇌ four-membered ring ^1C–^3R/^2C–^4R
2+2_cycloaddition_CO	$^1C{=}^2O$ + $^3R{=}^4R$ ⇌ four-membered ring ^1C–^3R/^2O–^4R
Birad_recombination	$^1\dot{R}$⁓$^2\dot{R}$ ⇌ ^1R⁓^2R
Cyclic_Ether_Formation	$^1\dot{R}$⁓^2O–^3OR ⇌ ^1R⁓^2O + $^3\dot{OR}$
Diels_alder_addition	^4R–^3R/$^5R{=}^6R$ + $^1R{=}^2R$ ⇌ six-membered ring
Disproportionation	$^1\dot{R}$ + $^3\dot{R}$–^2R–^4H ⇌ ^1R–^4H + $^3R{=}^2R$
H_Abstraction	^1R–^2H + $^3\dot{R}$ ⇌ $^1\dot{R}$ + ^2H–^3R
H_shift_cyclopentadiene	cyclopentadiene with ^6H at ^1C ⇌ cyclopentadiene with ^6H at ^2C
HO2_Elimination_from_PeroxyRadical	^5H–^1R–^2R–^3O–$^4\dot{O}$ ⇌ $^3\dot{O}$–^4O–^5H + $^1R{=}^2R$
Intra_Diels_alder	six-membered ring ⇌ $^1R{=}^2R$–$^3R{=}^4R$⁓$^5R{=}^6R$
Intra_Disproportionation	$^1\dot{R}$⁓$^3\dot{R}$–^2R–^4H ⇌ ^4H–^1R⁓$^3R{=}^2R$
intra_H_migration	^3H–^2R⁓$^1\dot{R}$ ⇌ $^2\dot{R}$⁓^1R–^3H
intra_NO2_ONO_conversion	^1R–$^2N^+({=}O)$–$^3O^-$ ⇌ $^2N({:})({=}O)$–^3O–^1R
intra_OH_migration	$^1\dot{R}$⁓^2O–^3OH ⇌ ^3HO–^1R⁓$^2\dot{O}$
Intra_R_Add_Endocyclic	$^1\dot{R}$⁓$^2R{=}^3R$ ⇌ ^1R⁓$^2\dot{R}$–^3R (ring)

Table 2 (*continued*)

Intra_R_Add_Exocyclic	$^1\dot{R}\sim^2R=^3R \rightleftharpoons {}^1R\sim^2R-^3R^\bullet$
Intra_R_Add_ExoTetCyclic	$^1\dot{R}\sim^2R-^3R \rightleftharpoons {}^1R\sim^2R + {}^3R^\bullet$
Intra_RH_Add_Endocyclic	$^4H-^1R\sim^2R=^3R \rightleftharpoons {}^1R\sim^2R-^3R$ with ^4H
Intra_RH_Add_Exocyclic	$^4H-^1R\sim^2R=^3R \rightleftharpoons {}^1R\sim^2R-^3R-^4H$
intra_substitutionCS_cyclization	$^3\dot{R}\sim^1C-^2S \rightleftharpoons {}^3R\sim^1C + {}^2S^\bullet$
intra_substitutionCS_isomerization	$^3\dot{R}\sim^2S-^1C \rightleftharpoons {}^1C-^3R\sim^2S^\bullet$
intra_substitutionS_cyclization	$^3\dot{R}\sim^1S-^2R \rightleftharpoons {}^3R\sim^1S + {}^2R^\bullet$
intra_substitutionS_isomerization	$^3\dot{R}\sim^2R-^1S \rightleftharpoons {}^1S-^3R\sim^2R^\bullet$
ketoenol	$^1R=^2R-^2O-^4R \rightleftharpoons {}^4R-^1R-^2R=^2O$
Korcek_step1	(reaction scheme)
Korcek_step2	(reaction scheme)
lone_electron_pair_bond	(reaction scheme)
Oa_R_Recombination	$^2O: + {}^1R^\bullet \rightleftharpoons {}^1R-^2O^\bullet$
R_Addition_COm	$^1C\equiv^3O^+ + {}^2R^\bullet \rightleftharpoons {}^2R-^1C$ with ^3O
R_Addition_CSm	$^1C\equiv^3S^+ + {}^2R^\bullet \rightleftharpoons {}^2R-^1C$ with ^3S
R_Addition_MultipleBond	$^2R=^1R + {}^3R^\bullet \rightleftharpoons {}^2R-^1R-^3R$
R_Recombination	$^1R^\bullet + {}^2R^\bullet \rightleftharpoons {}^1R-^2R$
Substitution_O	$R-^1O-^2R + {}^3R^\bullet \rightleftharpoons R-^1O-^3R + {}^2R^\bullet$
SubstitutionS	$R-^1S-^2R + {}^3R^\bullet \rightleftharpoons R-^1S-^3R + {}^2R^\bullet$

generates a geometrically averaged rate rule using the formula:

$$\log k(T) = \sum_{i=1}^{n} \frac{\log k_i(T)}{n} \tag{6}$$

where $k_i(T)$ is the ith child rate rule and n is the total number of children. Parents which are higher up in the tree may include children that are averaged rate rules themselves. To estimate the kinetic rate parameter for an individual reaction, the most specific functional groups that describe the reaction are determined by descending the reactant trees as far as possible. This set of functional groups is then used to match rate rules in the database. Consider the simple hierarchical trees for functional groups in a bimolecular reaction shown in Fig. 7. If a reaction matches the functional group pair (A_3, B_3), but the database is missing this particular rate rule, then RMG will first search the nearest distance

parent pairs (A_1, B_3) and (B_1, A_3) to see if these pairs contain data. If one or more of these pairs contains data, the geometric average of those rates will be used as the kinetics estimate. If neither contains data, RMG will move on to the distance 2 pairs: (A_1, B_1), (A, B_3), (B, A_3), and once again check for data and use the geometric average of these data if they exist, continuing until it finds kinetic data or reaches the topmost general set of groups, which always contain data. Given the averaging approximation for kinetic parameter estimation, RMG's accuracy is highly dependent on the amount of data present in its hierarchical trees.

2.3.1. Training reactions

One drawback of a rate rule based database is the loss of molecular information. The user inputting the rate rule into the database selects the best fitting groups at the time of entry to
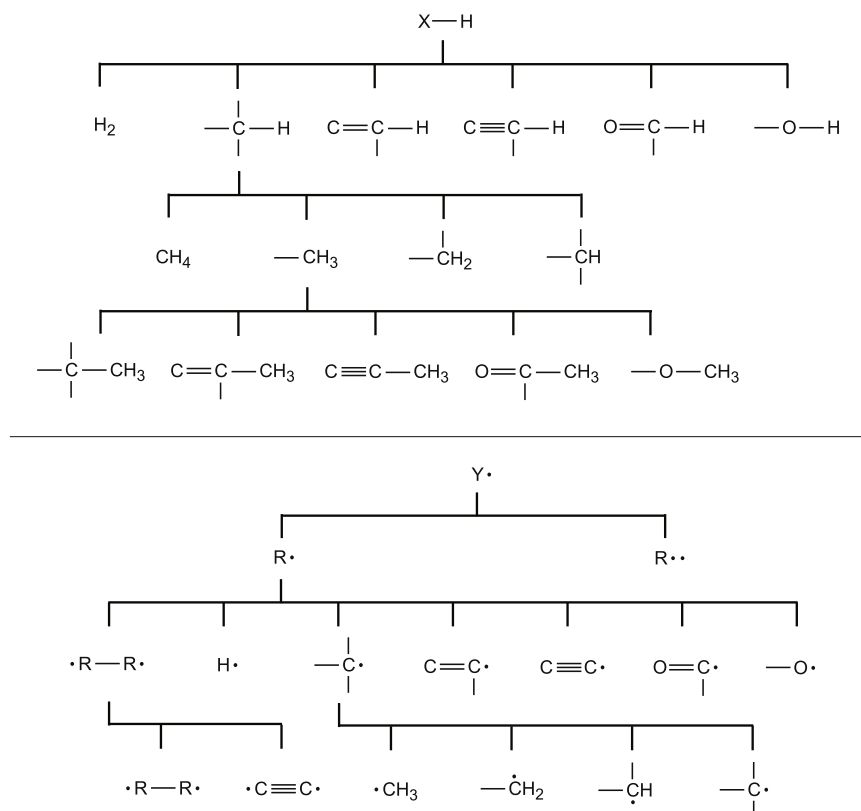
Fig. 6. Hierarchical trees for the reactants in the `H_Abstraction` family. Top: Partial tree of the X–H reactant. Bottom: Partial tree of the Y reactant.
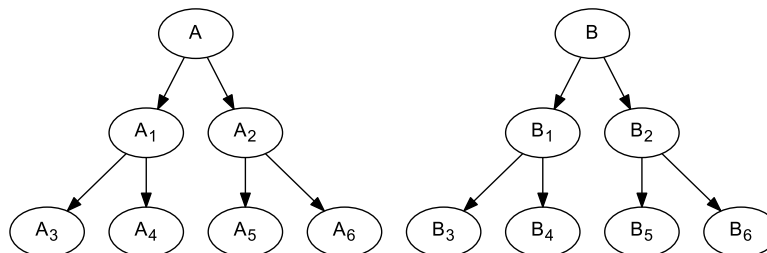

Fig. 7. Simple hierarchical trees for a bimolecular reaction family containing reactants A and B.

represent the reaction; however, this provides only functional group information rather than complete molecular information regarding the specific reactants and products. If the functional group hierarchical trees were to be altered, it becomes very difficult to reassign the rate rules without the original reaction information. Therefore, in RMG-Py, users are encouraged to add new kinetic parameter data to the database through "training" reactions, which retain information about the real molecules. When RMG finds the training reaction during model generation, it uses the exact kinetics from the training reaction. In addition, RMG automatically generates rate rule data for the set of functional groups that best match the training reaction in the current hierarchical tree so that it can improve the kinetics of similar reactions.

2.3.2. Reaction libraries

Sometimes users will wish to use reaction kinetics from literature or calculations that they do not wish to influence other kinetics. In this case, they can create a "reaction library" which contains individual reactions and kinetic data that overrides RMG's native kinetic parameter estimation scheme for those reactions

only. In RMG, multiple reaction libraries can be used with user-assigned priority.

2.3.3. Seed mechanisms

To include an entire submechanism in the model, the user can use a "seed mechanism" in the model generation process. By doing so, the seed mechanism's kinetic parameters will both override RMG's native parameter estimation as well as be forcibly included in the model. This is differentiated from reactions within a reaction library, which will only enter the model if they are deemed to be important through the flux-based model expansion algorithm. Multiple seed mechanisms can be specified in RMG with user-assigned priority.

2.4. Rate-based algorithm

RMG uses the rate-based algorithm of Susnow et al. [18] to determine which species and reactions to include in the model. The flow chart shown in Fig. 8 demonstrates the model generation process, which begins with a user-specified set of initial species and conditions (i.e. temperature, pressure) and some termination
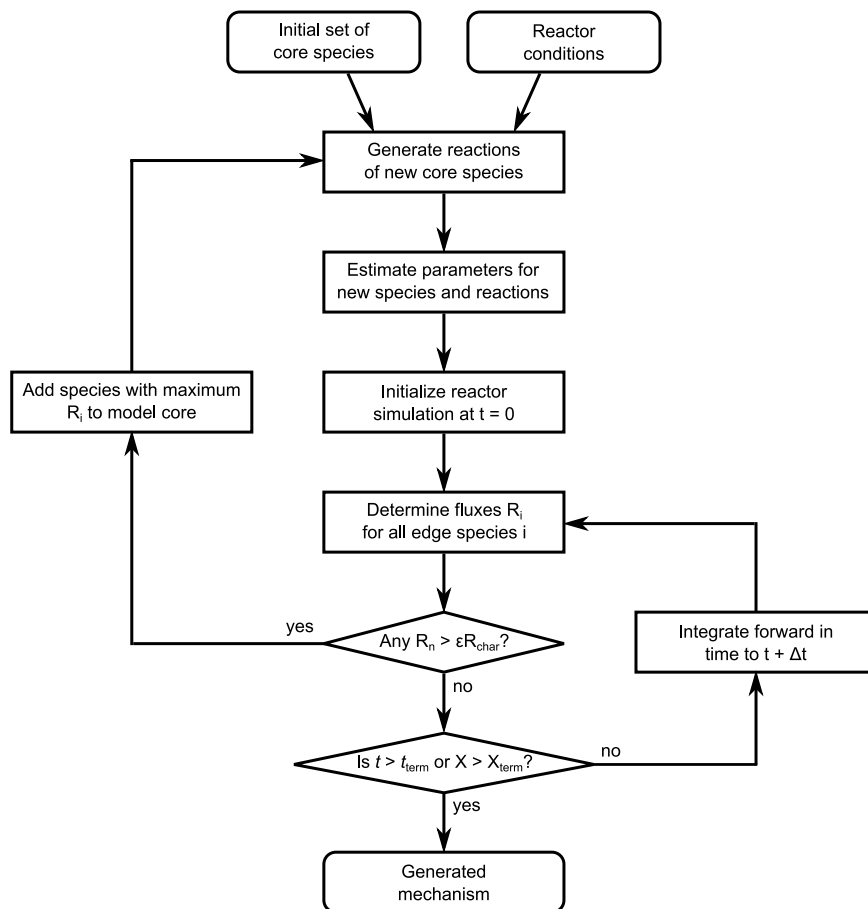
Fig. 8. Flowchart of the rate-based algorithm as implemented in RMG. The generated mechanism contains the final set of species and reactions in the core.
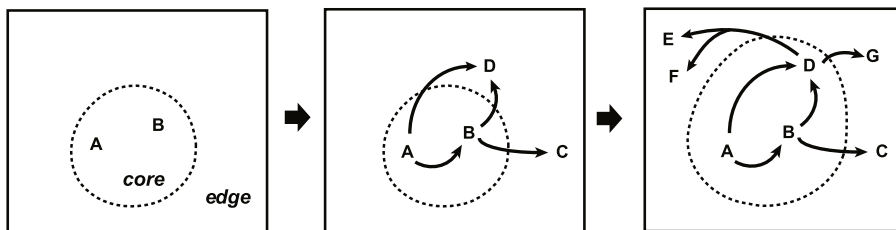

Fig. 9. Schematic depicting expansion of the model core as RMG proceeds.

criteria for which to end the simulation (i.e. a specified end time or goal conversion for some initial species). The initial species in the reaction system are placed into the "core" of the model and RMG determines all the possible reactions that can result from the core species, generating a list of possible product species on the "edge". The reactor is initialized at $t = 0$ and integrated in time until the flux $R_i = \frac{dC_i}{dt}$ to an edge species i exceeds ϵR_{char}, where ϵ is the user-specified error tolerance and R_{char} is the characteristic flux of the system, defined by:

$$R_{char} = \sqrt{\sum_j R_j^2} \quad \text{species } j \in \text{core.} \tag{7}$$

The edge species with the largest flux is brought into the core, and the reaction generation and integration steps are repeated until the termination criteria is satisfied, generating the final kinetic model, which now contains all the species and reactions that have significant fluxes at the reaction conditions. The expansion of the model core is depicted in Fig. 9.

Currently, there are two reactor types that can be simulated within RMG. The first is the `SimpleReactor`, which is an isothermal, isobaric reactor in the gas phase. The second is the `LiquidReactor`, which is isothermal and isochoric reactor in the liquid phase. More information regarding liquid phase solvation and diffusion-limited kinetics estimation is detailed in Section 3.2.

3. Additional features

A number of additional features in RMG include the ability to automatically generate pressure-dependent rate coefficients, reactions in the liquid phase, estimation of transport properties, and sensitivity analysis. In addition, the latest version of CanTherm is bundled within RMG and can be used to calculate thermochemical and kinetic quantities through transition state theory when used in conjunction with quantum chemistry software. In order to assist browsing the database and working with RMG-style species representations, a web front end has been developed and hosted on http://rmg.mit.edu, where many tools for RMG are available.
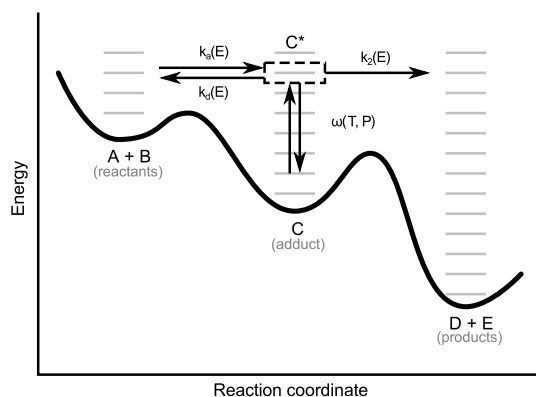
Fig. 10. A typical unimolecular system. An activated species $C*$ can be formed either from chemical activation (as the product of an association reaction) or thermal activation (via collisional excitation). Once activated, multiple isomerization and dissociation reactions may become competitive with one another and with collisional stabilization; these combine to form a network of unimolecular reactions described by a set of phenomenological rate coefficients $k(T, P)$ that connect each pair of configurations, not just those directly adjacent.

3.1. Estimation of pressure-dependent rate coefficients

Thermal unimolecular reactions proceed via nonreactive collisions with an inert third body to provide or remove the energy necessary for reaction. The reaction rate for these unimolecular reactions depend on the number of nonreactive collisions, which in turn is dependent on the pressure of the system. Under conditions where such collisions are rate-limiting, the observed phenomenological rate coefficient $k(T, P)$ is a function of both temperature T and pressure P. A unimolecular system is shown in Fig. 10.

A framework for estimating these pressure-dependent rate coefficients using high-pressure-limit kinetic data has been implemented in RMG and is described thoroughly in a separate paper [22]. The master equation model describes the unimolecular reaction network mathematically but is very computationally intensive. Thus, three methods for reducing the master equation and estimating the phenomenological rate coefficients have been implemented within RMG in addition to the master equation model: the modified strong collision method [23], the reservoir state method [24], and the chemically-significant eigenvalues method [25]. In the case of automatic generation of pressure-dependent rate coefficients, the modified strong collision method is recommended for its speed and robustness. However, detailed investigation of individual reaction networks should be refined using either the reservoir state or chemically-significant eigenvalues method as they are more accurate.

3.2. Liquid phase solvation and diffusion

A framework for modeling solvent effects in RMG has been implemented [26] and is described briefly here. To model solution phase chemistry, we must estimate the changes in the thermochemical properties of a species going from the gas phase to the solvent phase. The thermodynamics of solvation for a species can be modeled through the partition coefficient K, which is defined as the ratio of the concentration of the species in the solvent phase to that in the gas phase at equilibrium:

$$K = \left(\frac{C_{solv}}{C_{gas}} \right)_{eq}. \tag{8}$$

Assuming the chemical potential of a species i in each phase may be modeled using the equation:

$$\mu_i = \mu_i^\circ + RT \ln C_i \tag{9}$$

the change in the standard Gibbs free energy of a species in going from the gas phase to the solvent phase may be written as

$$\Delta G_{i,solv}^\circ = -RT \ln \left(\frac{C_{i,solv}}{C_{i,gas}} \right)_{eq} = -RT \ln K_i. \tag{10}$$

In other words, the free energy of solvation can be directly calculated from the partition coefficient. Thus, RMG is able to calculate the free energy change using estimated partition coefficient data.

Linear Solvation Energy Relationships (LSERs) have been developed in order to understand the fundamental nature of solute–solvent interactions. In particular, the Abraham model [27,28] uses molecular descriptors to predict the partition coefficient of a species in a large number of solvents:

$$\log_{10} K = c + aA + bB + sS + eE + lL \tag{11}$$

where K is the partition coefficient, the upper case parameters A, B, E, S and L are properties of the solute, and the lower case letters c, s, a, b, e and l are properties of the solvent.

The Abraham model is an empirical model that relies on experimental partition coefficient data to fit the model parameters. The aA and bB account for the free energy change associated with the formation of hydrogen bonds between the solute and the solvent, the sS and eE terms account for intermolecular interactions such as dipole moments, the lL term accounts for the free energy change associated with the cavity formation process, and c is a correction factor.

The solvent parameters are obtained through multiple linear regression techniques on partition coefficient data of several solutes in the solvent of interest; these parameters are found in RMG's database as a library. The technique used to obtain solute parameters A, B, E, S, L for a compound for which experimental data are available is similar to the method used for the solvent parameters; however, in order to use the model for a large variety of solutes where experimental data are unavailable, a predictive method is necessary. RMG uses the Group Additivity based scheme for the estimation of Abraham solute parameters published by Platts et al. [29].

In the absence of a quantitative understanding of the temperature dependence of solvation thermodynamics [30], we use the simple approximation to model first order temperature dependence of ΔG_{solv}:

$$\Delta G_{solv}(T) = \Delta H_{solv}^\circ - T \Delta S_{solv}^\circ. \tag{12}$$

In RMG we use the Mintz [31–33] correlations to estimate ΔH_{solv}° empirically:

$$\Delta H_{solv}^\circ = c' + a'A + b'B + e'E + s'S + l'L \tag{13}$$

where A, B, E, S and L are the same solute descriptors used in the Abraham model for the estimation of ΔG_{solv}. The lowercase coefficients c', a', b', e', s' and l' characterize the solvent and were obtained by regression to experimental data in a manner similar to that employed for the Abraham correlations.

Generation of kinetic models requires an understanding of solvation effects on elementary reaction rates. Solution phase reactions can be limited by transport of the reacting species towards each other (known as diffusive limits) and the cage-effect, which describes the increased probability of reaction between species trapped in a solvent cage. The theory behind diffusive limits in solution phase reactions is well established [34] and gives the expression of the effective rate constant k_{eff} for a diffusion limited reaction:

$$k_{\text{eff}} = \frac{4\pi RD k_r}{4\pi RD + k_r} \tag{14}$$

where k_r is the intrinsic reaction rate, R is the sum of radii of the reactants and D is the sum of the diffusivities of the reacting species. This expression represents the simplest equilibrium treatment of diffusive limits in the solution phase and is based on the Smoluchowski theory with corrections made by Collins and Kimball [35]. The effect of diffusive limits on reaction rates depends on the relative magnitudes of the intrinsic reaction rate k_r and the diffusive limit $4\pi RD$. Estimation of diffusive limits for a given reaction requires estimates of the species radii and diffusivities in different solvents. In RMG, we use the McGowan scheme [36] to estimate the species volume and its effective radius and the Stokes–Einstein correlation to estimate species diffusivities, which requires the solvent viscosity as input. Temperature-dependent viscosity correlations are included for a variety of solvents.

In order to maintain thermodynamic consistency, the forward rate constant, k_f in the reaction scheme $A \underset{k_{r,\text{eff}}}{\overset{k_f}{\rightleftharpoons}} B + C$ shall be affected if the reverse process $k_{r,\text{eff}}$ is slowed down by diffusion. In cases where both the forward and the reverse reaction rates are bimolecular, both diffusive limits are evaluated and the direction with the larger effect is used.

3.3. Transport property estimation

RMG is capable of estimating the transport properties of chemical species in a reaction mechanism automatically [37]. The transport data are saved in a CHEMKIN compatible format and can be used to run transport-dependent simulations such as laminar flames. RMG includes the GRI-Mech3.0 [38] transport library and estimates the transport properties for other molecules. The transport properties outputted are the parameters for the Lennard-Jones potential, which describes the intermolecular potential between two molecules or atoms:

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \tag{15}$$

where V is the intermolecular potential, ϵ is the well depth and measures the strength of attraction between the two particles, σ is the internuclear distance at which the intermolecular potential is zero, and r is the internuclear distance between the two particles. RMG estimates σ (in Ångstroms) and ϵ (in Joules) using the properties of the fluid at the critical point (c) through empirical correlations taken from Tee et al. [39]:

$$\sigma = 2.44 \left(\frac{T_c}{P_c} \right)^{\frac{1}{3}} \tag{16}$$

$$\frac{\epsilon}{k_B} = 0.77 T_c \tag{17}$$

where k_B is the Boltzmann constant, T_c is the critical temperature in Kelvin, and P_c is the critical pressure in bar. The critical temperature and pressure, as well as boiling point T_b, for each molecule are estimated using the Joback group additivity method [40,41]:

$$T_c = \frac{T_b}{0.584 + 0.965 \sum_i T_{c,i} - \left(\sum_i T_{c,i} \right)^2} \tag{18}$$

$$P_c = \frac{1}{\left(0.113 + 0.0032 n_a - \sum_i P_{c,i} \right)^2} \tag{19}$$

$$T_b = 198 + \sum_i T_{b,i} \tag{20}$$

where n_a is the total number of atoms in the molecule, and $T_{c,i}$, $P_{c,i}$, and $T_{b,i}$ are the group contributions to critical temperature, critical pressure, and boiling point, respectively.

RMG also provides the shape index, which indicates whether the molecule is monatomic (shape index = 0), linear (shape index = 1) or nonlinear (shape index = 2) in geometry. Currently, RMG sets the dipole moment, polarizability, and rotational relaxation collision number to zero.

3.4. Sensitivity analysis

Sensitivity analysis can be performed within RMG with respect to either the forward kinetic rate parameters or thermochemistry $\Delta G_f^\circ(T)$ of an individual species. The kinetic model can be described as a set of ordinary differential equations of the form:

$$\frac{dy}{dt} = f(y, t; \lambda) \tag{21}$$

$$y(t_0) = y_0 \tag{22}$$

where y is the solution vector, t is time, λ is the time-independent vector of input parameters, and y_0 is the initial value of y. In this definition the first order sensitivity coefficient of output y_i with respect to parameter λ_j is given as:

$$S_{i,j} = \frac{\partial y_i}{\partial \lambda_j}. \tag{23}$$

We are particularly interested in the normalized sensitivity of species i with respect to the rate coefficient k_j of reaction j:

$$\frac{\partial \ln c_i}{\partial \ln k_j} = \frac{k_j}{c_i} \left(\frac{\partial c_i}{\partial k_j} \right) \tag{24}$$

and the semi-normalized sensitivity with respect to the ΔG_j° of species j:

$$\frac{\partial \ln c_i}{\partial \Delta G_j^\circ} = \frac{1}{c_i} \left(\frac{\partial c_i}{\partial \Delta G_j^\circ} \right). \tag{25}$$

Both types of sensitivities can be automatically calculated within RMG, either in a stand-alone analysis or at the end of a model generation job. RMG relies on the differential equation solver DASPK3.1 [42] to compute the sensitivities automatically.

3.5. CanTherm

The most up to date version of CanTherm is bundled as a sub-program of RMG and contains additional features and improvements on the original program [43]. It is a tool used for computing the thermodynamic properties of chemical species, the pressure-dependent rate coefficients, and the high-pressure limit rate coefficients for chemical reactions using results from quantum chemistry calculations. 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 corrections. CanTherm is compatible with output files from several well known quantum chemistry software programs, including Gaussian, MOPAC, QChem, and MOLPRO. For different methods, CanTherm applies additional atom, bond, and spin–orbit coupling energy corrections to adjust the computed energies to the usual gas-phase reference states.

3.6. Web front-end

The RMG interactive website is located at http://rmg.mit.edu and includes a web form for generating input files for the RMG software package. The web-front end also visually displays all the databases: kinetics, thermodynamics, solvation, statistical mechanics, and transport data, and has molecular structure search features enabled. With an user-inputted molecule or reaction, the website can return RMG's estimate for kinetics or other properties, along with the original sources for those estimates. This helps provide transparency for RMG's databases and methodology. Since the RMG adjacency list format for molecules can be difficult to construct by hand, the website also provides tools to convert SMILES, InChI, CAS number, and common species names into the adjacency list format. Visualization of the molecular structures and reactions within an RMG-generated model is also possible by uploading the outputted CHEMKIN file and associated species dictionary file. Visualization through the interactive website also enables filtering of the reactions by family and species, and automatic display of evaluated kinetics and heats of reaction. Additional web tools include visual model comparison, model merging, and kinetics plotting. The RMG interactive website is written on a Django python-based framework and is also open source. The source code can be found in the Github repository https://github.com/ReactionMechanismGenerator/RMG-website/.

4. Example: n-heptane pyrolysis

The following example uses an n-heptane model generated by RMG to simulate recent pyrolysis experimental data gathered by Yuan et al. [44] at Hefei. The experimental study was performed at low pressure (400 Pa) with temperatures ranging from 780 to 1780 K. The input.py file for the n-heptane pyrolysis model is shown below:

```
database(
    thermoLibraries = ["primaryThermoLibrary"],
    reactionLibraries = [],
    seedMechanisms = [],
    kineticsDepositories = ["training"],
    kineticsFamilies = "default",
    kineticsEstimator = "rate rules",
)
generatedSpeciesConstraints(
    maximumRadicalElectrons = 1,
)
species(
    label="n-heptane",
    reactive=True,
    structure=SMILES("CCCCCCC"),
)
species(
    label="Ar",
    reactive=False,
    structure=SMILES("[Ar]"),
)
simpleReactor(
    temperature=(1000,"K"),
    pressure=(400,"Pa"),
    initialMoleFractions={
        "n-heptane": 0.02,
        "Ar": 0.98,
    },
    terminationConversion={
        "n-heptane": 0.99,
    },
```

```
    terminationTime=(1e6,"s"),
)
simpleReactor(
    temperature=(1500,"K"),
    pressure=(400,"Pa"),
    initialMoleFractions={
        "n-heptane": 0.02,
        "Ar": 0.98,
    },
    terminationConversion={
        "n-heptane": 0.99,
    },
    terminationTime=(1e6,"s"),
)
simpleReactor(
    temperature=(2000,"K"),
    pressure=(400,"Pa"),
    initialMoleFractions={
        "n-heptane": 0.02,
        "Ar": 0.98,
    },
    terminationConversion={
        "n-heptane": 0.99,
    },
    terminationTime=(1e6,"s"),
)
simulator(
    atol=1e-16,
    rtol=1e-8,
)
model(
    toleranceMoveToCore=0.02,
    toleranceInterruptSimulation=0.02,
)
pressureDependence(
    method="modified strong collision",
    maximumGrainSize=(0.5,"kcal/mol"),
    minimumNumberOfGrains=250,
    temperatures=(300,3000,"K",8),
    pressures=(0.001,100,"bar",5),
    interpolation=("Chebyshev", 6, 4),
)
```

This input file first describes databases to be used: the specific libraries and estimation method. A generatedSpeciesConstraints option is used to constrain the maximum number of electrons in the model to 1 or fewer, excluding biradical species from appearing in the model. This parameter is used to speed up model convergence by restricting the types of species that RMG considers within the model.

Then, the reactive and nonreactive species are declared: n-heptane and argon, followed by a series of one or more reaction systems that describe the initial quantities of the reactants and the temperature and pressure conditions. Since the experimental conditions span a wide range of temperatures, 3 reactor systems were used with temperatures of 1000, 1500, and 2000 K. Each reactor has a termination criterion of 99% conversion of n-heptane, with a fallback reaction time termination criterion.

Then the numerical simulation tolerances are given, followed by the user's desired model generation error tolerance ϵ described in Section 2.4. This tolerance $\epsilon = 0.02$ can be further tightened to a smaller value if the user wishes to obtain a larger and more comprehensive model.

Pressure dependence in this example is turned on because the reaction conditions are at low pressure and high temperatures, making pressure dependence highly relevant for kinetics. Note that
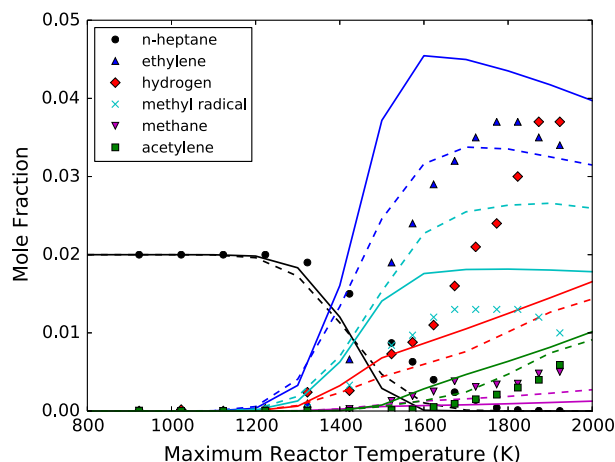
Fig. 11. N-heptane flow tube experiments at $P = 400$ Pa for an initial mixture of 2% n-heptane and 98% argon. Simulated mole fraction profiles by the RMG-generated model (lines) and LLNL v3.1 model (dotted lines.)

these input options are incomplete; the complete set of input file options can be found in the documentation. Several example input files are included within the `RMG-Py/examples/rmg/` folder.

The completed n-heptane pyrolysis RMG model contains 49 species and 638 reactions. Simulations of the flow tube pyrolysis experiments were carried out in CHEMKIN-PRO [45] for the RMG-generated model and the LLNL n-heptane detailed mechanism version 3.1 [46]. The major species found in experiment along with the model simulations are shown in Fig. 11. The RMG-generated model is a first-pass model constructed automatically solely using RMG's databases. It is able to match the LLNL model for predicting the species conversion of n-heptane. Sensitivity analysis and refinement of thermodynamic and kinetic parameters within the database typically follow in the model development cycle.

5. Design principles

RMG-Py is implemented using a new modularized design that improves upon its predecessor, RMG-Java, using smaller modules and packages grouped by functionality. The database is distinctly separate from the code and stored in a separate `RMG-database` folder. RMG-Py has also been developed using the software principles of unit testing, strong error handling, integrated documentation, and distributed version control through GitHub. The shift to Python allows us to work with a number of existing chem informatic libraries which provide a number of advanced features to RMG. Selective optimization of performance-critical areas of the code has been done through Cython [47, 48]. Cython compiles Python-style code to C through static typing, leading to over an order of magnitude or higher speed up for numerically intensive code. We are currently developing parallelization methods to further increase the model generation speed and reduce memory requirements. The new version of RMG is designed with the user in mind, with a web front end to improve ease of use and transparency within the kinetics community.

6. Conclusion

RMG is one of the most widely used automatic reaction mechanism generation codes currently available. With the shift to Python, it now takes advantage of several existing chemistry libraries and is capable of constructing mechanisms for species involving carbon, hydrogen, oxygen, sulfur, and nitrogen. RMG's features include

thermodynamic and kinetic parameter estimation, automatic generation of pressure-dependent rate coefficients, liquid phase solvation corrections, transport property estimation, and sensitivity analysis. RMG also contains the subprogram CanTherm, which computes thermodynamic parameters and high-pressure limit and pressure-dependent rate coefficients from quantum chemistry calculations. RMG has been developed for over a decade and has generated numerous validated reaction networks. This new version of RMG provides the most advanced features for reaction mechanism generation in a single open-source package, with additional web-based tools for convenience.

References

[1] F. Battin-Leclerc, E. Blurock, R. Bounaceur, R. Fournet, P.-A. Glaude, O. Herbinet, B. Sirjean, V. Warth, Towards cleaner combustion engines through groundbreaking detailed chemical kinetic models, Chem. Soc. Rev. 40 (2011) 4762–4782. http://dx.doi.org/10.1039/C0CS00207K.

[2] L.J. Broadbelt, J. Pfaendtner, Lexicography of kinetic modeling of complex reaction networks, AIChE J. 51 (8) (2005) 2112–2121. http://dx.doi.org/10.1002/aic.10599.

[3] R. Vinu, L.J. Broadbelt, Unraveling reaction pathways and specifying reaction kinetics for complex systems, Annu. Rev. Chem. Biomol. Eng. 3 (1) (2012) 29–54. http://dx.doi.org/10.1146/annurev-chembioeng-062011-081108.

[4] E. Blurock, F. Battin-Leclerc, T. Faravelli, W.H. Green, Automatic generation of detailed mechanisms, in: Cleaner Combustion, Green Energy and Technology, Springer-Verlag, London, 2013, pp. 59–92. http://dx.doi.org/10.1007/978-1-4471-5307-8_3.

[5] R. Van de Vijver, N.M. Vandewiele, P.L. Bhoorasingh, B.L. Slakman, F. Seyedzadeh Khanshan, H.-H. Carstensen, M.-F. Reyniers, G.B. Marin, R.H. West, K.M. Van Geem, Automatic mechanism and kinetic model generation for gas- and solution-phase processes: A perspective on best practices, recent advances, and future challenges, Int. J. Chem. Kin. 47 (4) (2015) 199–231. http://dx.doi.org/10.1002/kin.20902.

[6] J. Song, Building robust chemical reaction mechanisms: Next generation of automatic model construction software (Ph.D. thesis), Massachusetts Institute of Technology, 2004, http://hdl.handle.net/1721.1/30058.

[7] W.H. Green, P.I. Barton, B. Bhattacharjee, D.M. Matheu, D.A. Schwer, J. Song, R. Sumathi, H.H. Carstensen, A.M. Dean, J.M. Grenda, Computer construction of detailed chemical kinetic models for gas-phase reactors, Ind. Eng. Chem. Res. 40 (23) (2001) 5362–5370. http://dx.doi.org/10.1021/ie001088s.

[8] J.M. Grenda, I.P. Androulakis, A.M. Dean, W.H. Green, Application of computational kinetic mechanism generation to model the autocatalytic pyrolysis of methane, Ind. Eng. Chem. Res. 42 (5) (2003) 1000–1010. http://dx.doi.org/10.1021/ie020581w.

[9] M.R. Harper, K.M.V. Geem, S.P. Pyl, G.B. Marin, W.H. Green, Comprehensive reaction mechanism for n-butanol pyrolysis and combustion, Combust. Flame 158 (1) (2011) 16–41. http://dx.doi.org/10.1016/j.combustflame.2010.06.002.

[10] J.W. Allen, A.M. Scheer, C.W. Gao, S.S. Merchant, S.S. Vasu, O. Welz, J.D. Savee, D.L. Osborn, C. Lee, S. Vranckx, Z. Wang, F. Qi, R.X. Fernandes, W.H. Green, M.Z. Hadi, C.A. Taatjes, A coordinated investigation of the combustion chemistry of diisopropyl ketone, a prototype for biofuels produced by endophytic fungi, Combust. Flame 161 (3) (2014) 711–724. http://dx.doi.org/10.1016/j.combustflame.2013.10.019.

[11] G.R. Magoon, J. Aguilera-Iparraguirre, W.H. Green, J.J. Lutz, P. Piecuch, H. Wong, O.O. Oluwole, Detailed chemical kinetic modeling of JP-10 (exo-tetrahydrodicyclopentadiene) high-temperature oxidation: Exploring the role of biradical species in initial decomposition steps, Int. J. Chem. Kin. 44 (3) (2012) 179–193. http://dx.doi.org/10.1002/kin.20702.

[12] S.V. Petway, H. Ismail, W.H. Green, E.G. Estupinan, L.E. Jusinski, C.A. Taatjes, Measurements and automated mechanism generation modeling of oh production in photolytically initiated oxidation of the neopentyl radical, J. Phys. Chem. A 111 (19) (2007) 3891–3900. http://dx.doi.org/10.1021/jp0668549.

[13] J.W. Allen, Predictive chemical kinetics: Enabling automatic mechanism generation and evaluation (Ph.D. thesis), Massachusetts Institute of Technology, 2013, http://hdl.handle.net/1721.1/81677.

[14] A.S. Tomlin, T. Turányi, M.J. Pilling, Mathematical tools for the construction, investigation and reduction of combustion mechanisms, in: Low-Temperature Combustion and Autoignition, in: Comprehensive Chemical Kinetics, vol. 35, Elsevier, 1997, pp. 293–437. http://dx.doi.org/10.1016/S0069-8040(97)80019-2.

[15] S.W. Benson, J.H. Buss, Additivity rules for the estimation of molecular properties. thermodynamic properties, J. Chem. Phys. 29 (3) (1958) 546–572. http://dx.doi.org/10.1063/1.1744539.

[16] S.W. Benson, Thermochemical Kinetics, John Wiley and Sons, New York, 1968.

[17] G.R. Magoon, W.H. Green, Design and implementation of a next-generation software interface for on-the-fly quantum and force field calculations in automated reaction mechanism generation, Comput. Chem. Eng. 52 (2013) 35–45. http://dx.doi.org/10.1016/j.compchemeng.2012.11.009.

[18] R.G. Susnow, A.M. Dean, W.H. Green, P. Peczak, L.J. Broadbelt, Rate-based construction of kinetic models for complex systems, J. Phys. Chem. A 5639 (20) (1997) 3731–3740. http://dx.doi.org/10.1021/jp9637690.

[19] L.P. Cordella, P. Foggia, C. Sansone, M. Vento, A (sub)graph isomorphism algorithm for matching large graphs, IEEE Trans. Pattern Anal. Mach. Intell. 26 (10) (2004) 1367–1372. http://dx.doi.org/10.1109/TPAMI.2004.75.

[20] T.H. Lay, J.W. Bozzelli, A.M. Dean, E.R. Ritter, Hydrogen atom bond increments for calculation of thermodynamic properties of hydrocarbon radical species, J. Phys. Chem. 99 (39) (1995) 14514–14527. http://dx.doi.org/10.1021/j100039a045.

[21] G. Landrum, RDKit, http://rdkit.org.

[22] J.W. Allen, C.F. Goldsmith, W.H. Green, Automatic estimation of pressure-dependent rate coefficients, Phys. Chem. Chem. Phys. 14 (3) (2012) 1131–1155. http://dx.doi.org/10.1039/c1cp22765c.

[23] A.Y. Chang, J.W. Bozzelli, A.M. Dean, Kinetic analysis of complex chemical activation and unimolecular dissociation reactions using QRRK theory and the modified strong collision approximation, Z. Phys. Chem. 214 (2000) 1533–1568. http://dx.doi.org/10.1524/zpch.2000.214.11.1533.

[24] N.J.B. Green, Z.A. Bhatti, Steady-state master equation methods, Phys. Chem. Chem. Phys. 9 (2007) 4275–4290. http://dx.doi.org/10.1039/b704519k.

[25] J.A. Miller, S.J. Klippenstein, Master equation methods in gas phase chemical kinetics, J. Phys. Chem. A 110 (2006) 10528–10544. http://dx.doi.org/10.1021/jp062693x.

[26] A. Jalan, R.H. West, W.H. Green, An extensible framework for capturing solvent effects in computer generated kinetic models, J. Phys. Chem. B 117 (10) (2013) 2955–2970. http://dx.doi.org/10.1021/jp310824h.

[27] M.J. Kamlet, J.L.M. Abboud, M.H. Abraham, R.W. Taft, Linear solvation energy relationships. 23. a comprehensive collection of the solvatochromic parameters, π, α, and β, and some methods for simplifying the generalized solvatochromic equation, J. Org. Chem. 48 (17) (1983) 2877–2887. http://dx.doi.org/10.1021/jo00165a018.

[28] R.W. Taft, J.-L.M. Abboud, M.J. Kamlet, M.H. Abraham, Linear solvation energy relations, J. Solut. Chem. 14 (3) (1985) 153–186. http://dx.doi.org/10.1007/BF00647061.

[29] J.A. Platts, D. Butina, M.H. Abraham, A. Hersey, Estimation of molecular linear free energy relation descriptors using a group contribution approach, J. Chem. Inf. Comput. Sci. 39 (5) (1999) 835–845. http://dx.doi.org/10.1021/ci980339t.

[30] C.J. Cramer, D.G. Truhlar, Implicit solvation models: equilibria, structure, spectra, and dynamics, Chem. Rev. 99 (8) (1999) 2161–2200. http://dx.doi.org/10.1021/cr960149m.

[31] C. Mintz, M. Clark, K. Burton, W.E. Acree Jr., M.H. Abraham, Enthalpy of solvation correlations for gaseous solutes dissolved in toluene and carbon tetrachloride based on the abraham model, J. Solut. Chem. 36 (8) (2007) 947–966. http://dx.doi.org/10.1007/s10953-007-9163-0.

[32] C. Mintz, M. Clark, K. Burton, W.E. Acree Jr., M.H. Abraham, Enthalpy of solvation correlations for gaseous solutes dissolved in benzene and in alkane solvents based on the abraham model, QSAR Comb. Sci. 26 (8) (2007) 881–888. http://dx.doi.org/10.1002/qsar.200630152.

[33] C. Mintz, T. Ladlie, K. Burton, M. Clark, W.E. Acree Jr., M.H. Abraham, Enthalpy of solvation correlations for gaseous solutes dissolved in alcohol solvents based on the abraham model, QSAR Comb. Sci. 27 (5) (2008) 627–635. http://dx.doi.org/10.1002/qsar.200730128.

[34] S.A. Rice, Comprehensive Chemical Kinetics: Diffusion-limited Reactions, Elsevier, 1985.

[35] F.C. Collins, G.E. Kimball, Diffusion-controlled reaction rates, J. Colloid Sci. 4 (4) (1949) 425–437. http://dx.doi.org/10.1016/0095-8522(49)90023-9.

[36] M.H. Abraham, J.C. McGowan, The use of characteristic volumes to measure cavity terms in reversed phase liquid chromatography, Chromatographia 23 (4) (1987) 243–246. http://dx.doi.org/10.1007/BF02311772.

[37] M.R. Harper Jr., Automated reaction mechanism generation: data collaboration, heteroatom implementation, and model validation (Ph.D. thesis), Massachusetts Institute of Technology, 2011, http://hdl.handle.net/1721.1/65756.

[38] G.P. Smith, D.M. Golden, M. Frenklach, N.W. Moriarty, B. Eiteneer, M. Goldenberg, R.K.H.C.T. Bowman, S. Song, J.W.C. Gardiner, V.V. Lissianski, Z. Qin, GRI-Mech 3.0, 2011. http://www.me.berkeley.edu/gri_mech/.

[39] L.S. Tee, S. Gotoh, W.E. Stewart, Molecular parameters for normal fluids. lennard-jones 12-6 potential, Ind. Eng. Chem. Res. 5 (3) (1966) 356–363. http://dx.doi.org/10.1021/i160019a011.

[40] K.G. Joback, A unified approach to physical property estimation using multivariate statistical techniques (Master's thesis), Massachusetts Institute of Technology, 1984, http://hdl.handle.net/1721.1/15374.

[41] K.G. Joback, R.C. Reid, Estimation of pure-component properties from group-contributions, Chem. Eng. Comm. 57 (1-6) (1987) 233–243. http://dx.doi.org/10.1080/00986448708960487.

[42] L. Petzold, S. Li, DASPK3.1, http://www.cs.ucsb.edu/~cse/software.html.

[43] S. Sharma, M.R. Harper, W.H. Green, CanTherm v1.0, http://cantherm.sourceforge.net/.

[44] T. Yuan, L. Zhang, Z. Zhou, M. Xie, L. Ye, F. Qi, Pyrolysis of n-heptane: Experimental and theoretical study, J. Phys. Chem. A 115 (9) (2011) 1593–1601. http://dx.doi.org/10.1021/jp109640z.

[45] Reaction Design, CHEMKIN-PRO 15131, San Diego, 2013.

[46] M. Mehl, W.J. Pitz, C.K. Westbrook, H.J. Curran, Kinetic modeling of gasoline surrogate components and mixtures under engine conditions, Proc. Combust. Inst. 33 (1) (2011) 193–200. http://dx.doi.org/10.1016/j.proci.2010.05.027.

[47] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D.S. Seljebotn, K. Smith, Cython: The best of both worlds, Comput. Sci. Eng. 13 (2) (2011) 31–39. http://dx.doi.org/10.1109/MCSE.2010.118.

[48] R. Bradshaw, S. Behnel, D.S. Seljebotn, G. Ewing, et al. The Cython compiler, http://cython.org/.