An adaptive auto-reduction solver for speeding up integration of chemical kinetics
Implementation in the Kinetics Pre-Processor (KPP) 3.0.0

Haipeng Lin, Michael S. Long, Rolf Sander, Adrian Sandu, Robert M. Yantosca, Lucas A. Estrada, Lu Shen, Daniel J. Jacob

Atmospheric Chemistry Mechanisms (ACM) Conference, December 2022
The **Kinetics Pre-Processor (KPP)** is a software tool to generate solver code for chemical kinetics.

The reaction mechanism is described as follows:

- \(<1>\) \( \text{O}_2 + \text{hv} = \text{O} + \text{O}_2 : (2.643\times10^{-10}) \times \text{SUN}^3 \)
- \(<2>\) \( \text{O} + \text{O}_2 = \text{O}_3 : (8.018\times10^{-17}) \)
- \(<3>\) \( \text{O}_3 + \text{hv} = \text{O} + \text{O}_2 : (6.120\times10^{-4}) \times \text{SUN} \)
- \(<4>\) \( \text{O} + \text{O}_3 = 2\text{O}_2 : (1.576\times10^{-15}) \)

Running through KPP, the generated code is included in atmospheric models such as GEOS-Chem, WRF-Chem, and MESSy.

KPP 2.1: Sandu and Sander, 2006; Damian et al., 2002

---

**Describe the mechanism**

**Run through KPP**

**Generated code is included in atmospheric model**
(e.g., GEOS-Chem, WRF-Chem, MESSy...)**

KPP 2.1: Sandu and Sander, 2006; Damian et al., 2002
KPP has recently been updated to **version 3.0.0**, with improved documentation, diagnostics, and an **adaptive solver option**

- New documentation at kpp.readthedocs.io
- GitHub-hosted repository (https://github.com/KineticPreProcessor) with Continuous Integration (CI): zero numerical differences across versions
- New diagnostics such as **production and loss families**
- (optional) **Adaptive solver for auto-reduction of chemical mechanisms**
Integration of chemical kinetics is a major computational bottleneck in atmospheric chemistry models

\[ \frac{dC_i}{dt} = P_i(C) - L_i(C), \quad i = 1, 2, \ldots, N \]

- \( N \sim 291 \) chemical species (GEOS-Chem 13.4.0) and **rapidly increasing**!
- “Chemistry” most expensive operation in the GEOS-Chem model
- **However, not all the complexity is needed everywhere!**
Reducing chemical mechanisms in global models is made difficult by the wide range of conditions and difficulty in constraining error growth.

Santillana et al., 2010
Each species is ‘fast’ or ‘slow’ in each grid box

Shen et al., 2020, 2022
Select from pre-selected 20 chemical regimes

Keller and Evans, 2019
Machine learning methods have high error growth
High dimensionality is hard!
Classifying species into fast and slow based on production and loss rates is a **mechanism-agnostic** method

\[
\frac{dC_i}{dt} = P_i(C) - L_i(C), \quad i = 1, 2, \ldots, N
\]

• If \( P_i > \delta \) or \( L_i > \delta \), then the species is **fast** and kept within the full solver.
  Otherwise, use an explicit first-order approximation.
• \( \delta \) can be defined based on local conditions – e.g., based on OH or NO\(_2\) rates.
Previous implementations of the adaptive solver suffered from excessive overhead due to the need to reconstruct the reduced mechanism Jacobian. Can we find a representation of the ‘reduced mechanism’ in the same format?

(R1) $B + C \to A + \text{products}$  (R2) $A + C \to \text{products}$  (R3) $C + X \to \text{products}$

(a) Full Mechanism

\[
\begin{array}{ccc}
A & B & C \\
A & \frac{\partial s_A}{\partial [A]} & \frac{\partial s_A}{\partial [B]} & \frac{\partial s_A}{\partial [C]} \\
B & \frac{\partial s_B}{\partial [A]} = 0 & \frac{\partial s_B}{\partial [B]} & \frac{\partial s_B}{\partial [C]} \\
C & \frac{\partial s_C}{\partial [A]} & \frac{\partial s_C}{\partial [B]} & \frac{\partial s_C}{\partial [C]} \\
\end{array}
\]

Sparse Jacobian:

NVAR: 3  LU_NONZERO: 8  
LU_IROW: 1 1 1 2 2 3 3 3  
LU_ICOL: 1 2 3 2 3 1 2 3

(b) Reduced Mechanism (B is slow)

\[
\begin{array}{ccc}
A & B & C \\
A & \frac{\partial s_A}{\partial [A]} & \frac{\partial s_A}{\partial [C]} \\
B & \frac{\partial s_B}{\partial [B]} & \frac{\partial s_B}{\partial [C]} \\
C & \frac{\partial s_C}{\partial [A]} & \frac{\partial s_C}{\partial [C]} \\
\end{array}
\]
We implement the adaptive solver while reusing KPP’s sparse linear algebra structures, avoiding expensive memory re-allocation.

\[
J_{\text{full}} = \begin{pmatrix}
\frac{\partial s_A}{\partial [A]} & \frac{\partial s_A}{\partial [B]} & \frac{\partial s_A}{\partial [C]} \\
\frac{\partial s_B}{\partial [A]} & 0 & \frac{\partial s_B}{\partial [C]} \\
\frac{\partial s_C}{\partial [A]} & \frac{\partial s_C}{\partial [B]} & \frac{\partial s_C}{\partial [C]}
\end{pmatrix}
\]

\[
J_{\text{reduced}} = \begin{pmatrix}
\frac{\partial s_A}{\partial [A]} & \frac{\partial s_A}{\partial [C]} \\
\frac{\partial s_C}{\partial [A]} & \frac{\partial s_C}{\partial [C]}
\end{pmatrix}
\]

Sparse Jacobian:

- NVAR: 3
- LU_NONZERO: 8
- LU_IROW: [1 1 1 2 2 3 3 3]
- LU_ICOL: [1 2 3 2 3 1 2 3]

Pre-allocated Sizes (Good!)

Reduced mechanism With same data structure and memory space

- rNVAR: 2
- cNONZERO: 4
- JVS_MAP: [1 3 6 8]
- cLU_IROW: [1 1 2 - - - - - -]
- cLU_ICOL: [1 2 1 - - - - - -]
We implement this adaptive solver within GEOS-Chem via KPP, with a tropospheric error of \(~1\%\) for key species and \(~32\%\) reduction in solver time.

Most species have \(<1\%\) errors in the troposphere, except in halogen species. Halogen species can be kept within the fast sub-mechanism if needed.
Error accumulation is minimal over 1-year test runs but higher in the strat.
Take-home messages

• An adaptive solver to reduce chemical mechanisms on-the-fly based on local conditions is implemented in GEOS-Chem through KPP.

• The adaptive solver achieves ~1% error for key species in the troposphere while reducing integration time by ~32%.

• The adaptive solver and other improvements to KPP are now available in KPP version 3.0.0, maintaining previous features and with zero numerical differences compared to previous version 2.1
Here be dragons.

No slides beyond this point
Explicit approximation with first-order loss rate constant $k_i(t_0) = L_i(t_0)/C_i(t_0)$

$$C_i(t_0 + \Delta t) = \frac{P_i(t_0)}{k_i(t_0)} + \left( C_i(t_0) - \frac{P_i(t_0)}{k_i(t_0)} \right) e^{-k_i(t_0)\Delta t}$$
Mass conservation of odd $O_x$, reactive N, Br, Cl atoms is indeed ‘inconsequential’ (<0.3% globally)