Reaction Mechanism Simulator (RMS)

Matt Johnson
The Reaction Mechanism Generator (RMG) Software

• Open source software developed by the Green Research Group at MIT and the CoMoChEng Group at Northeastern University.

• Software aims to **predict** the chemistry of a given system by **automatically** determining the relevant species and reactions under the specified conditions.

• Particularly useful for systems involving pyrolysis and combustion chemistry.
How RMG works

Inlet composition and conditions (T, P)

Kinetic and thermodynamic parameters from libraries

User specified tolerance termination criteria (time and/or conversion)

Kinetic model

Cantera
Flux-Based Algorithm Overview

Starting species in "core"

1. Generate all possible reactions
2. Assign kinetic, thermodynamic, and transport coefficients
3. Simulate reactor

4. Check fluxes to "edge"; add "edge" species to "core" based on characteristic rate and user tolerance

5. Based on termination criteria, is "core" complete?
   - Yes
   - No

Add "E" to Model Core

Kinetic model
Automatic Mechanism Generation

1. Determine what chemical reactions can happen
   - Graph Theory and Cheminformatics

2. Estimate rates for those reactions
   - Machine Learning and Quantum Chemistry

3. Decide what reactions are important
   - Differential Equations and Graph Theory

4. Sensitivity and flux analysis
   - RMS

RMG
RMS Current Features

• Extensive and easy to extend rate and thermodynamic property calculators
• Reactors
  – Batch Constant Temperature and Pressure Ideal Gas
  – Batch Constant Temperature and Volume Ideal Dilute Liquid
  – Batch Constant Volume Adiabatic Ideal Gas
  – Ideal Plug Flow Reactor Constant T and P (implicitly)
• Sensitivity analysis for all reactors
• High quality flexible flux diagram generation
• Simulation rate of production analysis and plotting tools
• YAML based input file that is both easy to extend and easy to change
Dominant Other Codes

• Chemkin Pro
  – Commercial code
  – Based on a Department of Energy code that first released in 2001
  – Fortran

• Cantera
  – Open source code
  – Development started in 1998 first released in 2003
  – C++
Comparison With Other Codes
Chemkin Flux Diagram
RMS Flux Diagram:
Timings
Methyl Propyl Ether Combustion
403 species, 9712 reactions
Stoichiometric with O2
Constant Temperature and Pressure at 10 bar
Averages of 10 runs

RMS is ~1.7x faster than Cantera or Chemkin
RMS Improvements over Chemkin and Cantera

• More flexible and easier to manipulate input file format
• More informative and more flexible flux diagram generation
• API is much easier to write
• API is much easier to use
• New objects are automatically recognized by the parser after they are added to the API
• Faster solve
• When analytic jacobians are not available falls back to automatic differentiation using ForwardDiff
• Features
  – Diffusion limitations
  – Constant concentration species
RMS Current Dependencies

- DifferentialEquations
- Parameters
- Unitful
- ForwardDiff
- StaticArrays
- LinearAlgebra
- MathProgBase
- Clp
- PyCall
  - rdkit
  - rmgpy
  - pydot
- Images
- YAML
- SpecialFunctions
- PyPlot
- Test
Codes used in testing

- Chemkin Pro
- Cantera
RMS Future Features

• Analytic Jacobians (mostly done)
• Domains with parameterized temperature, pressure and volume (very soon)
• Interfaces
  – Inlets
  – Outlets
  – Pipes
  – Heat sources
  – Flows and Diffusive Fluxes
• Multi-phase reactors (gas-liquid and catalysis)
Chemkin Input File:

```
RECTIONS     KCAL/MOLE  MOLES

H(3)+O2(2)=OH(5)+O(4)            1.040000e+14  0.000   15.286
H2(6)+O(4)=OH(5)+H(3)            3.818000e+12  0.000   7.948
DUPLICATE
H2(6)+O(4)=OH(5)+H(3)            8.792000e+14  0.000   19.170
DUPLICATE

OH(5)+H2(6)=H2O(7)+H(3)          2.160000e+00  1.510   3.430
OH(5)+OH(5)=H2O(7)+O(4)          3.340000e+04  2.420  -1.930

H2(6)+M=H(3)+H(3)+M              4.577e+19  -1.400  104.380
CO2(13)/3.80/ H2O(7)/12.00/ H2(6)/2.50/ He(9)/0.00/ CO(12)/1.90/ Ar(8)/0.00/

H2(6)+Ar(8)=Ar(8)+H(3)+H(3)      5.840000e+18  -1.100  104.380
H2(6)+He(9)=He(9)+H(3)+H(3)     5.840000e+18  -1.100  104.380

O(4)+O4(4)=O2(2)+M               6.165e+15  -0.500   0.000
CO2(13)/3.80/ H2O(7)/12.00/ H2(6)/2.50/ He(9)/0.00/ CO(12)/1.90/ Ar(8)/0.00/

Ar(8)+O(4)+O(4)=Ar(8)+O2(2)      1.886000e+13  0.000  -1.788
He(9)+O(4)+O(4)=He(9)+O2(2)     1.886000e+13  0.000  -1.788

H(3)+O(4)=OH(5)+M                4.714e+18  -1.000   0.000
CO2(13)/3.80/ H2O(7)/12.00/ H2(6)/2.50/ He(9)/0.75/ CO(12)/1.90/ Ar(8)/0.75/

H2O(7)+M=OH(5)+H(3)+M            6.064e+27  -3.322  120.790
CO2(13)/3.80/ H2O(7)/0.00/ H2(6)/3.00/ He(9)/1.10/ 02(2)/1.50/ N2/2.00/ CO(12)/1.90/

H2O(7)+H2O(7)=OH(5)+H2O(7)+H(3)  1.006000e+26  -2.440  120.180

H(3)+O(2)(+M)=H2O(18)(+M)        4.651e+12  0.440   0.000
CO2(13)/3.80/ H2O(7)/14.00/ H2(6)/2.00/ He(9)/0.80/ 02(2)/0.70/ CO(12)/1.90/ Ar(8)/0.67/
LDW/ 6.366e+20  -1.720  0.525 /
TROE/ 5.000e-01 1e-30 1e+30 /

H2O(18)+H(3)=H2(6)+O2(2)         2.750000e+06  2.090  -1.451
```
Cantera Input File:

```plaintext
species(name='S(117474)',
   atoms='H:5 C:4 O:5',
   thermo={NASA([100.00, 1355.96],
       [-9.32154322E-01, 9.24590680E-02, -1.00027691E-04,
        5.16967734E-08, -1.00098286E-11, -8.51414276E+04,
        3.42204119E+01]),
       NASA([1355.96, 5000.00],
       [2.58818270E+01, 8.22784427E-03, -2.87207840E-06,
        3.0273915E-10, -1.96623441E-14, -9.19414117E+04,
        -1.01567108E+02]))},
   transport=gas_transport(geom='nonlinear',
       diam=7.873,
       well_depth=664.505))

# Reaction data

# Reaction 1
reaction('H(3) + O2(2) <= O(4) + OH(5)', [1.040000e+14, 0.0, 15.206])

# Reaction 2
reaction('O(4) + H2(6) <= H(3) + OH(5)', [3.818000e+12, 0.0, 7.940],
   options='duplicate')

# Reaction 3
reaction('O(4) + H2(6) <= H(3) + OH(5)', [8.792000e+14, 0.0, 19.17],
   options='duplicate')

# Reaction 4
reaction('OH(5) + H2(6) <= H(3) + H2O(7)', [2.160000e+08, 1.51, 3.43])

# Reaction 5
reaction('OH(5) + OH(5) <= O(4) + H2O(7)', [3.340000e+04, 2.42, -1.93])

# Reaction 6
three_body_reaction('H2(6) + M <= H(3) + H(3) + M', [4.577000e+19, -1.4, 104.38],
   efficiencies='He(9):0.0 CO(12):1.9 H2(6):2.5 CO2(13):3.8 Ar(8):0.0 H2O(7):12.0')

# Reaction 7
reaction('H2(6) + Ar(8) <= H(3) + H(3) + Ar(8)', [5.840000e+18, -1.1, 104.38])

# Reaction 8
reaction('H2(6) + He(9) <= H(3) + H(3) + He(9)', [5.840000e+18, -1.1, 104.38])

# Reaction 9
three_body_reaction('O(4) + O(4) + M <= O2(2) + M', [6.165000e+15, -0.5, 0.0],
   efficiencies='He(9):0.0 CO(12):1.9 H2(6):2.5 CO2(13):3.8 Ar(8):0.0 H2O(7):12.0')

# Reaction 10
reaction('O(4) + O(4) + Ar(8) <= Ar(8) + O2(2)', [1.886000e+13, 0.0, -1.788])

# Reaction 11
reaction('O(4) + O(4) + He(9) <= He(9) + O2(2)', [1.886000e+13, 0.0, -1.788])

# Reaction 12
three_body_reaction('H(3) + O4 + M <= OH(5) + M', [4.714000e+18, -1.0, 0.0],
   efficiencies='He(9):0.75 CO(12):1.9 H2(6):2.5 CO2(13):3.8 Ar(8):0.75 H2O(7):12.0')

# Reaction 13
three_body_reaction('H2O(7) + M <= H(3) + OH(5) + M', [6.064000e+27, -3.322, 120.79],
```