# Re-introduction to RMG, Enlarge Reaction Filter Algorithm, and User Accessibility Features

Nick Vandewiele and Connie Gao

RMG Meeting

January 15, 2015

# Agenda

- Intro to RMG
  - Learning
  - Contributing
  - Running
- Update on recent advances:
  - enlarge reaction filter
  - windows anaconda
  - new visualization features
- Discussion of issues

# Learning about RMG

http://rmg.mit.edu

http://cheme.scripts.mit.edu/green-group/rmg/

http://rmg.sourceforge.net

Many publications:

RMG-Py: Gao et al. *Comp. Phys. Comm. (*accepted)

P-dep: Allen et al. *PCCP* **14**(3) , 1131-1155 (2012)

QMTP: Magoon et al. *Comp. Chem. Eng.* **52**, 35-45 (2013)

...

# Contributing to RMG

through a *version control* system called “git”

Collaboration platform: Github.com
https://github.com/ReactionMechanismGenerator/

Reporting issues

Tests: unit tests, continuous integration

# Running RMG

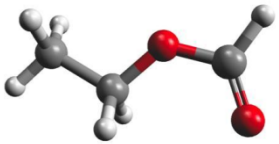
On your local computer

On pharos.mit.edu

- Get an account on pharos.mit.edu (talk to admin Nick)
- See e-mail.

# Enlarge Reaction Filter Algorithm

# Original Flux-based algorithm



**Starting species in "core"**

**Kinetics and thermo databases and reaction templates**

**Reactor conditions (T, P)**

**1** Find all possible reactions

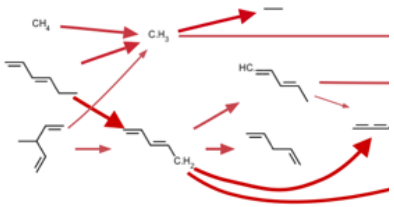
**2** Simulate reactor

**3** Is core complete?

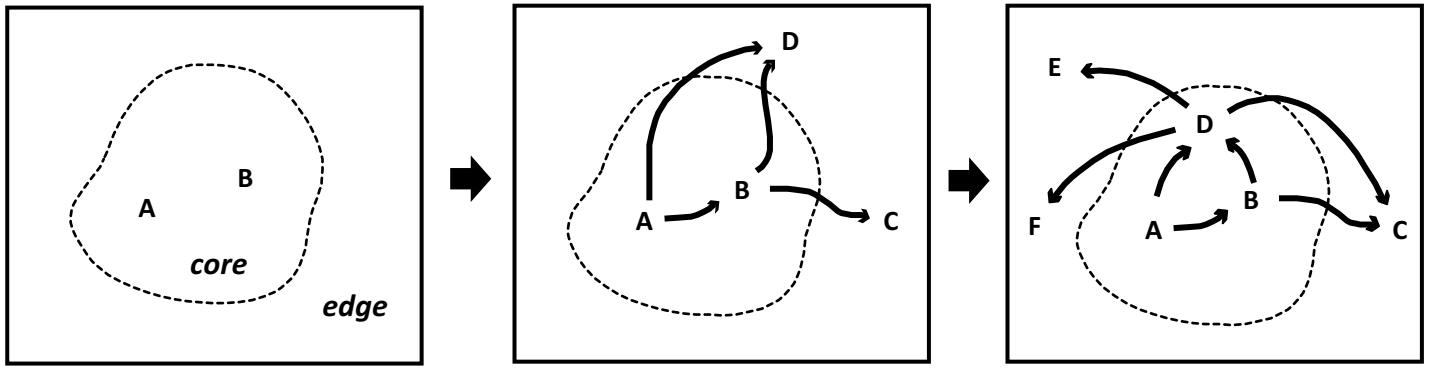
**Y**

**N**

**4** Add edge species with highest flux to core



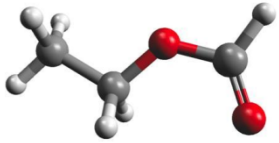
**Kinetic model**



# Isolating the speed and memory pain point in RMG

- `Pain point is generateReactions()`
  - Applying a reaction template recipe between permutations of all reactants
- Current algorithm:
  1. Reacts species together (slow)
  2. Determines which reactions are negligible (fast)
- Let's prevent species from reacting together through **pre-filtering** when we know their reactions will be negligible

# Modified flux algorithm with reaction filtering



**Starting species in "core"**

**Kinetics and thermo databases and reaction templates**

**Key Takeaways**

- Don't waste CPU time on generating reactions that will always have negligible flux
- Don't react all core species together at each step, only react them when deemed 'ready' to react. This avoids the problem of congestion we used to have with too many initial species being added by seed mechanisms.

1 Create arrays `unimolculecularThreshold` and `bimolecularThreshold` based on initial concentrations

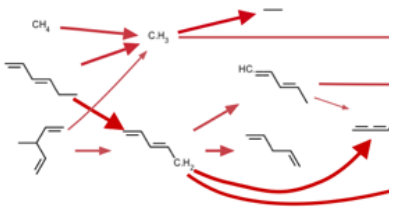
2 Generate `unimolecularReact` and `bimolecularReact` flags

3 Find all possible reactions using react flags

4 Simulate reactor, Check fluxes of edge reactions

5 Is core complete?

Y



**Kinetic model**

N

6 Add edge species with highest flux to core

7 Simulate reactor using new core, update `unimolecularThreshold` and `bimolecularThreshold`

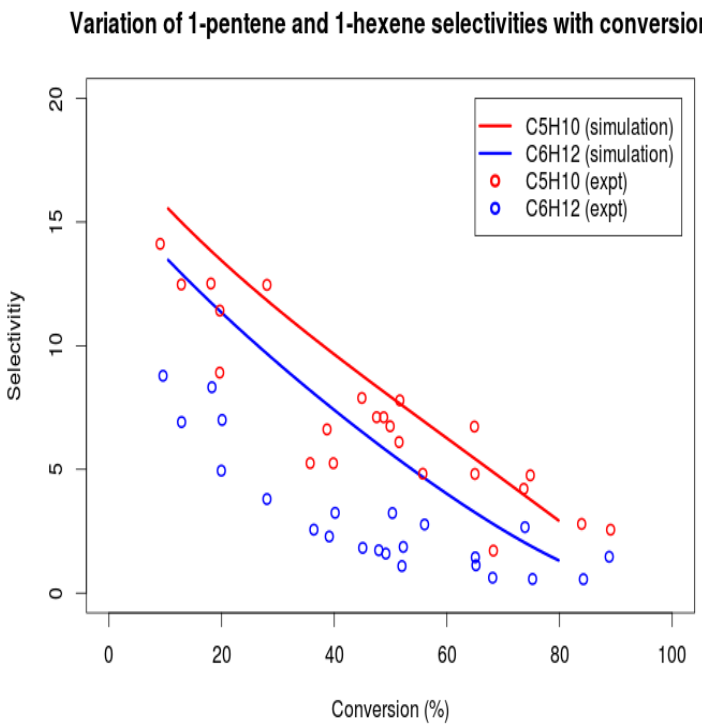
# Key variables for filtering reactions based on diffusion limit rate $k_{diff}$

- `unimolecularThreshold` and `bimolecularThreshold`
  - Binary arrays storing flag for whether a species or a pair of species are above a reaction threshold
  - Threshold is set to `True` if rate = $k_{diff}*C_A$ > toleranceMoveToCore*$rate_{char}$ at **any given time t** in the reaction system
- `unimolecularReact` and `bimolecularReact`
  - Binary arrays storing flags for when the `unimolecularThreshold` or `bimolecularThreshold` flag shifts from `False` to `True`
    - RMG should then react those species together

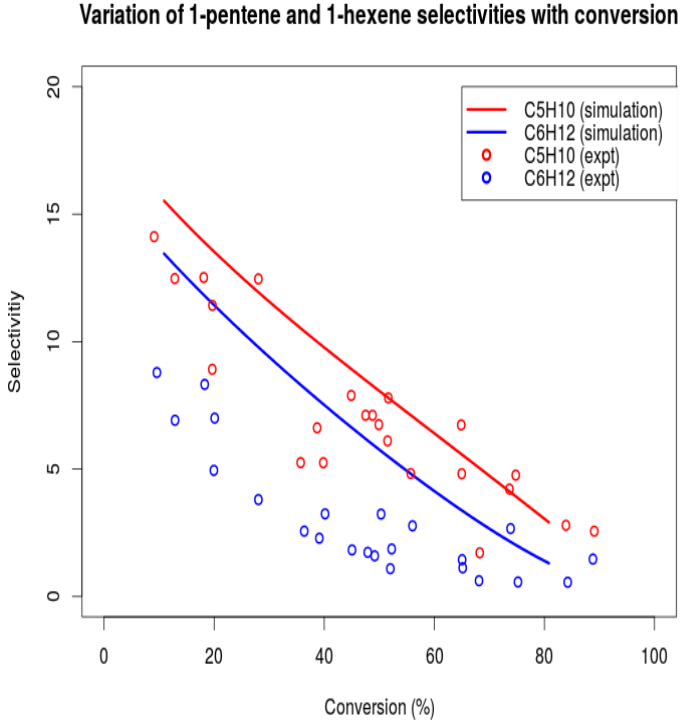
# Results show more than 10x speedup for heptane case with similar accuracy and memory usage

- Master branch: 1 day, 2500 MB, 128 spc/1410 rxns (core)
- `enlargeReactionFilter` branch: 2.5 hr, 2000 MB, 248 spc/2420 rxns (core)

Master




eRF



# Testing for robustness

- Handling pressure dependence
- Testing buiding real models

# Opportunities forfurther speedup

- Individual kmax for each family may further filter reactions than the conservative kdiff, especially when evaluated at lower temperatures
  - Pre-estimates of thermo $\Delta H_{rxn}$ may be necessary
- Numerical optimization and cythonization

# User Accessibility Features

# RMG-Py binaries for Windows

- Currently compiling dependencies using Anaconda for Windows
  - It is possible to compile gfortran and c/c++ code without Visual Studio!!
    - Uses `mingwpy` conda package
  - Compiling and testing using Windows 7 Enterprise 32-bit and 64-bit VMs
  - PyDAS and PyDQED are working!
  - Only openbabel left to package
- Binaries to be released with RMG-Py paper

# New CHEMKIN visualization features on RMG website

- Thermo comments
- Heats of Reaction: $\Delta H_{rxn}$ ,$\Delta S_{rxn}$, and $\Delta G_{rxn}$
- Filtering reactions by species as well as families
- Flux pairs now saved into CHEMKIN comments
  - Useful for external flux analysis tools
  - iPython and automatic flux analysis tools in the pipeline!

This is all live on http://rmg.mit.edu

# Future Wiki for RMG

- How can we store guides that don't belong in the documentation but are helpful to developers?
  - i.e. How to set up a job on Pharos, how to use and build anaconda binaries, how to set up RMG web server
- Richard has suggested Github Wiki
  - Open source documents for quickly editable guides
    - Does not require explicit sharing: better than Slack and Google docs
    - Markdown support
  - But is there security for password sensitive information?

# Key variables for filtering reactions based on diffusion limit rate $k_{diff}$

- `unimolecularThreshold` and `bimolecularThreshold`
  - Binary arrays storing information about whether a species or a pair of species should react
  - Threshold is `True` if unimolecular rate = $k_{diff}*C_A$ or bimolecular rate = $k_{diff}*C_A*CB$ is greater than toleranceMoveToCore*$rate_{char}$ at **any given time t** in the reaction system.
  - unimolecular $k_{diff}$=1e14 s^-1 for unimolecular
  - bimolecular $k_{diff}$=1e13 cm^3/mol*s
  - In the initialization step, the threshold arrays are set by initial concentrations
    - If Species A has a positive initial concentration, then its unimolecular threshold is `True`
    - If Species A and Species B both have positive initial concentrations, then their bimolecular threshold is `True`
    - If Species A has positive concentration but Species B has zero concentration, their bimolecular threshold is `False`
- `unimolecularReact` and `bimolecularReact`
  - Binary arrays storing flags for when the unimolecularThreshold or bimolecularThreshold shifts from `False` to `True` for any core species or pair of core species, indicating that RMG should explore reactions for those species and enlarge the edge

# Detailed write-up for reaction filter algorithm

1. Set up unimolecular and bimolecular reaction threshold arrays in reaction model initialization step. Allow any positive concentration species to have their unimolecular and bimolecular reaction thresholds to be set to TRUE.
   - If Species A has a positive initial concentration, then its unimolecular threshold is TRUE
   - If Species A and Species B both have positive initial concentrations, then their bimolecular threshold is TRUE,
   - If Species A has positive concentration but Species B has zero concentration, their bimolecular threshold is FALSE
2. Create unimolecular and bimolecular react flags based on changes in reaction threshold during model generation.
3. Enlarge Model Edge based on unimolecular and bimolecular react flags (the idea is that if these species concentrations go above the threshold for reaction during the entirety of the reactor residence time, we will allow them to react, otherwise, don't waste time on reaction CPU time for reactions that will always have negligible flux)
4. Add species with largest flux to core, add all associated edge reactions to core
5. Simulate reaction system with new core
   - Returns binary array of unimolecular and bimolecular reaction thresholds:
   - At every time step, evaluate if unimolecular rate = $k*C_A$ or bimolecular rate = $k*C_A*CB$ is greater than $toleranceMoveToCore*rate_{char}$
6. Check if unimolecular and bimolecular reaction thresholds have changed, any core species that can now reaction together will produce a True react flag. This tracks if previous species are reacted.
   - Note that the key difference here is that we don't react all core species together at each step. They are only reacted together when they are deemed 'ready' to react. This avoids the problem of congestion we used to have during seed mechanism addition.
7. Instruct RMG to enlarge edge using the new set of unimolecular and bimolecular react flags
8. Resimulate reaction system the normal way until the edge species with highest flux above user tolerance if found, repeat process.