

20 Abstract

21 Kinetic integration of large and stiff chemical mechanisms is a computational bottleneck in models of
22 atmospheric chemistry. It requires implicit solution of the coupled system of kinetic differential equations
23 with time-consuming construction and inversion of the Jacobian matrix. We present here a new version of
24 the Kinetic Pre-Processor (KPP 3.0.0) for fast integration of chemical kinetics featuring a range of
25 improvements over previous versions in performance, diagnostics, versatility, and community openness.
26 KPP 3.0.0 includes a new adaptive auto-reduction solver to decrease the size of any mechanism locally and
27 on the fly under conditions where full complexity is not needed, by partitioning species as “fast” or “slow”
28 based on their local production and loss rates. Previous implementations of this adaptive solver suffered
29 from excessive overhead in the repeated construction of the local Jacobian matrix or were hard-wired to
30 specific mechanisms. Here we retain the general applicability of the method to any mechanism and avoid
31 overhead by using pre-computed Jacobian matrix terms for the full mechanism and cropping the matrix
32 locally to remove the slow species with no change in memory allocation. We apply this adaptive solver
33 within KPP 3.0.0 to the GEOS-Chem global 3-D model of atmospheric chemistry and demonstrate a 32%
34 reduction in solver time while maintaining a mean error lower than 1% for key species in the troposphere.

35

36 Plain Language Summary

37 Calculating chemical evolution in global atmospheric chemistry models is computationally expensive
38 because the chemical mechanisms typically include hundreds of species to account for all conditions from
39 urban to remote. However, the full chemical complexity is not needed under most conditions. Here we have
40 developed an adaptive auto-reduction chemical solver that reduces any mechanism on the fly depending on
41 local conditions and without significant computational overhead. We apply this adaptive solver as an option
42 in a new version 3.0.0 of the Kinetic Pre-Processor (KPP) chemical solver software package that also
43 includes a number of updates relative to previous versions. The adaptive solver achieves a 32% reduction
44 in solver time in a global model simulation while incurring less than 1% average errors for key species.

45 **1 Introduction**

46 Modeling atmospheric chemistry is a grand computational challenge. Current global 3-D models of oxidant-
47 aerosol chemistry use chemical mechanisms that may involve hundreds of coupled chemical species with
48 lifetimes ranging from less than a second to many years. Chemical evolution in such a mechanism is
49 computed by solving a large, stiff system of coupled non-linear ordinary differential equations (ODEs)
50 expressing the chemical kinetics of individual species. Implicit solvers are required to accommodate the
51 coexistence of short and long time constants but are computationally expensive because of the need for
52 repeated construction and inversion of the Jacobian matrix (Brasseur & Jacob, 2017). Chemical integration
53 often dominates the overall computational cost of global 3-D atmospheric chemistry model simulations,
54 even in massively parallel environments or using graphics processing units (GPUs) (Alvanos &
55 Christoudias, 2017; Eastham et al., 2018; Zhuang et al., 2020; Dawson et al., 2022).

56
57 Considerable research has gone into devising algorithms to speed up chemistry solvers. A common strategy
58 is to split the mechanism species by time scales in order to decrease the stiffness of the system (Young &
59 Boris, 1977; Gong & Cho, 1993; Djouad & Sportisse, 2002), but this tends to be mechanism-specific and
60 is difficult to apply in global models because of the wide range of conditions that may be experienced.
61 Wholesale reduction of the mechanism, such as in the Super-Fast mechanism used in some climate models
62 (Brown-Steiner et al., 2018), may lead to large errors (Kelp et al., 2022) and incorrect chemical responses
63 to perturbations. Machine learning methods can in principle speed up chemical integration by orders of
64 magnitude but have met with little success because of the large dimensionality of the problem resulting in
65 error growth (Keller & Evans, 2019; Kelp et al., 2020, 2022).

66
67 A promising approach for global models is to recognize that the full complexity of the mechanism is not
68 needed everywhere. For example, reactions involving volatile organic compounds (VOCs) and their short-
69 lived products typically account for much of the complexity but may be unimportant outside of continental
70 boundary layers where they are emitted. Jacobson (1995) thus applied separate mechanisms in a global
71 model for the urban boundary layer, the global troposphere, and the stratosphere. However, this approach
72 can result in errors and inefficiencies by not accounting for the interactions at chemical boundaries between
73 domains (Rastigejev et al., 2007) and not allowing for a continuum of chemical regimes from source regions
74 to the remote atmosphere. Santillana et al. (2010) developed an adaptive mechanism reduction method in
75 which the size of the mechanism is adjusted at each grid cell and time step, classifying species as fast
76 (coupled) or slow (uncoupled) on the basis of their total production and loss rates. However, the overhead
77 involved in local definition of the reduced mechanism offset the computational gains. Sander et al. (2019)

78 and Shen et al. (2020, 2022) improved the method by pre-compiling a limited ensemble of chemical sub-
79 mechanisms. Sub-mechanisms were selected based on local conditions at each timestep, thus avoiding the
80 overhead required to define sub-mechanisms. Shen et al. achieved a 30%-50% reduction in computational
81 cost compared to the full parent mechanism in a global simulation of the troposphere and stratosphere, but
82 the selection of sub-mechanisms had to be customized to the parent mechanism.

83
84 Here we develop a mechanism-agnostic, ready-to-use method for adaptive auto-reduction of any chemical
85 mechanism and implement it as an option in a new version 3.0.0 of the Kinetic PreProcessor (KPP). KPP,
86 originally developed by Damian et al. (2002) and Sandu and Sander (2006), is a software tool that
87 automatically generates code to efficiently integrate chemical mechanisms. KPP takes in a set of human-
88 readable input files describing the mechanism species and reactions and generates Fortran 90, C, or
89 MATLAB code solving the corresponding system of ODEs using one of a suite of integration methods.
90 KPP is used to generate the chemistry solver source code in several atmospheric chemistry models including
91 MECCA within MESSy (Sander et al., 2019; Jöckel et al., 2010), WRF-Chem (Grell et al., 2005; Fast et
92 al., 2006), the forward and adjoint GEOS-Chem models (Henze et al., 2007), and the adjoint for the CMAQ
93 model (Hakami et al., 2007; Zhao et al., 2020). Our new version KPP 3.0.0 incorporates several
94 performance and diagnostic updates over the previous version KPP 2.1 (Sandu & Sander, 2006), in addition
95 to the adaptive solver option presented below.

96 **2 Adaptive solver for chemical kinetics**

97 Atmospheric chemistry models alternate chemical integration and transport calculations through operator
98 splitting (Brasseur & Jacob, 2017), The chemistry solver is called for a time interval of length Δt , referred
99 to as the external time step, and returns a vector of updated concentrations \mathbf{C} at the end of that time step to
100 be operated on by transport. The kinetic integration of the mechanism by the chemistry solver is done over
101 internal time steps $\delta t \leq \Delta t$ to reach the desired accuracy.

102

103 The chemistry solver integrates a system of N coupled nonlinear first-order ODEs of the form

$$104 \quad \frac{dC_i}{dt} = P_i(\mathbf{C}) - L_i(\mathbf{C}) \quad (i = 1, 2, \dots, N) \quad (1)$$

105 where N is the number of coupled species in the mechanism, \mathbf{C} is the vector of species concentrations of
106 dimension N , and $P_i(\mathbf{C})$ and $L_i(\mathbf{C})$ are the production and loss rates of species i that depend on the
107 concentrations of other species in the mechanism through the reaction rate expressions.

108

109 The external time step in a global model is typically $\sim 10^3$ s, but many species in the mechanism have
 110 lifetimes ~ 1 s or shorter. An explicit solver would require internal time steps shorter than the lifetime of the
 111 shortest-lived species in order to achieve stability, but this is not computationally practical. An implicit
 112 solver is required. The simplest such solver is the first-order backward Euler method, which approximates
 113 the solution to (1) over the internal time step δt with

$$114 \quad \mathbf{C}(t + \delta t) = \mathbf{C}(t) + \mathbf{s}(\mathbf{C}(t + \delta t))\delta t \quad (2)$$

115 where we define the net source term $\mathbf{s} = \mathbf{P} - \mathbf{L}$ as a vector of functions ($P_i - L_i$). Solving (2) for the
 116 unknown quantity $\mathbf{C}(t + \delta t)$ using the Newton-Raphson method requires the repeated construction and
 117 inversion of the $N \times N$ Jacobian matrix \mathbf{J} :

$$118 \quad \mathbf{J} = \frac{\partial \mathbf{s}}{\partial \mathbf{C}} \quad (3)$$

119 The construction and inversion of this Jacobian matrix is computationally expensive. Higher-order solvers
 120 generally used in atmospheric chemistry models, such as Rosenbrock (Rosenbrock, 1963; Hairer & Wanner,
 121 1991; Sandu et al., 1997) or Gear (Jacobson & Turco, 1994), similarly require the repeated construction
 122 and inversion of the Jacobian over internal time steps. The Jacobian is typically $\sim 90\%$ sparse allowing for
 123 efficient sparse-matrix inversion methods (Sandu et al., 1997), so that the overall cost of construction and
 124 inverting the Jacobian scales as $\sim N$ rather than a higher power.

125
 126 A way to reduce the dimensionality N of the problem is to split the mechanism into “fast” species for which
 127 the coupled implicit solution is necessary and “slow” species that may be solved independently over the
 128 external time step using a fast explicit method. Young and Boris (1977) and Gong and Cho (2003) separate
 129 species into fast and slow based on their lifetimes compared with the integration time step. However, the
 130 separation results in non-conservation of mass because the reaction rates are not computed consistently.
 131 This may not be of consequence in a regional model (as used in those applications) where the domain is
 132 ventilated by the boundary conditions, so that errors do not accumulate, but it is more problematic in a
 133 global model. Santillana et al. (2010) separated instead “fast” and “slow” species on the basis of their
 134 production and loss rates, with the slow species having sufficiently low rates that their influences on other
 135 species would be negligible and the non-conservation of mass would be inconsequential. This is more
 136 relevant for global models where concentrations and rates of VOCs become very small outside of their
 137 source regions. Shen et al. (2020, 2022) used the same approach to partition species between fast and slow.

138
 139 Here we also follow the partitioning method of Santillana et al. (2010). At the beginning of each external
 140 time step we calculate \mathbf{P} and \mathbf{L} and classify species i as fast if $\max(P_i, L_i) > \delta$ and as slow otherwise,
 141 where δ is a user-selected partitioning threshold. Fast species are assigned to the coupled implicit solver as

142 a subset (sub-mechanism) of the full mechanism, while the evolution of slow species over the external time
 143 step is calculated using an explicit first-order approximation with first-order loss rate constant $k_i(t_0) =$
 144 $L_i(t_0)/C_i(t_0)$:

$$145 \quad C_i(t_0 + \Delta t) = \frac{P_i(t_0)}{k_i(t_0)} + \left(C_i(t_0) - \frac{P_i(t_0)}{k_i(t_0)} \right) e^{-k_i(t_0)\Delta t} \quad (4)$$

146 In the Santillana et al. (2010) implementation, the Jacobian had to be reconstructed locally at every external
 147 time step for the identified subset of fast species and that incurred large overhead, canceling the benefit of
 148 the method. Here we avoid the overhead by taking advantage of the pre-computed Jacobian matrix terms
 149 for the full mechanism in the KPP solver to simply remove rows and columns corresponding to the slow
 150 species. This is explained in Section 3.2 and is the key new development to make the method
 151 computationally practical.

152
 153 The partitioning threshold δ is set prior to integration and is tuned to balance performance and accuracy.
 154 Previous work (Santillana et al., 2010; Shen et al, 2020, 2022) considered that in a typical tropospheric
 155 chemistry mechanism, much of the coupling is associated directly or indirectly with cycling of the hydroxyl
 156 radical (OH). OH has a daytime concentration of $\sim 10^6$ molecules cm^{-3} and a lifetime ~ 1 s, so its production
 157 and loss rates are $\sim 10^6$ molecules $\text{cm}^{-3} \text{ s}^{-1}$. A species with production and loss rates that are several orders
 158 of magnitude smaller would not be expected to contribute significantly to the coupling. They found
 159 $\delta \sim 10^2 - 10^3$ molecules $\text{cm}^{-3} \text{ s}^{-1}$ to be adequate after testing for performance and accuracy.

160
 161 Here we include the option to dynamically define δ instead of specifying a uniform value over the domain,
 162 to account for rates varying with local conditions. This is done by identifying a target species which is
 163 considered central to the mechanism and scaling its production and loss rates to define a local partitioning
 164 threshold:

$$165 \quad \delta = \alpha_{\text{target}} \max(P_{\text{target}}, L_{\text{target}}) \quad (5)$$

166 where P_{target} and L_{target} are the local production and loss rates of the target species, and $\alpha_{\text{target}} \ll 1$ is
 167 a user-selected coefficient that depends on the target species but is otherwise fixed for the model domain.
 168 For example, a model may use OH as target species for daytime and NO_2 for nighttime. When using OH as
 169 a target species and with $\max(P_{\text{OH}}, L_{\text{OH}}) \sim 10^6$ molecules $\text{cm}^{-3} \text{ s}^{-1}$, a value $\alpha_{\text{OH}} \sim 10^{-4} - 10^{-3}$ would
 170 correspond to the criteria for δ used by Santillana et al. (2010). But $\max(P_{\text{OH}}, L_{\text{OH}})$ can in fact vary over
 171 orders of magnitude depending on pressure, UV flux, and other factors, and our local specification of δ
 172 accounts for this variability.

173

174 We also include two new options in the algorithm. First is to force individual species to remain in the
175 coupled implicit solver even if $\max(P_i, L_i) \ll \delta$. As we will see, this may be helpful for inorganic halogen
176 species that cycle between radical and non-radical forms across sunrise/sunset. Second is to include an
177 ‘append’ functionality in the algorithm so that a species initially diagnosed as slow at the beginning of the
178 external time step can be transferred into the coupled implicit solver if it becomes fast over the course of
179 the integration. This increases accuracy with minimum overhead.

180 **3 Kinetic Pre-Processor (KPP) version 3.0.0**

181 **3.1 KPP 3.0.0 overview**

182 The Kinetic Pre-Processor (KPP, Damian et al., 2002; Sandu & Sander, 2006) generates code for solving
183 the chemical kinetics for a given chemical mechanism defined by a list of species, reactions, and rate
184 constants. It is designed for speed by exploiting sparse matrix algebra and pre-computation of terms. We
185 have made several improvements to KPP relative to the previous version 2.1 (Sandu & Sander, 2006),
186 including the adaptive solver option. We present these improvements as KPP version 3.0.0, available for
187 download from <https://github.com/KineticPreProcessor/KPP/> (DOI: 10.5281/zenodo.6828026) with
188 detailed documentation at <https://kpp.readthedocs.io>.

189
190 KPP takes in as input one or more text files, with an example shown in Figure 1. The input is not necessarily
191 in one single file, and may be split into multiple files for readability. The files describe the chemical
192 mechanism, the choice of the numerical solver (e.g., Rosenbrock), target language (e.g., Fortran 90, C, or
193 MATLAB), floating point type (single- or double-precision), production and loss diagnostics for selected
194 chemical families (optional), and whether to use the adaptive solver option (optional). Some inputs are not
195 part of the KPP code generation and are instead left for users to adjust at runtime, including convergence
196 criteria (absolute and relative tolerance), numerical order (e.g., order of the Rosenbrock solver), and
197 adaptive solver options, in order to enable the user to experiment with different thresholds for performance
198 and accuracy.

```

#INTEGRATOR  rosenbrock_autoreduce
#LANGUAGE    Fortran90
#MINVERSION  3.0.0
#AUTOREDUCE  on
#FAMILIES
POx : O3 + NO2 + HNO3;
LOx : O3 + NO2 + HNO3;

#DEFVAR
O3   = IGNORE;
NO   = IGNORE;
...

#EQUATIONS
O3 + hv = 2OH + O2 : PHOTOL(1);
NO + O3 = NO2 + O2 : ARR(1.8d-12, -1370.0);
CO + OH = CO2 + H2O : 2.4d-13;
H2 + NO = OH + NO2 : ARR(3.7d-12, 240.0);
NO2 + hv = NO + O3 : PHOTOL(2);
H2O2 + hv = 2OH : PHOTOL(3);
CH4 + OH = CO + H2O : ARR(3.1d-12, -187.0);
...

```

KPP Options
Integrator, target language, and other code generation options

Production and Loss Families (Optional)

Definition of species
"IGNORE" if mass balance checks are not required
Otherwise specify stoichiometric composition

List of reactions
Reactants = Products : Rate constant
Fortran 90 expressions are supported in rate constants

199

200 **Figure 1. Example KPP input file.** The KPP input file includes options for code generation, adaptive auto-
 201 reduction, diagnostics for production and loss rates of chemical families, and a list of species and reactions.
 202 Reaction rate constants can be specified as expressions in the target language (in this case Fortran 90), here
 203 showing calls to functions calculating photolysis and Arrhenius rate expressions.

204

205 Based on the specifications in the KPP input file(s), KPP creates files in the target language containing a
 206 description of the system of ODEs (number of species, reactions, and a numbered list of species) code to
 207 calculate the time derivatives, Jacobian matrix, and solution by back-substitution, along with a copy of the
 208 numerical solver (such as Rosenbrock) and supporting routines for sparse linear algebra. This set of files
 209 can be either run standalone as a box model or can be included in a 3-D model to update concentrations
 210 locally over external time steps.

211

212 Auto-reduction of the mechanism with the adaptive solver described in Section 2 is an option in KPP 3.0.0
 213 using the Fortran 90 language, enabled in the configuration file by #AUTOREDUCE on and specifying the
 214 corresponding integrator (e.g., #INTEGRATOR rosenbrock_autoreduce). This sets up the
 215 capability for the user to reduce the mechanism locally through specification of the partitioning threshold
 216 δ between fast and slow species, listing any species for which this partitioning should not be applied. These
 217 specifications are done at runtime for flexibility. Even when the auto-reduction solver is used, mechanism
 218 auto-reduction can be disabled at runtime, defaulting to the behavior of the original integrator. A test case
 219 box model for auto-reduction is included as part of KPP 3.0.0 documentation.

220

221 In addition to the option for adaptive mechanism auto-reduction, several additional features and
 222 improvements were added to KPP 3.0.0. These include:

223

224 **a. Redeployment of KPP source code, continuous integration, and documentation for community**
 225 **development.** KPP 3.0.0 source code has been redeployed on GitHub for community development.
 226 The GitHub repository incorporates continuous integration (CI) tests which automatically compile the
 227 KPP source code to build and run combinations of sample chemistry mechanisms and integrators into
 228 box models at every code revision. This helps to ensure that new features and updates added to KPP do
 229 not break existing functionality. The documentation has also been relocated to
 230 <https://kpp.readthedocs.io> where it is automatically built with each code revision on GitHub.

231

232 **b. Diagnostic improvements.** Diagnostic features have been added to KPP 3.0.0:

- 233 • **Production and loss rates for chemical families.** KPP 3.0.0 allows for the definition of families of
 234 chemical species for computing production and loss for that family, ignoring interconversion
 235 reactions within the family. Families are defined in the #FAMILIES section of the KPP input file.
- 236 • **Stoichiometric numbers.** The stoichiometric numbers of all reactions in the mechanism are now
 237 available in the CalcStoichNum subroutine in the KPP-generated code. This feature is used to
 238 calculate the importance of chemical species in a mechanism for the skeletal mechanism reduction
 239 in Sander et al., 2019.
- 240 • **Individual reaction rates and time derivatives.** The reaction rates and time derivatives are now
 241 available as optional outputs from the KPP-generated code.

242

243 **c. Addition of new solvers.** Several solvers have been added as options for integrating chemical kinetics
 244 including VODE (Brown et al., 1989), SDIRK (Hairer & Wanner, 1991), 3-stage Runge-Kutta, and
 245 forward and backward Euler methods.

246

247 **d. Addition of new rate law functions.** Rate law functions for three-body reactions using the formulas
 248 proposed by JPL (<https://jpldataeval.jpl.nasa.gov>) and IUPAC (<https://iupac.aeris-data.fr/>) have been
 249 added to the built-in rate laws in KPP 3.0.0. Rate law functions are not limited to those built in KPP
 250 and can be added by including extra source code files in the KPP input.

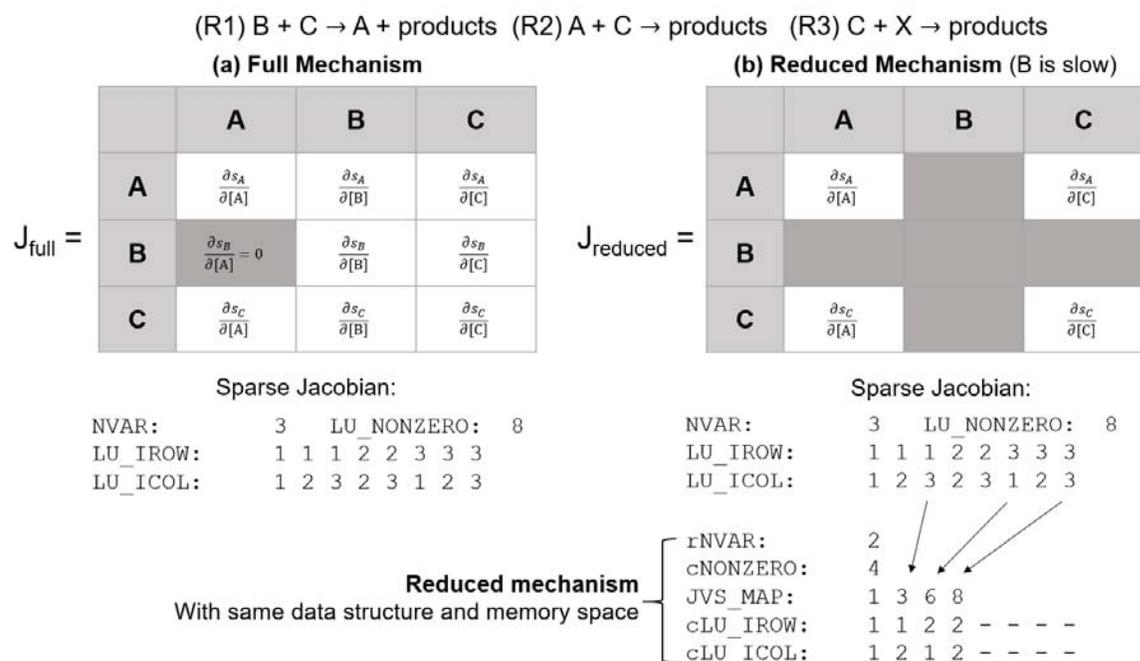
251

252 **e. Miscellaneous performance improvements in Fortran 90.** KPP 3.0.0 optimizes for Fortran 90
 253 performance by applying several guidelines in coding, particularly in reaction rate calculations that are
 254 computed repeatedly in loops:

- 255
- 256 • **Unifying number precision.** Previous inputs to KPP used both single- and double-precision
257 numbers. KPP input files and code now do not mix number precision to avoid conversion, which
258 loses precision and costs computational time.
 - 259 • **Switch to control update of reaction rate constants.** Previously, KPP called subroutines to update
260 reaction rate constants at every internal time step, but this is computationally expensive and may
261 not be needed in most model applications, because the variables affecting rate constants are not
262 updated between internal time steps in these models. An optional switch has been added so rate
263 constants can be updated only at the beginning of the external time step.
 - 264 • **Optimized rate law functions.** Rate law functions have been split to avoid computation of
265 unnecessary terms. For example, previously a single function with Arrhenius temperature
266 dependence was used for all reactions: $ARR_{abc}(A, B, C) = A * \exp(B/T) * (300/T)^C$. However,
267 certain reactions may have $B = 0$ or $C = 0$, but in this case the expressions $300/T$ or $\exp(0)$ are
268 still computed, leading to the waste of CPU cycles. Separate rate functions where coefficients lead
269 to constants, such as $ARR_{ab}(A, B) = A * \exp(B/T)$ have been added to skip computations that
270 can be pre-evaluated to constants, leading to a 44% performance improvement of reaction rate
271 computations in a full-chemistry GEOS-Chem model run.
 - 272 • **Avoiding conditionals and optional arguments.** Conditional clauses such as IF, ELSE, and
273 SELECT CASE, and testing for optional arguments add significant computational cost if called
274 thousands of times. If a conditional clause or optional argument is present in a frequently called
275 subroutine, the subroutine is split into different functions for each case.
 - 276 • **Thread safety for generated code.** The code generated is now thread safe, so that calls for
277 updating rate constants and running the integrator can be placed in an OpenMP parallel loop for
278 parallelization.
 - 279 • **Improved expressions for vector and array functions.** Several functions for basic vector and
280 array operations originally used reference BLAS (Basic Linear Algebra Subprograms)
281 implementations. These have been replaced with Fortran 90 expressions to allow for compilers to
282 better optimize the code.
- 283 **f. Miscellaneous code improvements in KPP.** The C source code of KPP has been improved so that no
284 compiler warnings are generated. A more consistent memory allocation helps to avoid buffer overflow
285 problems. The KPP language is now parsed by `bison` instead of `yacc`.

286 3.2 Adaptive solver implementation in KPP 3.0.0

287 We implemented the adaptive solver as an option within KPP's Fortran 90 version of the Rosenbrock solver
 288 using sparse matrix algebra. KPP is computationally efficient because the functions to compute the time
 289 derivatives for each species and the Jacobian matrix (expressed in terms of reaction rates and species
 290 concentrations), along with all sparse matrix algebra routines, are pre-generated and use fixed indices to
 291 access species vectors and matrices. This means that the problem size and memory space is fixed at compile
 292 time, avoiding expensive memory allocation operations. However, this also yields a fixed problem structure
 293 that is difficult to manipulate. One major source of overhead in locally defined sub-mechanisms as in
 294 Santillana et al. (2010) is associated with repeatedly re-allocating and de-allocating memory to
 295 accommodate changing problem sizes for each sub-mechanism (Shen et al., 2020). Our adaptive solver
 296 implementation in KPP 3.0.0 uses a mapping operation to project the full mechanism into sub-mechanisms,
 297 thus reusing the same memory space to avoid expensive resizing operations.



298
 299 **Figure 2. Mechanism auto-reduction in KPP.** Panel (a) shows the data structure of the sparse Jacobian
 300 data within KPP for a sample 3-species mechanism. If species B is diagnosed as slow, then the
 301 corresponding rows and columns of the Jacobian are no longer calculated (panel (b)), and the indices
 302 pointing to the sparse Jacobian data (LU_IROW, LU_ICOL) are adjusted to remove the slow species
 303 (cLU_IROW, cLU_ICOL) through a mapping array (JVS_MAP) while preserving the sparse matrix in
 304 row-compressed form. The result is a reduced mechanism with the same data structure as the original one,
 305 but with smaller dimensions.

306

307 Figure 2(a) shows the sparse Jacobian data as stored within KPP. In this example mechanism, there are 3
308 species (NVAR) and 8 non-zero entries in the Jacobian (LU_NONZERO) of the full mechanism. The row and
309 column indices of these 8 non-zero entries in Jacobian matrix are correspondingly specified in LU_IROW,
310 LU_ICOL in row-compressed form. At the beginning of every external time step, the production and loss
311 rates of each species are calculated and compared to the partitioning threshold to separate species into fast
312 and slow. Figure 2(b) shows an example where species B is slow. The entries in the Jacobian corresponding
313 to B no longer need to be computed, and a mapping operation is performed: JVS_MAP corresponds to the
314 non-zero Jacobian matrix entries still present in the sub-mechanism consisting of fast species. The smaller
315 sub-mechanism can now be described by 2 species (rNVAR) and 4 non-zero entries in the reduced Jacobian
316 (cNONZERO) with indices described by cLU_IROW and cLU_ICOL. The data structure of the smaller sub-
317 mechanism is identical to the full mechanism, and the same routines are used to solve it, without the need
318 to generate extra code, or resizing memory.

319
320 The mechanism auto-reduction is performed once at the beginning of every external time step. The set of
321 fast and slow species are established according to the runtime options for the partitioning threshold δ and
322 the list of species to be excluded from partitioning and kept in the fast subset under all conditions
323 (keepSpActive). Based on the list of species in the fast set, the mapping (JVS_MAP) from the full
324 mechanism to the fast sub-mechanism is created. Because KPP generates hard-coded source code to
325 compute each term of the Jacobian matrix and back-substitution for computational efficiency, two logical
326 control vectors, DO_JVS and DO_SLV, are created to skip computation of terms corresponding to slow
327 species as these are no longer necessary.

328
329 The separation between fast and slow species is controlled by the initial conditions at the beginning of the
330 external time step, but an optional “append” function is added to account for an initially slow species
331 becoming fast over the course of the internal time steps. This function appends new species to the fast sub-
332 mechanism and adjusts the logical control vectors if these species are initially partitioned as slow but their
333 production or loss rate exceed the partitioning threshold over the course of the internal time stepping.
334 Diagnosing this has little overhead, because the production and loss rates of all species are already
335 computed at every internal time step.

336
337 We used a box model integration of the GEOS-Chem chemical mechanism (described below) to analyze
338 the overhead of the adaptive solver implemented within KPP. By forcing the adaptive solver routines to run
339 with a threshold of 0, we determined that the KPP overhead added by the auto-reduction is 10-16%,
340 depending on the initial condition provided to the box model. The main source of overhead is the copying

341 of data between the full and reduced sub-mechanism (Figure 2), where the worst-case scenario is when all
342 species are partitioned as fast and all species' data need to be copied between the full to the "reduced" data
343 structures. Profiling tests show that other steps such as the partitioning of species between fast and slow, or
344 the first-order approximation for slow species, contribute negligible overhead.

345 **4 Adaptive mechanism auto-reduction in the GEOS-Chem model using KPP version 3.0.0**

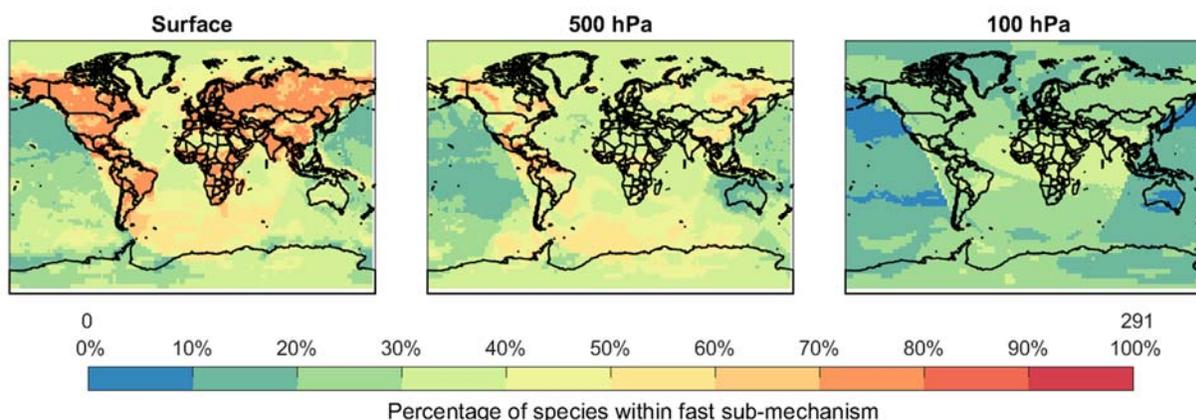
346 We demonstrate the adaptive solver capability in KPP version 3.0.0 with the GEOS-Chem global 3-D
347 atmospheric chemistry model version 13.4.0 (Bey et al., 2001; <https://doi.org/10.5281/zenodo.6511970>).
348 The chemical mechanism includes comprehensive oxidant-aerosol chemistry in the troposphere and the
349 stratosphere with 291 chemical species and 913 reactions. Recent updates to the mechanism include Cl-Br-
350 I tropospheric halogen chemistry (Wang et al., 2021), isoprene chemistry (Bates & Jacob, 2019), aromatic
351 chemistry (Bates et al., 2021), hydroxymethanesulfonate cloud chemistry (Moch et al., 2020), and NO_y
352 cloud and aerosol chemistry (Holmes et al., 2019). Heterogeneous sulfur chemistry that was previously
353 simulated with a separate module was brought into KPP in version 13.4.0 with the addition of new rate
354 functions.

355
356 GEOS-Chem has been structured to interact with the KPP-generated solver code through the FlexChem
357 interface which prepares data for the solver code, executes the code, and retrieves concentrations at the end
358 of the time step. FlexChem allows GEOS-Chem to use modules outside of KPP for computing reaction
359 rates, for example interfacing with Fast-JX for photolysis rates (Bian & Prather, 2002; Mao et al., 2010).
360 FlexChem also includes a derived type object, `State_Het`, which passes state variables from GEOS-
361 Chem model for calculating heterogeneous chemistry reaction rates including cloud liquid water content,
362 aerosol size distribution, pH, and/or alkalinity. This derived type object holds common intermediate
363 quantities necessary for heterogeneous reaction rate computations, such as aerosol area, avoiding repeated
364 computation and memory use. FlexChem also allows GEOS-Chem users to modify the chemical
365 mechanism input into KPP without modifying the GEOS-Chem source code.

366
367 We evaluate the accuracy and computational performance of the adaptive solver in KPP version 3.0.0 using
368 global GEOS-Chem simulations at 2°×2.5° resolution with 72 vertical levels extending up to 0.1 hPa. The
369 model is driven by the Modern-Era Retrospective analysis for Research and Applications, Version 2
370 (MERRA-2) meteorological fields from the NASA Global Modeling and Assimilation Office (GMAO).
371 The external time step for the chemistry solver is 20 minutes. All simulations are conducted on the same
372 single-node hardware with 24 Intel Cascade Lake physical cores (Intel(R) Xeon(TM) Platinum 8268 CPUs

373 with a base clock speed of 2.90GHz, no hyper-threading logical cores), 100 GB of RAM, and a high-
 374 performance Lustre parallel file system. The model was compiled using Intel(R) Fortran Compiler (ifort)
 375 version 2021.2.0.

376
 377 We select as the standard configuration of the adaptive solver within the GEOS-Chem model a dynamically
 378 defined threshold with OH as target species during daytime, and NO₂ during nighttime, with coefficients
 379 $\alpha_{OH} = 5 \times 10^{-5}$ and $\alpha_{NO_2} = 1 \times 10^{-4}$. We do not force any species to remain in the implicit KPP solver
 380 as fast and we do not use the append functionality. We find that we achieve a net 32% reduction in
 381 integration time in this standard configuration.



382
 383 **Figure 3. Percentage of GEOS-Chem species retained in the fast sub-mechanism when using the**
 384 **adaptive solver.** The full GEOS-Chem mechanism has 291 species to describe tropospheric and
 385 stratospheric oxidant-aerosol chemistry. Only a fraction of species is retained as fast in the KPP solver,
 386 while the other slow species are solved individually using equation (4). Results are shown for a snapshot in
 387 time on July 1, 2014, 12:00 UTC at different altitudes. The adaptive solver uses a dynamically defined
 388 threshold (equation (5)) with target species OH in daytime ($\alpha_{OH} = 5 \times 10^{-5}$) and NO₂ at night ($\alpha_{NO_2} =$
 389 1×10^{-4}).

390
 391 Figure 3 shows the percentage of species retained in the fast sub-mechanism using the adaptive solver's
 392 standard configuration. Over 60% are retained in surface air over land, reflecting VOC sources, whereas
 393 only 10-50% are needed over the ocean. The fraction of retained species decreases with altitude, and fewer
 394 than 40% are needed in the stratosphere where VOC chemistry is mainly limited to methane. Fewer species
 395 are needed at night than in daytime. These results are consistent with Shen et al. (2020).
 396

397 We evaluate the accuracy of the adaptive solver (AS) relative to the full mechanism solver over the global
 398 GEOS-Chem domain using the relative root mean squared (RRMS) error metric. For each species i , the
 399 RRMS error is:

$$400 \quad RRMS_i = \sqrt{\frac{1}{N_i} \sum_{j=1}^{N_i} \left(\frac{C_{i,j,\text{full}} - C_{i,j,\text{AS}}}{C_{i,j,\text{full}}} \right)^2} \quad (6)$$

401 where $C_{i,j,\text{full}}$, $C_{i,j,\text{AS}}$ are the concentrations of species i in grid box j for simulations without and with the
 402 adaptive solver. The RRMS is computed over the ensemble N_i of ordered grid boxes that account for 99%
 403 of the total mass of species i in the boundary layer (surface to PBL height from MERRA2), free troposphere
 404 (boundary layer height to tropopause), and stratosphere, respectively, and where $C_{i,j,\text{full}}$ is greater than 10
 405 molecules cm^{-3} .

406
 407 Figure 4(a) shows the mean errors at the end of a 1-year simulation for species in different chemical
 408 categories (Table S1) and Figure 4(b)-(d) shows the time evolution of errors over the 1-year simulation for
 409 all species in the standard configuration of the adaptive solver within the GEOS-Chem model. The mean
 410 error for most categories is below 1% within the boundary layer and the free troposphere. There is no error
 411 growth in the troposphere but there is some in the stratosphere. Figure 5 shows the geographical distribution
 412 of errors for ozone, OH, and NO_2 at the surface and 500 hPa for the end of the 1-year simulation. Errors are
 413 generally lower than 1% except for OH at high latitudes and NO_2 over the Southern Ocean, where errors
 414 are up to 3% for OH and 6~10% for NO_2 . Errors over land for these species are minimal.

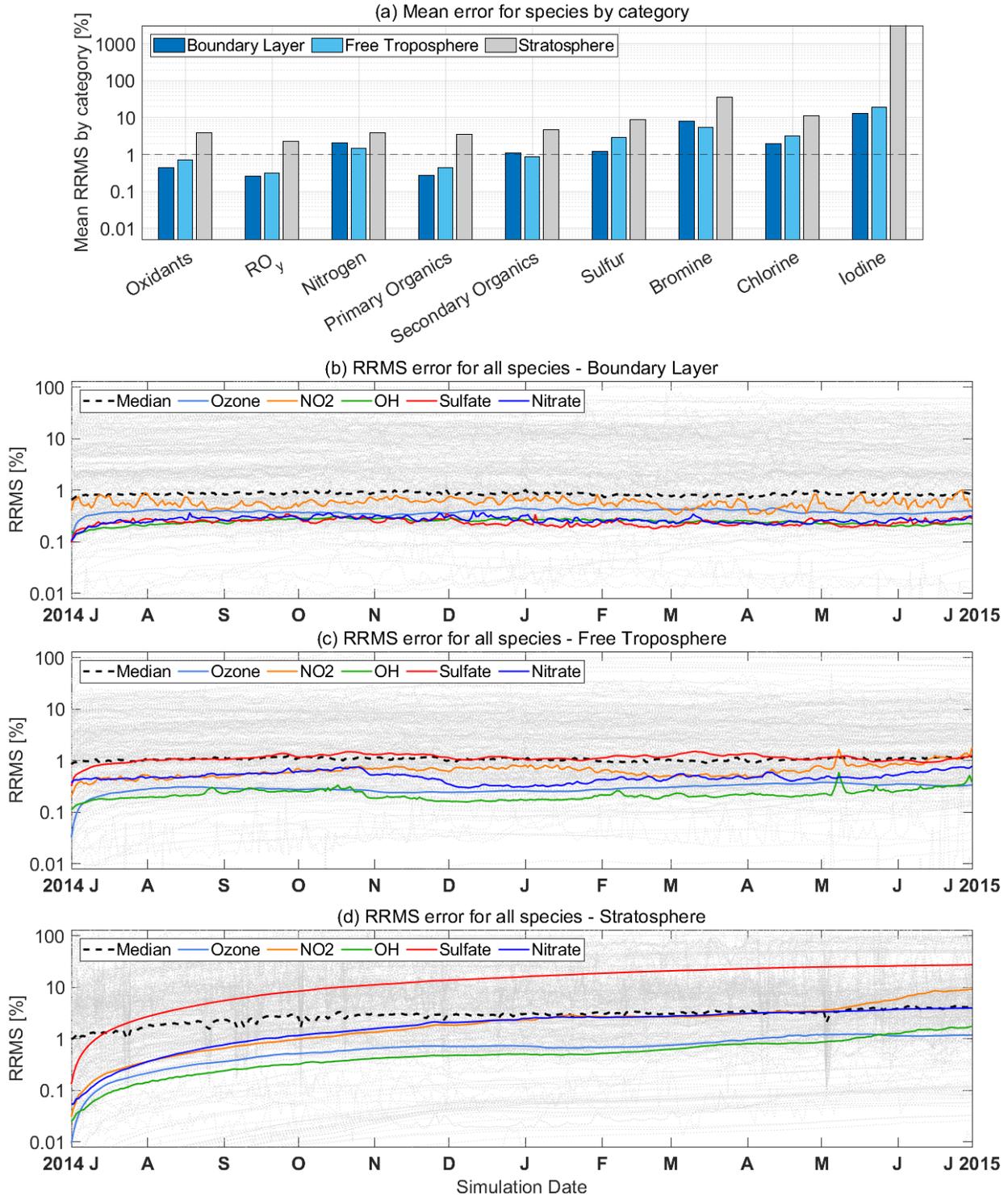
415
 416 The largest errors in Figure 4 are found for inorganic halogen radicals and their reservoirs. Halogens cycle
 417 through species with lifetimes spanning several orders of magnitude, and include several highly reactive
 418 photochemical forms. As a result, partitioning between fast and slow species at sunrise/sunset leads to large
 419 errors and mass balance issues associated with the use of the first-order approximation (equation (4)).
 420 Halogen errors are compounded due to their cycling through gas and aerosol phases. These problems were
 421 previously identified by Shen et al. (2020, 2022). Halogen errors compounded with long residence times
 422 are the cause of the slow error growth for other species in the stratosphere. The impact is much less in the
 423 troposphere.

424
 425 Our results in Figure 4 indicate that halogen species should be kept in the implicit solver as fast species if
 426 the primary interest of the simulation is the stratosphere. Keeping the halogen radicals and their reservoir
 427 species as fast prevent large errors for halogen species in the stratosphere, in addition to avoiding error
 428 growth and spikes in errors as shown in Figure 4(d). Similarly, atomic oxygen (O) and HO_2 radicals are the

429 cause of high sulfate errors in the stratosphere (exceeding 10% above 85 hPa), and may be kept as fast to
430 reduce this error. The performance improvement of the adaptive mechanism is then 23%, as compared to
431 32% in the standard configuration. Such ad hoc adjustments to the fast/slow species partitioning would be
432 mechanism-specific and can be experimented with by users but we find in GEOS-Chem that they are not
433 needed for standard applications focusing on oxidant-aerosol tropospheric chemistry.

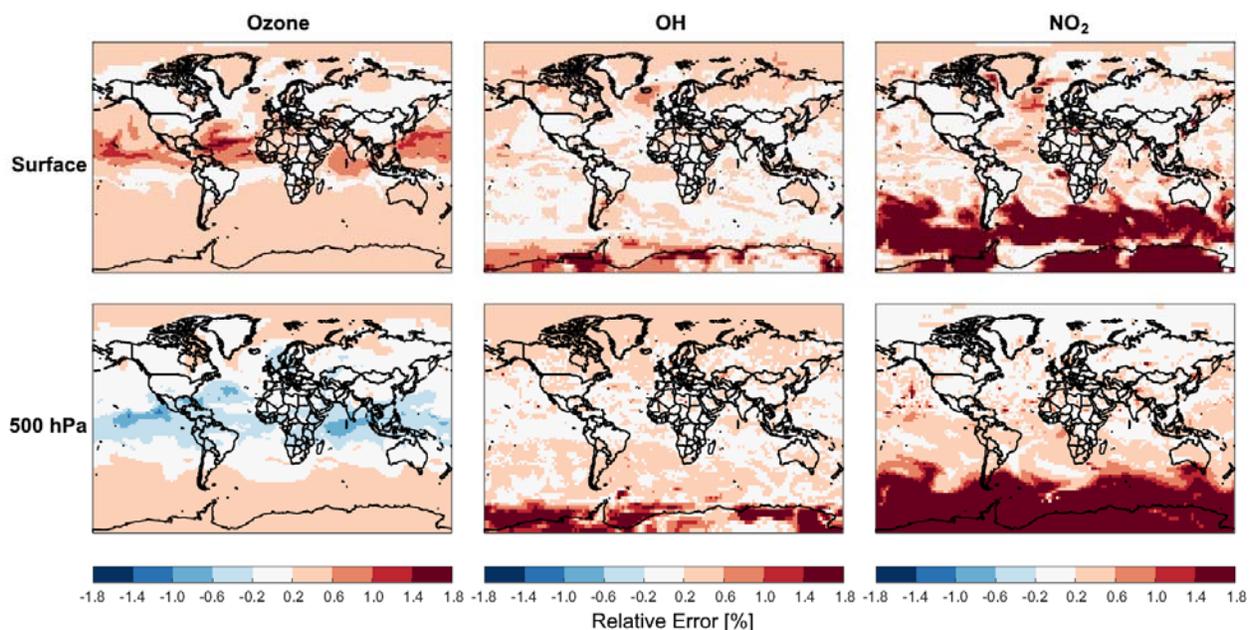
434

435 We find in this application that the append functionality (allowing species to switch from slow to fast over
436 internal time steps) does not provide significant error reduction, in particular for the halogen species, and
437 degrades the performance improvement of the adaptive mechanism to 24% instead of 32%. We retain it as
438 an option in the adaptive solver code as it may be helpful for longer external time steps (here 20 minutes)
439 or other chemical mechanisms.



440
 441 **Figure 4. Accuracy of the adaptive solver in a 1-year GEOS-Chem simulation.** Panel (a) shows the
 442 mean RRMS errors for species in different categories at the end of the 1-year simulation starting on July 1,
 443 2014. The categories are as defined in the standard GEOS-Chem benchmarking output diagnostics (Table

444 S1). Panels (b)-(d) show the time evolution of RRMS errors for all species in the boundary layer, free
 445 troposphere, and stratosphere, respectively, with colored lines for species of particular interest.
 446 The adaptive solver uses a dynamically defined threshold (equation (5)) with target species OH in daytime
 447 ($\alpha_{\text{OH}} = 5 \times 10^{-5}$) and NO₂ at night ($\alpha_{\text{NO}_2} = 1 \times 10^{-4}$), and does not force any species below that
 448 threshold to remain as fast.



449
 450 **Figure 5. Accuracy of the adaptive solver for ozone, OH, and NO₂ global distributions.** The Figure
 451 shows the errors relative to the full solver for the daily mean concentrations on July 1, 2015, the last day of
 452 the 1-year simulation. The relative error can be up to $\pm 3\%$ at high latitudes for OH and up to $\pm 10\%$ over
 453 the Southern Ocean for surface NO₂.

454 5 Conclusions

455 We presented an updated version of the Kinetic Pre-Processor (KPP 3.0.0) to integrate stiff chemical
 456 mechanism kinetics. KPP was originally designed for flexibility and speed. KPP 3.0.0 features several
 457 improvements for performance, diagnostics, choice of solvers, and code openness. It includes an adaptive
 458 solver capability for mechanism auto-reduction where and when the full mechanism is not needed.

459
 460 The adaptive solver performs auto-reduction of the chemical mechanism locally and on-the-fly at runtime,
 461 by comparing the local production and loss rates of each species with a partitioning threshold (δ). Species
 462 with production and loss rates higher than the threshold are considered fast and are solved as a coupled sub-
 463 mechanism within KPP, while other species are considered slow and solved individually by an explicit

464 method. Previous application of this adaptive solver method suffered from large overhead due to the need
465 for local reconstruction of the reduced Jacobian matrix and the associated memory allocation and
466 deallocation. We solved this problem here by using pre-computed Jacobian terms for the full mechanism
467 with a mapping operation to crop rows and columns corresponding to the slow species without changing
468 the memory allocation.

469
470 KPP 3.0.0 features additional improvements for performance, diagnostics, versatility, and openness.
471 Improved performance includes more efficient calculation of reaction rates from the KPP rate functions
472 and thread safety for parallelization. New diagnostics include individual reaction rates, production and loss
473 rates for chemical families, and stoichiometric numbers. Improved versatility includes expanded choice of
474 chemical solvers. KPP 3.0 is now hosted on GitHub (<https://github.com/KineticPreProcessor/KPP>) to
475 enable community access, development, and testing.

476
477 We evaluated the adaptive solver implemented in KPP 3.0.0 in a 1-year simulation with the global 3-D
478 GEOS-Chem atmospheric chemistry model including 291 species in the full mechanism for oxidant-aerosol
479 chemistry in the troposphere and stratosphere. Results show that a 32% performance improvement in the
480 solver can be achieved with a target error of 1% for key species in the troposphere. Errors in the stratosphere
481 can be larger, driven by halogen chemistry. Lower errors especially in halogen species can be achieved by
482 keeping these species within the fast sub-mechanism but reduces the performance improvement to 23% and
483 is mainly beneficial in the stratosphere. Our standard configuration of KPP and the adaptive solver within
484 the GEOS-Chem model is mechanism-independent and can be readily applied to other models using KPP.

485
486 The release of KPP 3.0.0 introduces improvements in development infrastructure, diagnostics, and
487 performance, particularly in Fortran 90 applications. However, one of the strengths of the KPP software is
488 the capability to generate code for different programming languages. Development directions for future
489 versions include (1) adding support for modern languages such as Python and Julia; (2) refactoring of the
490 generated code to avoid global data structures for easier parallelization; (3) streamlining inputs and outputs
491 of all integrators for consistency; (4) supporting the adaptive solver option in other integrators and
492 programming languages; and (5) Improve interaction and compatibility with the Master Chemical
493 Mechanism (<http://mcm.york.ac.uk>). These improvements will allow KPP to better serve the community
494 as a versatile tool for solving chemical kinetics.

495 **Acknowledgements**

496 This work was supported by the US EPA Science to Achieve Results (STAR) Program, by the NASA
497 Modeling, Analysis, and Prediction (MAP) Program, and by the NASA Atmospheric Composition
498 Modeling and Analysis Program (ACMAP).

499

500 **Open Research**

501 ***Data Availability Statement***

502 *Model Code Availability:* Source code for KPP 3.0.0 is available at

503 <https://github.com/KineticPreProcessor/KPP> (<https://doi.org/10.5281/zenodo.6828026>). The adaptive

504 solver box model is available at <https://github.com/KineticPreProcessor/KPP-AR-boxmodel/>

505 (<https://doi.org/10.5281/zenodo.6791657>). The adaptive solver implementation within the GEOS-Chem

506 atmospheric chemistry model used in this work is available at [https://github.com/jimmielin/geos-](https://github.com/jimmielin/geos-chem/tree/staging/autoreducekpp)

507 [chem/tree/staging/autoreducekpp](https://github.com/jimmielin/geos-chem/tree/staging/autoreducekpp) (<https://doi.org/10.5281/zenodo.6791655>).

508 **References**

- 509 Alvanos, M. & Christoudias, T. (2017). GPU-accelerated atmospheric chemical kinetics in the
510 ECHAM/MESSy (EMAC) Earth system model (version 2.52), *Geoscientific Model Development*, 10,
511 3679-3693. <https://doi.org/10.5194/gmd-10-3679-2017>
- 512 Bates, K. H. & Jacob, D. J. (2019). A new model mechanism for atmospheric oxidation of isoprene:
513 global effects on oxidants, nitrogen oxides, organic products, and secondary organic aerosol,
514 *Atmospheric Chemistry and Physics*, 19, 9613–9640. <https://doi.org/10.5194/acp-19-9613-2019>
- 515 Bates, K. H., Jacob, D. J., Li, K., Ivatt, P. D., Evans, M. J., Yan, Y., & Lin, J. (2021). Development and
516 evaluation of a new compact mechanism for aromatic oxidation in atmospheric models, *Atmospheric*
517 *Chemistry and Physics*, 21, 18351–18374. <https://doi.org/10.5194/acp-21-18351-2021>
- 518 Bey, I., Jacob, D. J., Yantosca, R. M., Logan, J. A., Field, B. D., Fiore, A. M., Li, Q., Liu, H. Y., Mickley,
519 L. J., & Schultz, M. G. (2001). Global modeling of tropospheric chemistry with assimilated
520 meteorology: Model description and evaluation, *Journal of Geophysical Research: Atmospheres*, 106,
521 23073–23095. <https://doi.org/10.1029/2001JD000807>
- 522 Bian, H., & Prather, M. J. (2002). Fast-J2: Accurate Simulation of Stratospheric Photolysis in Global
523 Chemical Models, *Journal of Atmospheric Chemistry*, 41, 281-296.
524 <https://doi.org/10.1023/A:1014980619462>
- 525 Brasseur, G. P. & Jacob, D. J. (2017). *Modeling of atmospheric chemistry*, Cambridge: Cambridge
526 University Press. <https://doi.org/10.1017/9781316544754>
- 527 Brown-Steiner, B., Selin, N. E., Prinn, R., Tilmes, S., Emmons, L., Lamarque, J.-F., & Cameron-Smith,
528 P. (2018). Evaluating simplified chemical mechanisms within present-day simulations of the
529 Community Earth System Model version 1.2 with CAM4 (CESM1.2 CAM-chem): MOZART-4 vs.
530 Reduced Hydrocarbon vs. Super-Fast chemistry, *Geoscientific Model Development*, 11, 4155–4174.
531 <https://doi.org/10.5194/gmd-11-4155-2018>
- 532 Brown, P. N., Byrne, G. D., & Hindmarsh, A. C. (1989). VODE: A Variable-Coefficient ODE Solver,
533 *SIAM Journal on Scientific and Statistical Computing*, 10(5), 1038-1051.
534 <https://doi.org/10.1137/0910062>
- 535 Damian, V., Sandu, A., Damian, M., Potra, F., & Carmichael, G. R. (2002). The kinetic preprocessor
536 KPP-a software environment for solving chemical kinetics, *Computers & Chemical Engineering*,
537 26(11), 1567–1579. [https://doi.org/10.1016/S0098-1354\(02\)00128-X](https://doi.org/10.1016/S0098-1354(02)00128-X)
- 538 Dawson, M. L., Guzman, C., Curtis, J. H., Acosta, M., Zhu, S., Dabdub, D., Conley, A., West, M.,
539 Riemer, N., & Jorba, O. (2022). Chemistry Across Multiple Phases (CAMP) version 1.0: an integrated
540 multiphase chemistry model, *Geoscientific Model Development*, 15, 3663-3689.
541 <https://doi.org/10.5194/gmd-15-3663-2022>

- 542 Djouad, R., & Sportisse, B. (2002). Partitioning techniques and lumping computation for reducing
543 chemical kinetics. APLA: An automatic partitioning and lumping algorithm, *Applied Numerical*
544 *Mathematics*, 43(4), 383-398. [https://doi.org/10.1016/S0168-9274\(02\)00111-3](https://doi.org/10.1016/S0168-9274(02)00111-3)
- 545 Eastham, S. D., Long, M. S., Keller, C. A., Lundgren, E., Yantosca, R. M., Zhuang, J., Li, C., Lee, C. J.,
546 Yannetti, M., Auer, B. M., Clune, T. L., Kouatchou, J., Putman, W. M., Thompson, M. A., Trayanov, A.
547 L., Molod, A. M., Martin, R. V., & Jacob, D. J. (2018). GEOS-Chem High Performance (GCHP v11-
548 02c): a next-generation implementation of the GEOS-Chem chemical transport model for massively
549 parallel applications, *Geoscientific Model Development*, 11, 2941–2953. [https://doi.org/10.5194/gmd-](https://doi.org/10.5194/gmd-11-2941-2018)
550 [11-2941-2018](https://doi.org/10.5194/gmd-11-2941-2018)
- 551 Fast, J. D., Gustafson Jr., W. I., Easter, R. C., Zaveri, R. A., Barnard, J. C., Chapman, E. G., Grell, G. A.,
552 & Peckham, S. E. (2006). Evolution of ozone, particulates, and aerosol direct radiative forcing in the
553 vicinity of Houston using a fully coupled meteorology-chemistry-aerosol model, *Journal of Geophysical*
554 *Research: Atmospheres*, 111(D21), D21305. <https://doi.org/10.1029/2005JD006721>
- 555 Gong, W., & Cho, H.-R. (1993). A numerical scheme for the integration of the gas-phase chemical rate
556 equations in three-dimensional atmospheric models, *Atmospheric Environment. Part A. General Topics*,
557 27(14), 2147-2160. [https://doi.org/10.1016/0960-1686\(93\)90044-Y](https://doi.org/10.1016/0960-1686(93)90044-Y)
- 558 Grell, G. A., Peckham, S. E., Schmitz, R., McKeen, S. A., Frost, G., Skamarock, W. C., & Eder, B.
559 (2005). Fully coupled “online” chemistry within the WRF model, *Atmospheric Environment*, 39(37),
560 6957–6975. <https://doi.org/10.1016/j.atmosenv.2005.04.027>
- 561 Hairer, E. & Wanner, G. (1991). *Solving Ordinary Differential Equations II. Stiff and Differential-*
562 *Algebraic Problems*, Berlin: Springer.
- 563 Hakami, A., Henze, D. K., Seinfeld, J. H., Singh, K., Sandu, A., Kim, S., Byun, D., and Li, Q. (2007).
564 The Adjoint of CMAQ, *Environmental Science & Technology*, 41, 7807-7817.
565 <https://doi.org/10.1021/es070944p>
- 566 Henze, D. K., Hakami, A., & Seinfeld, J. H. (2007). Development of the adjoint of GEOS-Chem,
567 *Atmospheric Chemistry and Physics*, 7, 2413-2433. <https://doi.org/10.5194/acp-7-2413-2007>
- 568 Holmes, C. D., Bertram, T. H., Confer, K. L., Graham, K. A., Ronan, A. C., Wirks, C. K., & Shah, V.
569 (2019). The Role of Clouds in the Tropospheric NO_x cycle: A New Modeling Approach for Cloud
570 Chemistry and Its Global Implications, *Geophysical Research Letters*, 46(9), 4980-4990.
571 <https://doi.org/10.1029/2019GL081990>
- 572 Jacobson, M. Z., & Turco, R. P. (1994). SMVGEAR: A sparse-matrix, vectorized gear code for
573 atmospheric models, *Atmospheric Environment*, 28(2), 273-284. [https://doi.org/10.1016/1352-](https://doi.org/10.1016/1352-2310(94)90102-3)
574 [2310\(94\)90102-3](https://doi.org/10.1016/1352-2310(94)90102-3)

- 575 Jacobson, M. Z. (1995). Computation of global photochemistry with SMVGEAR II, *Atmospheric*
576 *Environment*, 29(18), 2541–2546. [https://doi.org/10.1016/1352-2310\(95\)00194-4](https://doi.org/10.1016/1352-2310(95)00194-4)
- 577 Jöckel, P., Kerkweg, A., Pozzer, A., Sander, R., Tost, H., Riede, H., Baumgaertner, A., Gromov, S., &
578 Kern, B. (2010). Development cycle 2 of the Modular Earth Submodel System (MESSy2), *Geoscientific*
579 *Model Development*, 3, 717–752. <https://doi.org/10.5194/gmd-3-717-2010>
- 580 Keller, C. A., & Evans, M. J. (2019). Application of random forest regression to the calculation of gas-
581 phase chemistry within the GEOS-Chem chemistry model v10, *Geoscientific Model Development*, 12,
582 1209–1225. <https://doi.org/10.5194/gmd-12-1209-2019>
- 583 Kelp, M. M., Jacob, D. J., Kutz, J. N., Marshall, J. D., & Tessum, C. W. (2020). Toward stable, general
584 machine-learned models of the atmospheric chemical system, *Journal of Geophysical Research:*
585 *Atmospheres*, 125(23), e2020JD032759. <https://doi.org/10.1029/2020JD032759>
- 586 Kelp, M. M., Jacob, D. J., Lin, H., & Sulprizio, M. P. (2022). An Online-Learned Neural Network
587 Chemical Solver for Stable Long-Term Global Simulations of Atmospheric Chemistry, *Journal of*
588 *Advances in Modeling Earth Systems*, 14(6), e2021MS002926. <https://doi.org/10.1029/2021MS002926>
- 589 Mao, J., Jacob, D. J., Evans, M. J., Olson, J. R., Ren, X., Brune, W. H., Clair, J. M. St., Crounse, J. D.,
590 Spencer, K. M., Beaver, M. R., Wennberg, P. O., Cubison, M. J., Jimenez, J. L., Fried, A., Weibring, P.,
591 Walega, J. G., Hall, S. R., Weinheimer, A. J., Cohen, R. C., Chen, G., Crawford, J. H., McNaughton, C.,
592 Clarke, A. D., Jaeglé, L., Fisher, J. A., Yantosca, R. M., Le Sager, P., & Carouge, C. (2010). Chemistry
593 of hydrogen oxide radicals (HO_x) in the Arctic troposphere in spring, *Atmospheric Chemistry and*
594 *Physics*, 10, 5823–5838. <https://doi.org/10.5194/acp-10-5823-2010>
- 595 Moch, J. M., Dovrou, E., Mickley, L. J., Keutsch, F. N., Liu, Z., Wang, Y., Dombek, T. L., Kuwata, M.,
596 Budisulistiorini, S. H., Yang, L., Decesari, S., Paglione, M., Alexander, B., Shao, J., Munger, J. W., &
597 Jacob, D. J. (2020). Global Importance of Hydroxymethanesulfonate in Ambient Particulate Matter:
598 Implications for Air Quality, *Journal of Geophysical Research: Atmospheres*, 125(18), e2020JD032706.
599 <https://doi.org/10.1029/2020JD032706>
- 600 Rastigejev, Y., Brenner, M. P., & Jacob, D. J. (2007). Spatial Reduction Algorithm for Atmospheric
601 Chemical Transport Models, *PNAS*, 104(35), 13875–13880. <https://doi.org/10.1073/pnas.0705649104>
- 602 Rosenbrock, H. H. (1963). Some general implicit processes for the numerical solution of differential
603 equations, *The Computer Journal*, 5(4), 329–330, <https://doi.org/10.1093/comjnl/5.4.329>
- 604 Sander, R., Baumgaertner, A., Cabrera-Perez, D., Frank, F., Gromov, S., Groß, J.-U., Harder, H.,
605 Huijnen, V., Jöckel, P., Karydis, V. A., Niemeyer, K. E., Pozzer, A., Riede, H., Schultz, M. G.,
606 Taraborrelli, D., & Tauer, S. (2019). The community atmospheric chemistry box model
607 CAABA/MECCA-4.0, *Geoscientific Model Development*, 12, 1365–1385. [https://doi.org/10.5194/gmd-](https://doi.org/10.5194/gmd-12-1365-2019)
608 [12-1365-2019](https://doi.org/10.5194/gmd-12-1365-2019)

- 609 Sandu, A., Verwer, J., Blom, J., Spee, E., Carmichael, G., & Potra, F. (1997). Benchmarking stiff ode
610 solvers for atmospheric chemistry problems II: Rosenbrock solvers, *Atmospheric Environment*, 31(20),
611 3459–3472. [https://doi.org/10.1016/S1352-2310\(97\)83212-8](https://doi.org/10.1016/S1352-2310(97)83212-8)
- 612 Sandu, A., & Sander, R. (2006). Technical note: Simulating chemical systems in Fortran90 and Matlab
613 with the Kinetic PreProcessor KPP-2.1, *Atmospheric Chemistry and Physics*, 6, 187–195.
614 <https://doi.org/10.5194/acp-6-187-2006>
- 615 Santillana, M., Le Sager, P., Jacob, D. J., & Brenner, M. P. (2010). An adaptive reduction algorithm for
616 efficient chemical calculations in global atmospheric chemistry models, *Atmospheric Environment*,
617 44(35), 4426–4431. <https://doi.org/10.1016/j.atmosenv.2010.07.044>
- 618 Shen, L., Jacob, D. J., Santillana, M., Wang, X., & Chen, W. (2020). An adaptive method for speeding up
619 the numerical integration of chemical mechanisms in atmospheric chemistry models: application to
620 GEOS-Chem version 12.0.0, *Geoscientific Model Development*, 13, 2475–2486.
621 <https://doi.org/10.5194/gmd-13-2475-2020>
- 622 Shen, L., Jacob, D. J., Santillana, M., Bates, K., Zhuang, J., & Chen, W. (2022). A machine-learning-
623 guided adaptive algorithm to reduce the computational cost of integrating kinetics in global atmospheric
624 chemistry models: application to GEOS-Chem versions 12.0.0 and 12.9.1, *Geoscientific Model*
625 *Development*, 15, 1677–1687. <https://doi.org/10.5194/gmd-15-1677-2022>
- 626 Wang, X., Jacob, D. J., Downs, W., Zhai, S., Zhu, L., Shah, V., Holmes, C. D., Sherwen, T., Alexander,
627 B., Evans, M. J., Eastham, S. D., Neuman, J. A., Veres, P. R., Koenig, T. K., Volkamer, R., Huey, L. G.,
628 Bannan, T. J., Percival, C. J., Lee, B. H., & Thornton, J. A. (2021). Global tropospheric halogen (Cl, Br,
629 I) chemistry and its impact on oxidants, *Atmospheric Chemistry and Physics*, 21, 13973–13996.
630 <https://doi.org/10.5194/acp-21-13973-2021>
- 631 Young, T. R., & Boris, J. P. (1977). A Numerical Technique for Solving Stiff Ordinary Differential
632 Equations Associated with the Chemical Kinetics of Reactive-Flow Problems, *Journal of Physical*
633 *Chemistry*, 81(25), 2424–2427. <https://doi.org/10.1021/j100540a018>
- 634 Zhao, S., Russell, M. G., Hakami, A., Capps, S. L., Turner, M. D., Henze, D. K., Percell, P. B., Resler, J.,
635 Shen, H., Russell, A. G., Nenes, A., Pappin, A. J., Napelenok, S. L., Bash, J. O., Fahey, K. M.,
636 Carmichael, G. R., Stanier, C. O., & Chai, T. (2020). A multiphase CMAQ version 5.0 adjoint,
637 *Geoscientific Model Development*, 13, 2925–2944. <https://doi.org/10.5194/gmd-13-2925-2020>
- 638 Zhuang, J., Jacob, D. J., Lin, H., Lundgren, E. W., Yantosca, R. M., Gaya, J. F., Sulprizio, M. P., &
639 Eastham, S. D. (2020). Enabling High-Performance Cloud Computing for Earth Science Modeling on
640 Over a Thousand Cores: Application to the GEOS-Chem Atmospheric Chemistry Model, *Journal of*
641 *Advances in Modeling Earth Systems*, 12(5), e2020MS002064. <https://doi.org/10.1029/2020MS002064>