Capstone Project Final Presentation
Automated Model Reduction for Atmospheric Chemical Mechanisms (AMORE Project)

Faculty Mentor: Dr. Faye McNeil, Dr. Siddhartha Sen

Students: Lingrui Luo, Zihe Wang, Zijing Wang, Di Ye, Qiaoge Zhu
Contents

- Introduction
- Dataset Overview
- Procedure Walkthrough
  - Data Preprocessing
  - Directed Graph
- Product
- Discussion & Future Work
- References
Atmospheric Chemistry Modeling: Bridging Scales

GLOBAL

REGIONAL

PROCESS

MOLECULAR

Length Scale - Computational expense - ‘Size’ of chemical mechanism
AMORE (Atmospheric Chemistry MOdel REduction)

An automated tool for flexibly generating accurate reduced chemical mechanisms for use in atmospheric chemistry and air quality models.

Two-step model reduction: graph theory-based strongly-connected component analysis followed by simplified quasi-steady state analysis.
Introduction

Objective

- Develop and implement a graph-based algorithm to automatically generate reduced gas-phase chemical mechanisms.
- Improve accuracy and efficiency of large-scaled atmospheric chemistry models.

Why it is meaningful?

- Enable the improved, sustained and consistent development of chemical mechanisms for air quality forecasting, research, and policy analysis.
- Understand the air quality, especially how air pollutants move in the atmosphere.
- Limitation of the computational complexity
- The chemical stiffness in the detailed chemical mechanisms
Dataset Overview

Isoprene: Fuel for atmospheric oxidation

![Chemical Reaction Diagram]

Dataset Format: **Kinetic Preprocessor (KPP) format**

- A file of chemical reactions in the full mechanism.
- A file of species involved in the full mechanism.
- A file contains editable input parameters and initial values of included species, temperature, etc.

### Isoprene Mechanisms

<table>
<thead>
<tr>
<th>Isoprene Mechanisms</th>
<th>Number of species</th>
<th>Number of reactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCM v 3.3.1</td>
<td>602</td>
<td>1926</td>
</tr>
<tr>
<td>CB05</td>
<td>18</td>
<td>9</td>
</tr>
<tr>
<td>CB6r2</td>
<td>38</td>
<td>22</td>
</tr>
<tr>
<td>GEOS-Chem v11.02c</td>
<td>106</td>
<td>335</td>
</tr>
<tr>
<td>GFDL-AM3</td>
<td>23</td>
<td>12</td>
</tr>
<tr>
<td>GISS ModelE</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>Paulot 2009 (high and low NOx, reduced)</td>
<td>45</td>
<td>28</td>
</tr>
<tr>
<td>Wennberg 2018 (full)</td>
<td>385</td>
<td>810</td>
</tr>
</tbody>
</table>
Procedure

Data Preprocessing

An Example of Chemical Reaction

- List of tuple with reaction and reaction rate formula
  - ISOP + OH = ISOP1OHc: $2.7 \times 10^{-11} \times \exp\left(\frac{390}{\text{TEMP}}\right) \times 0.63 \times 0.5$

- A list of tuple with reaction and reaction rate formula
  - (‘ISOP + OH = ISOP1OHc’, $2.7 \times 10^{-11} \times \exp\left(\frac{390}{\text{TEMP}}\right) \times 0.63 \times 0.5$)
Procedure

Data Preprocessing

\[ \text{ISOP} + \text{OH} = \text{ISOPOHc} : 2.7E-11 \times \exp\left(\frac{390}{\text{TEMP}}\right) \times 0.63 \times 0.5 \]

The weight of the reaction for the product

\[
R_A = \sum_{i=1,I} v_{A,i}w_i
\]

\(v_{A,i}\) : the stoichiometric of species A

\(w_i\) : the production rate

\[
\tau_{\text{ISOPOHc}} = v_{\text{ISOPOHc},i} \times w_i
\]

\[= v_{\text{ISOPOHc},i} \times k \times c[\text{ISOP}] \times c[\text{OH}]\]

\[= 1 \times 2.7E-11 \times \exp\left(\frac{390}{\text{TEMP}}\right) \times 0.63 \times 0.5 \times c[\text{ISOP}] \times c[\text{OH}]\]

- \(k\) : the reaction rate for this particular reaction (equation)
- \(c[\text{ISOP}]\) and \(c[\text{OH}]\) are the initial concentration values of ISOP and OH respectively
Procedure

Data Preprocessing

How to measure the influence of one species on another?

\[ r_{AB} = \frac{\sum |\nu_{A,i} w_i \delta_{Bi}|}{\sum |\nu_{A,i} w_i|} \]

Normalized contribution of species B to the production rate of species A

\[ \delta_{Bi} = \begin{cases} 
1, & \text{if the } i\text{th elementary reaction involves species B} \\
0, & \text{otherwise}
\end{cases} \]
Procedure

Graph Construction

Directed Relation Graph (DRG)

- Node: a species in the detailed mechanism
- Edge from A to B: A is dependent on B (the removal of B would directly induce significant error to the production rate of A)

\[
\text{ISOP} + \text{OH} = \text{ISOP1OHc}: 2.7 \times 10^{-11} \times \exp\left(\frac{390}{\text{TEMP}}\right) \times 0.63 \times 0.5
\]
Procedure

Graph Construction

Directed Relation Graph (DRG)

- Perform a depth first search starting from the ‘starting sets’ and obtain a skeleton graph
- We used organic species as starting sets while treating others as background species.

Starting set = ['A']
Procedure

Graph Reduction

Method:

- Define a threshold value $\epsilon$
- Only keep the paths that all the edges have a weight greater than $\epsilon$ such that $r_{AB} \geq \epsilon$
Results

The number of Species for various threshold values

The number of species in reduced mechanism with different epsilon values
Product

Python Package Chem_graph

We created a python package called chem_graph accessible on Github. Our product contains following functionalities:

- Read and parse the raw file, and construct the graph accordingly
- Given threshold value and starting set, compute the reduced graph and dependent set
- Visualize the graph
Product

Algorithm 1 Original DFS Approach

Given $r_{AB}$ dictionary, list of $\epsilon \in E$. starting set $S$
Create graph $g$ based on $r_{AB}$

for each $\epsilon$ in $E$
do
    create visited set $v$
    create stack $s$, push $S$ into $s$
    while $s$ not empty do
        pop one element $e$ from $s$
        for each neighbor $n$ of $e$
do
            if the edge weight smaller than $\epsilon$, continue
            if $n$ visited, continue
            push $n$ to $v$ and $s$
        end
    end
    output $v$ for $\epsilon$
end

Running time: $O(n(V+E))$

Algorithm 2 Modified DFS Approach

Given $r_{AB}$ dictionary, list of $\epsilon \in E$. starting set $S$
Sort $r_{AB}$ dictionary based on the edge weight from high to low;
Sort $E$ from high to low;
Create visited set $v$
Create set $c$ to keep track of graph

for each $\epsilon$ in $E$
do
    include new edges with value smaller than last $\epsilon_0$ but larger than current $\epsilon$
    for each new edge do
        if the destination node of the new edge is not visited, continue
        DFS from the destination node, push the newly visited to $v$
    end
    output current $v$ for current $\epsilon$
end

Running time: $O(n\log(n) + V+E)$
Sample Usage

```python
from chem_graph import *
demo_graph = ChemGraph()
priority_species = ['ISOP1CO400Hc']
demo_graph.construct(path='isoprene_full_v5', starting_set=priority_species, must_contain=['ISOP'])
```
Product
Sample Usage

visualize_graph(graph=demo_graph.whole_graph,
highlight=priority_species)

visualize_graph(graph=demo_graph.reduced_graph,
highlight=priority_species)
Future Direction

- Evaluation on the reduce mechanism
- Evaluation on the generated graph
References

Thank You!