SANDIA REPORT
SAND2024-01807R
Printed February 2024

Sandia National Laboratories

Benchmarking TChem for Potential Incorporation into E3SM as a Replacement Chemical Kinetics Solver. No. 233396

Oscar Diaz-Ibarra, Michael J. Schmidt, Cosmin Safta

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@osti.gov
Online ordering:	http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.gov
Online order:	https://classic.ntis.gov/help/order-methods

ABSTRACT

In this quick target investigation (QTI), we have generated performance benchmarks to demonstrate that TChem is a viable replacement for E3SM's current atmospheric chemistry solver. Currently, E3SM uses the Community Atmospheric Model Pre-Processor (CAMPP) to generate a set of Fortran files to represent and solve the chemical kinetic model, which includes reaction coefficients, the hand side of volumetric mixing ratio (vmr, denoted η), and ODE solvers (Implicit/Explicit solver). Ultimately, the goal of CAMPP is to compute an updated vmr ($\eta_{t+\Delta t}$) for the troposphere and stratosphere. Thus, we aim to reproduce the $\eta_{t+\Delta t}$ values using TChem. We generate input files for TChem using the E3SM v3 chemistry mechanism, the "UCI mechanism" (University of California Irvine). This input file is created using CAMPP's input file `chem.inp`, CAMPP's output file `chem.dat`, a Mozart chemistry reaction file (`usrrxt.F90`), and UCI reaction types (`O1D_to_2OH_adj.F90`). We also implemented the necessary functionalities to compute the reaction constant of the UCI mechanism. To verify the TChem implementation and input files, we ran E3SM and produced data sets for comparison, in order to verify TChem's implementation and input files. Furthermore, we implemented a standalone driver using the CAMPP's Fortran files to produce outputs and performance measurements more rapidly, without the need to run full E3SM simulations. We compare TChem and E3SM's outputs: rate of progress, net production rates or RHS of η, and $\eta_{t+\Delta t}$. We show that the relative root-mean-square error (RRMSE difference between TChem and E3SM) for the rate of progress, net production rates are lower than 2×10^{-7} while RRMSE of $\eta_{t+\Delta t}$ is larger due to differences of the ODE solver.

In this work, we demonstrate that TChem can reproduce E3SM's gas chemistry results. The computational performance of TChem is similar to E3SM's chemistry solver when we compare both codes using the same input conditions, numerical tolerances, and ODE solver (implicit/explicit Euler) However, TChem uses multiple threads, and E3SM's chemistry solver is a serial code at the node level. We note that the E3SM's chemistry solver uses CAMPP-generated Fortran files, and that choice is always likely to be more computationally efficient than a flexible chemistry solver such as TChem. In its current form, TChem constructs the RHSs and Jacobians for a reaction mechanism at simulation time, leading to run-time overhead. In summary, we have presented evidence that TChem is a reasonable choice of chemistry solver for E3SM because it can reproduce the results of the incumbent chemistry solver and displays similar computational performance.

This page intentionally left blank.

ACKNOWLEDGEMENT

We would like to thank our colleagues Qi Tang (LLNL), Balwinder Singh (PNNL), and Susannah Burrows (PNNL) for all of their guidance and recommendations, and we would also like to thank Ziming Ke (LLNL) for his support and assistance with the climate simulation datasets.

This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

This page intentionally left blank.

CONTENTS

This page intentionally left blank.

LIST OF FIGURES

This page intentionally left blank.

LIST OF TABLES

This page intentionally left blank.

1. INTRODUCTION AND EXECUTIVE SUMMARY OF RESULTS

The Energy Exascale Earth System Model (E3SM) is a vital Department of Energy tool that is designed to perform whole-earth climate simulations in support of the mission of Sandia National Laboratories. Currently, E3SM uses the Community Atmospheric Model Pre-Processor (CAMPP) to solve atmospheric gas chemistry. A consequence of using this tool is that any change to the chemical mechanism requires code rewrites and rebuilds. Most importantly, there is no path for porting CAMPP to exascale architectures due to legacy code restraints. As it stands, "E3SM project leadership currently expects that future versions of E3SM (v4+) will need an atmospheric chemistry code that is written in C++ /Kokkos and runs efficiently on GPUs, but there is currently no settled plan for what will be used". The tool we propose, TChem, is a performance-portable software toolkit for analyzing complex kinetic chemistry models. TChem solves atmospheric chemistry problems and provides the host model with chemical source terms and the associated Jacobian. TChem is written in C++ and designed to be performant on next-generation heterogeneous computing platforms via the Kokkos programing model. In the long term, we seek to provide E3SM with a capable and flexible product that solves arbitrary chemical systems in a fast and computationally efficient manner. To support this goal, we will compare performance and validate TChem against CAMPP, as applied to the E3SM v3 chemistry mechanism, the "UCI mechanism" (University of California Irvine).

This page intentionally left blank.

2. DETAILED DESCRIPTION OF RESEARCH AND DEVELOPMENT AND METHODOLOGY

The overarching goal of this work is to provide evidence that TChem is a viable replacement for E3SM's current atmospheric chemistry solver, CAMPP. In service of this goal, we divided our work into three main thrusts:

1. Software development (Section 2.1), composed of 3 sub-tasks:

   i. Develop new TChem functionality for solving E3SM's UCI chemistry mechanism (Section 2.1.1).

   ii. Create standalone driver for the E3SM chemistry solver (Section 2.1.2).

   iii. Generate code infrastructure and scripts to automate testing (Section 2.1.3).

2. Validate TChem's results against those of E3SM to provide evidence of correctness and reliability/robustness.

   - **Methodology:** Section 2.2

   - **Results:** Section 3.1

3. Profile TChem's computational performance against that of E3SM in a controlled manner, so as to provide "apples-to-apples" comparison data.

   - Computational performance, in this case, is measured by various descriptions related to the two solver's run time or "wall-clock time" (also referred to as "wall time").

   - **Methodology:** Section 2.3

   - **Results:** Section 3.2

2.1. Software development Methodology

The software development required for this QTI includes: adding functionality to TChem so that it may solve the UCI chemistry mechanism; creating a standalone driver that runs the E3SM chemistry solver with the UCI chemistry inputs; and generating various scripts and code products to facilitate efficient validation testing and performance profiling.

2.1.1. New TChem Functionality

The core elements of TChem's atmospheric chemistry capabilities existed before this project. TChem was already capable of ingesting an atmospheric-chemistry-focused yaml input file, that was developed in collaboration with Matt Dawson, a researcher at the National Center for Atmospheric Research (NCAR). TChem had also been designed to process the standard Troe and Arrhenius reaction rate coefficients in the context of conventions specific to atmospheric chemistry, and TChem also contained a handful of custom reaction types that are commonly used in E3SM and other climate models.

Despite this preexisting functionality, the UCI chemistry mechanism contains a few peculiarities that necessitated additional development work. Notable additions are:

- Some reaction rates that are hard-coded into E3SM needed to be converted into standard format, and a reference document provided by Matt Dawson was instrumental in this effort. Examples include:
  - Representing UCI reactions (#4-5) as the sum of 2 Arrhenius reactions.
  - The addition of a *Ratio JPL-Arrhenius type* to represent UCI reactions that required custom-defined rate coefficients (#7-9) to be represented as the ratio of two Arrhenius coefficients.
- The addition of modification factors to standard reaction types to match values given by the UCI reactions(#1-3).
- Adding new chemical forcing terms to incorporate emissions (external forcing).
- Adding inputs to incorporate photolysis reactions.

Prior to the work of this project, TChem included native support for a second-order trapezoidal backward-difference formulation (TrBDF2) and support for the external solver package SUNDIALS that contains the CVODE solver [2, 3]. In contrast, E3SM CAMPP employs both an implicit and explicit first-order Euler solver. As such, both of these Euler solvers were hand-coded and added to TChem.

2.1.2. Standalone E3SM Chemistry Driver

Within E3SM, the CAMPP solver cannot be run on its own and instead requires building and running all of E3SM if one desires to analyze the inputs and outputs of a chemistry mechanism. This is the workflow that was employed to generate validation data for TChem but is not the best way of isolating chemistry calculations for performance profiling. As a result, part of the development of this project was focused on "carving out" the code related to solving atmospheric chemistry and building an isolated standalone driver. This standalone driver could be provided with input files that are functionally equivalent to those required by TChem. Using this standalone driver served the dual purposes of faster generation of comparison data for validation, as well as providing a "laboratory" condition for performance profiling comparison with no added overhead from running the entirety of E3SM.

2.1.3. Testing Infrastructure

In order to maximize productive throughput on this work, we made efforts to automate most steps of the validation testing and performance profiling. This was achieved by employing a number of shell scripts, Python scripts, and Jupyter Notebooks. While these tools are not quite general enough to be "plug and play" for a new user unfamiliar with TChem or E3SM, they greatly improved productivity for the investigators and could be readily adapted for other users with similar research goals. The tasks handled by these tools are:

- Running ensembles of E3SM simulations with UCI chemistry for a selection of atmospheric columns (see Table 2-1);
- Converting E3SM atmospheric chemistry subroutine inputs and outputs into compatible yaml files to serve as inputs and validation data for TChem and the standalone driver.
  - This was particularly handy when generating inputs and outputs for 15 different atmospheric columns.
- Running standalone driver ensembles.
- Running TChem ensembles composed of various choices of:
  - atmospheric conditions consisting of samples selected around the globe (see Table 2-1);
  - ODE solver;
  - number of chemistry cells ("batches");
  - thread configuration;
  - case type;
  - "device" type–i.e., chip architecture: CPU, GPU.
- Post-processing data and generating figures.

2.2. Validation Methodology

Making an argument that TChem is a viable replacement atmospheric chemistry solver for E3SM requires that the solutions generated by TChem are both correct and robust to varying initial conditions. Prior to this project, TChem has been verified against analytic solutions and canonical chemistry problems (both atmospheric and combustion). TChem has also been validated against NCAR's Chemistry Across Multiple Phases (CAMP) model for both single-reaction-type cases and *Carbon Bond 5 (CB05)* chemistry–a complex system with 67 chemical species and 187 reactions with reaction time scales varying across orders of magnitude.

The work of this project was therefore focused on recreating the results of E3SM's atmospheric chemistry solver for the UCI chemistry system. To accomplish this, we gathered input and output data from the E3SM atmosphere module's chemistry-solving subroutines with the names and functions given in Table 2-2. This data was sampled at a variety of locations that span the

Table 2-1. Abbreviation, Coordinates, and Location for the atmospheric columns used in validation and performance testing.

Abbreviation	Latitude	Longitude	Location
LA	34.0549	-118.2426	Los Angeles, CA, USA
BRW	71.323	-156.6114	North Slope, AK, USA
MHD	53.326	-9.899	Halfmace, County Galway, Ireland
PSA	-64.7742	-64.0527	Palmer Station, Antarctica
RPB	13.165	-59.432	Ragged Point, Barbados
SYO	-69.0125	39.59	Showa Station, Antarctica
ZEP	78.9067	11.8883	Zeppelin mountain, Ny-Ålesund, Norway

globe, and include New York City, Los Angeles, a point in the Southern Pacific Ocean, and the locations considered by the E3SM chemistry diagnostics package (ChemDyg) in the comparison of surface carbon monoxide calculations with observations made by NOAA (National Oceanic and Atmospheric administration). This data may be found at this webpage, and the location abbreviations and locations are given in Table 2-1.

The gathering of this data was facilitated by tools created in the refactoring of E3SM's aerosol module, MAM4 (4-mode Modal Aerosol Model). This work was conducted as a part of the joint PNNL/Sandia EAGLES project (Enabling Aerosol-cloud interactions at GLobal convection-permitting scalES). These tools were primarily designed by Balwinder Singh (PNNL) and were used to generate validation data for the newly-developed E3SM aerosol model MAM4xx (a performance-portable C++ implementation of MAM4). Additionally, we instrumented these functions with calls to E3SM's built-in timer for use in performance comparison (Section 2.3). We used this data from E3SM to create input files to run identical experiments with the standalone E3SM chemistry driver and TChem. The three measures of correctness we tracked for the UCI chemistry calculations are:

- ***Volumetric mixing ratio*** (vmr) $\left[\frac{\text{mol}}{\text{mol}}\right]$.

  The volumetric mixing ratio at the next time step, $\eta_{t+\Delta t}$, is computed by solving Eq (2.1) the ODE

$$\frac{\mathrm{d}\eta_k}{\mathrm{d}t} = \dot{\omega}_k. \tag{2.1}$$

- ***Net production rate*** $\left[\frac{\text{mol}}{\text{mol s}}\right]$.

  The net production rate of species k, $\dot{\omega}_k$, or the right hand side of Eq. (2.1) is computed using Eq. (2.2)

$$\dot{\omega}_k = \sum_{i=1}^{N_{\text{react}}} \nu_{ki} q_i, \quad \nu_{ki} = \nu''_{ki} - \nu'_{ki}, \tag{2.2}$$

Table 2-2. List of E3SM atmospheric chemistry subroutines relevant to this project and the associated purpose.

Subroutine	Purpose
`setrxt()`	compute reaction rates.
`usrrxt()`	compute user-defined reaction rates for Mozart model.
`adjrxt()`	multiply non-photolysis reactions by vmr of invariant species.
`O1D_to_2OH_adj()`	compute and adjust UCI reaction types.
`phtadj()`	multiply photolysis reactions by vmr of invariant species.
`imp_sol()`	implicit Euler solver (only tropospheric levels).
`indprd()`	invoked by `imp_sol()` to construct production term with vmr of invariant species.
`imp_prod_loss()`	invoked by `imp_sol()`, inside Newton-Raphson solver to compute loss and production terms.
`exp_sol()`	explicit Euler solver (tropospheric and stratospheric levels).
`set_rates`	compute net production rates (not involved in the solver, itself).

where q_i is the rate of progress of reaction i, N_{react} is the number of reactions, ν''_{ki} and ν'_{ki} are the stoichiometric coefficients of species k in reaction i for the reactant and product sides of the reaction, respectively.

- ***Rate of progress*** $\left[\frac{\text{mol}}{\text{mol s}}\right]$.

  Finally, the rate of progress of reaction i is computed as

  $$q_i = k_{f_i} \prod_{j=1}^{N_{\text{spec}}} \eta_j^{\nu'_{ji}}, \tag{2.3}$$

  where N_{spec} is the number of species, k_{f_i} is the reaction constant of reaction i. Note that in E3SM‘s CAMPP solver only forward reaction calculations are employed, and the single reaction constant values depend on the type of reaction. In TChem, we have implemented separate reaction types for Troe, Arrhenius, and JPL-Troe types.

2.3. Performance Profiling Methodology

The baseline measurements for TChem’s performance profiling were conducted on the *Chrysalis* system, a machine that is supported by the E3SM project and has a standard configuration to build and run on this platform. As mentioned in Section 2.2, timing functions were added to the subroutines that correspond to those in the standalone driver and TChem. We ran the same test case for UCI chemistry as was used in the validation testing, and we recorded execution time for the individual subroutine calls, which were aggregated for total run time. We compared these run time (wall-clock time) results to analogous runs using the standalone driver, and times were comparable.

This added confidence to the performance results for both types of runs and verified that they would serve as a reasonable basis of comparison for TChem performance results.

We ran simulation ensembles, as described in Section 2.1.3, for batch counts ranging from 72 to ≈1 million. We also used the default Kokkos settings for thread "team" and "vector" size, which is to say the threading strategy was unoptimized, and a deeper discussion of these abstractions is not germane to this work. TChem testing was conducted on a Sandia test-bed system that contains IBM Power9 CPU nodes (20 cores per socket, 8 SMT threads per core, and 2 sockets per node) as well as NVIDIA Tesla V100 GPU cards. The performance tests were run using the SPA atmosphere column (see Table 2-1). In post-processing, performance is measured by scaling the wall-clock time for an experiment by the number of batches in the experiment, generating line plots with wall-time units of seconds per batch vs. number of batches.

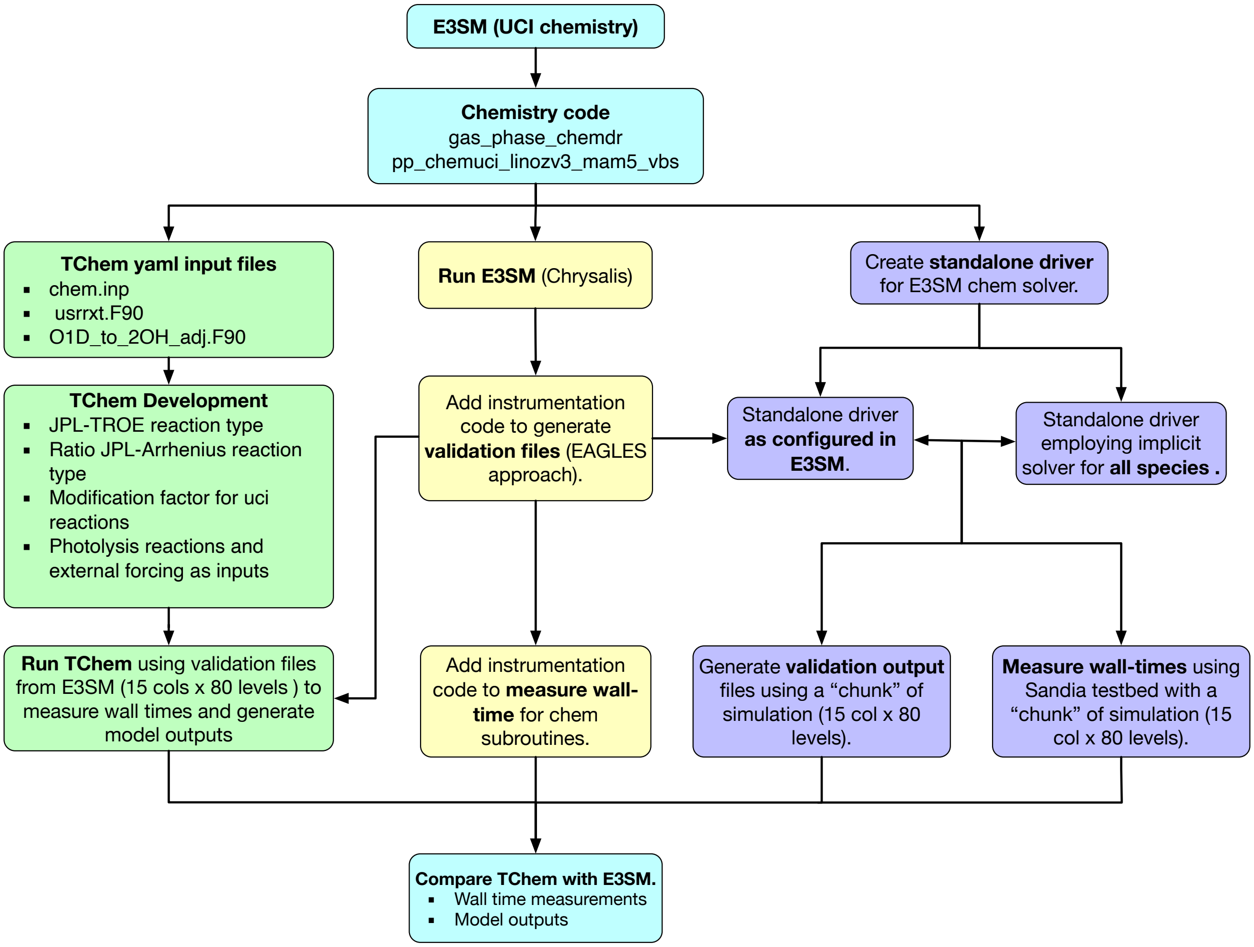
E3SM (UCI chemistry)
Chemistry code
gas_phase_chemdr
pp_chemuci_linozv3_mam5_vbs
TChem yaml input files
▪ chem.inp
▪ usrrxt.F90
▪ O1D_to_2OH_adj.F90
Run E3SM (Chrysalis)
Create standalone driver for E3SM chem solver.
TChem Development
▪ JPL-TROE reaction type
▪ Ratio JPL-Arrhenius reaction type
▪ Modification factor for uci reactions
▪ Photolysis reactions and external forcing as inputs
Add instrumentation code to generate validation files (EAGLES approach).
Standalone driver as configured in E3SM.
Standalone driver employing implicit solver for all species .
Run TChem using validation files from E3SM (15 cols x 80 levels) to measure wall times and generate model outputs
Add instrumentation code to measure wall-time for chem subroutines.
Generate validation output files using a "chunk" of simulation (15 col x 80 levels).
Measure wall-times using Sandia testbed with a "chunk" of simulation (15 col x 80 levels).
Compare TChem with E3SM.
▪ Wall time measurements
▪ Model outputs

Figure 2-1. Workflow schematic for validation and performance profiling of TChem vs. the incumbent E3SM chemistry solver. The leftmost/green column primarily summarizes the *development* work described in Section 2.1. The center/yellow column relates to the activities requiring hands-on running and analyzing the full E3SM model with UCI chemistry on the *Chrysalis* cluster. The rightmost/purple column summarizes the work related to standalone E3SM chemistry driver discussed in Section 2.1.2

3. RESULTS AND DISCUSSION

To verify TChem implementation for chemistry in the troposphere and stratosphere, we compare the TChem and E3SM standalone driver's outputs. We generate E3SM data for a selection of "chunks," composed of 15 atmospheric columns and 80 vertical levels (for locations, see Table 2-1) . As mentioned in Section 2.2, CAMPP's generated kinetic Fortran files are employed in the standalone driver and generate equivalent results in terms of: rate of progress (2.3), net production rate (2.2), and volumetric mixing ratio at the next time step, $\eta_{t+\Delta t}$ (2.1). TChem computes reaction rates per reaction using the kinetic parameters provided by the input files. We generate these input files using the CAMPP input file (`chem.inp`), the reported output file (`chem.doc`) produced by CAMPP, and the Fortran code wherein kinetic parameters are hard-coded. Additionally, definitions are used that are found in user-defined reaction types from the Mozart (`usrrxt.F90`) and UCI reaction types (`O1D_to_2OH_adj.F90`). We cannot directly compare the reaction constant values because of differences in the design of the CAMPP's generated Fortran code and TChem. However, the rate of progresses, and the net production rates, are reported by both the standalone driver and TChem. The measure of error we employ to compare the results of the E3SM standalone driver (y_E) and TChem (y_T), is the *relative root mean square error (RRMSE)*

$$\text{RRMSE} = \sqrt{\frac{\sum_{i=1}^{N}(y_{E,i} - y_{T,i})^2}{\sum_{i=1}^{N}(y_{E,i})^2}}. \tag{3.1}$$

3.1. TChem vs E3SM: verification

In E3SM, the Troposphere and Stratosphere contain distinct chemistry mechanisms, so we split the analysis of chemistry in these two layers into the following sections.

3.1.1. Troposphere

We present the comparison of TChem against the standalone driver for the rate of progress in Figure (3-1) and for the net production rates in Figure (3-2). In the ideal case of identical results, all points would lie along the diagonal line $y = x$ in these parity plots, and off-diagonal points indicate discrepancies between E3SM and TChem. In addition, we expect an RRMSE lower than 1×10^{-6} for the net production rates and rate of progress. Both figures show that all values are located perfectly along the $y = x$ diagonal, and RRMSE is $\approx 2\times10^{-7}$. To summarize, after implementing all the necessary reaction types from the UCI mechanism in TChem, we can reproduce outputs from the standalone driver. As a result, TChem is fully verified using the data points from an E3SM simulation.

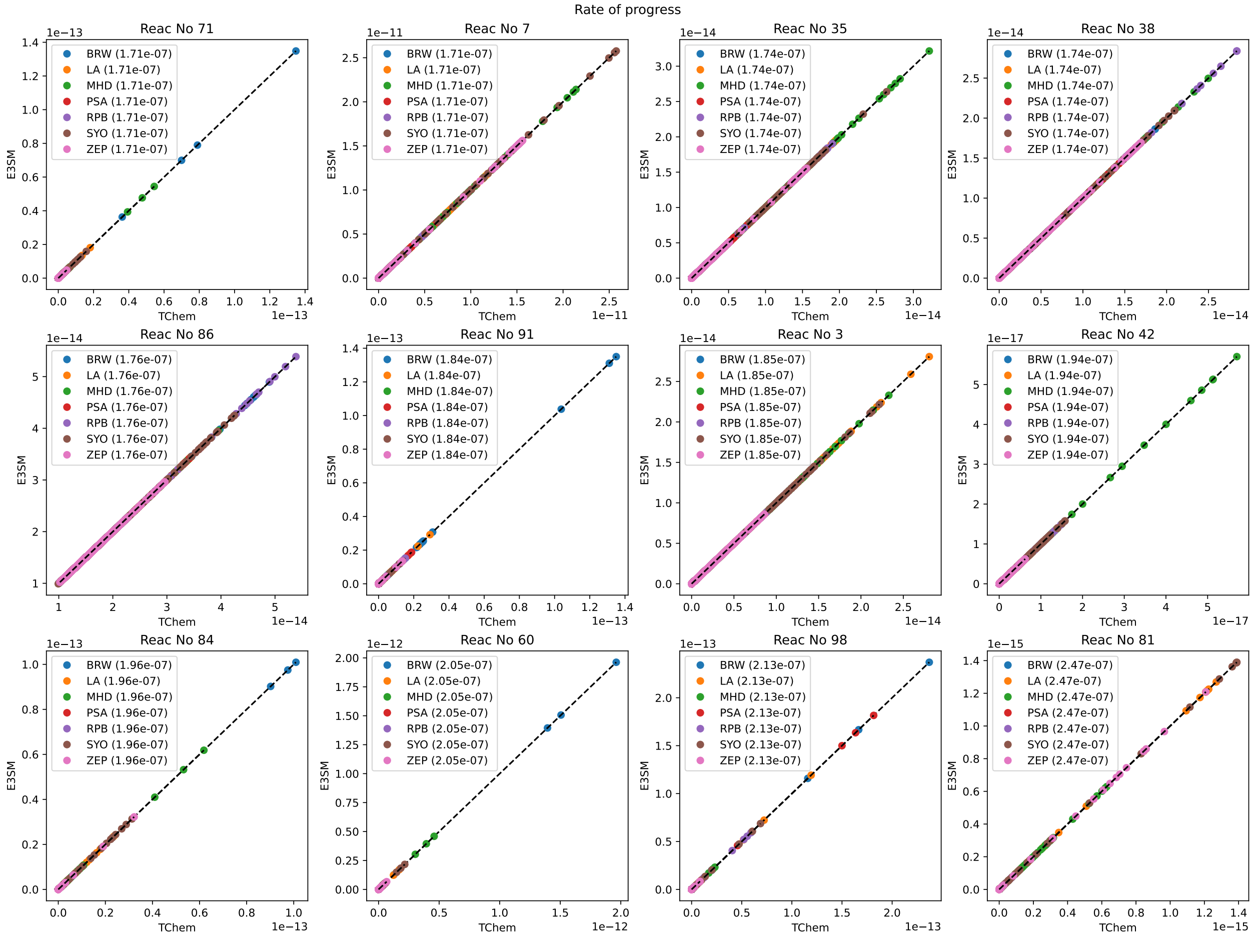

Figure 3-1. Parity plot for rate of progress in the troposphere. E3SM outputs are produced by the standalone driver. There are 104 reactions, and we display the 10 with the largest RRMSE for the locations given in Table 2-1. RRMSE per location is given in the legend of each plot.

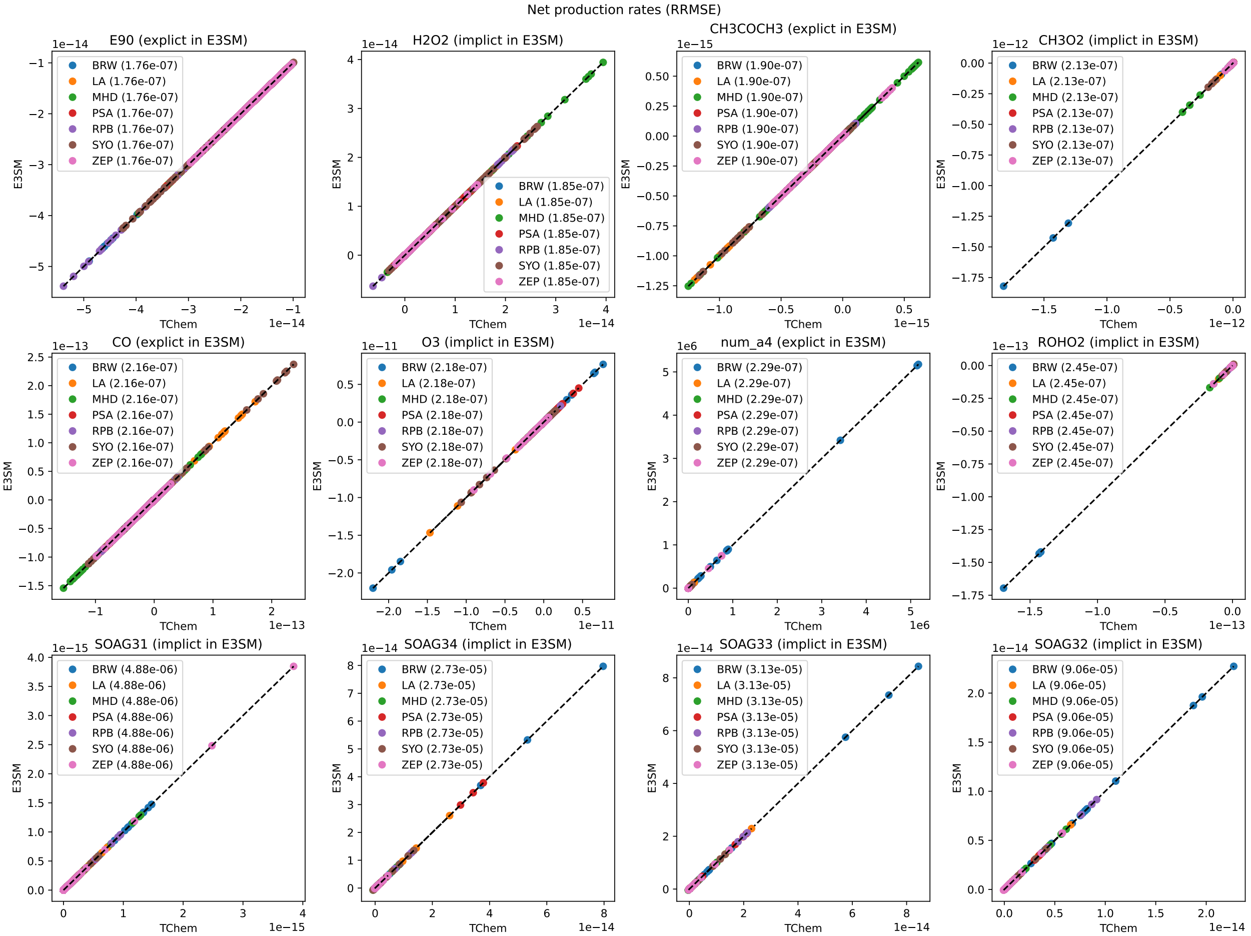

Figure 3-2. Parity plot for rate of progress in the troposphere. E3SM outputs are produced by CAMPP's generated code. There are 104 reactions, and we only display the 10 with the larges RRMSE for the locations given in Table 2-1. RRMSE per location is presented in each plot. The net production rates correspond to the RHS of the equations solved by E3SM and TChem.

Next, we compare vmr at the subsequent time step ($\eta_{t+\Delta t}$), which is the quantity-of-interest calculated for E3SM chemistry. After verifying the net production rates corresponding to the right-hand side (RHS) of vmr, we can be sure that any discrepancies between the values of $\eta_{t+\Delta t}$ computed by E3SM and TChem are attributable to the ODE solver. Note that the CAMPP's chemistry solver allows the user to split species into two classes: implicit and explicit, and these choices are specified in the `chem.inp` input file. The two classes are uncoupled in this approach, and the implicit solver is executed first. In TChem, the full ODE system is solved simultaneously, meaning an implicit solver may be applied to the whole system.

Figure 3-3 presents parity plots of $\eta_{t+\Delta t}$ for the standalone driver (E3SM) and TChem code. On the E3SM side, $\eta_{t+\Delta t}$ is computed by the explicit or implicit Euler solver. In the case of TChem, we employ an implicit Euler solver and use the same relative and absolute tolerance values of 1×10^{-3} that are used by the E3SM solver. We note that for vmr, RRMSE values are larger than the ones from the net production rates and rate of progress, though not excessively so.

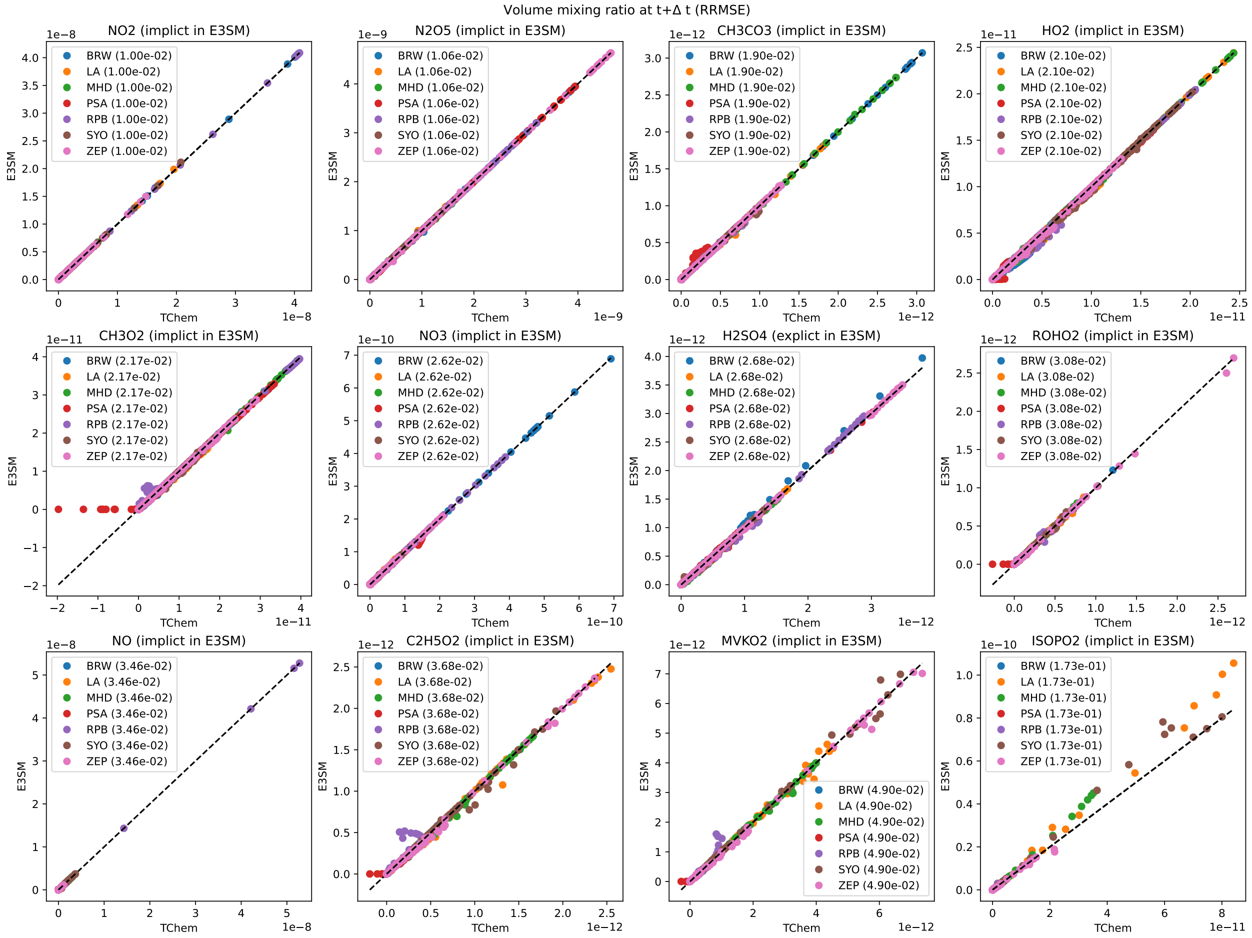

Figure 3-3. Parity plot for updated volumetric mixing ratio (vmr, $\eta_{t+\Delta t}$) in the troposphere. There are 82 species in the vmr solution vector, and we only display the ten species with the largest RRMSE for the locations given in Table 2-1.

3.1.2. Stratosphere

We compare the updated vmr values ($\eta_{t+\Delta t}$) and the net production rates (RHSs) in the stratosphere in Figures 3-5 and 3-4. CAMPP's generated Fortran files only compute the updated $\eta_{t+\Delta t}$ for four species (i.e., E90, DMS, SO2, and H2SO4) and do so using an explicit Euler solver. TChem uses a different input file for the stratosphere than the troposphere, which only includes these stratospheric species and their associated reactions. The input file for the stratosphere has seven species, including the previous four plus three invariant species: OH, NO3, and M. Furthermore, there are five reactions, three of which are of Arrhenius type, and two are JPL-Troe.

Figure 3-4 shows the net production rates, or RHSs, for the four active species in the stratosphere. TChem's net production rates have an RRMSE lower than 1×10^{-6} for SO2 and H2SO4. However, RRMSE is larger than 1×10^{-6} for E90 and DMS. For DMS and E90, CAMPP's generated code computes the RHS in the explicit Euler solver according to

$$\text{RHS} = \text{prod} + \text{ind\_prd} - \frac{\eta_t \exp\left[-\Delta t \frac{\text{loss}}{\eta_t}\right] - \eta_t}{\Delta t}, \tag{3.2}$$

while TChem computes these RHSs according to

$$\text{RHS} = \text{prod} + \text{ind\_prd} - \text{loss}, \tag{3.3}$$

for all species.

Despite the discrepancies between TChem and E3SM in the RHS for the stratosphere, $\eta_{t+\Delta t}$ has RRMSE lower than 5×10^{-4} for all species (see Figure 3-5). Furthermore, For SO2 and H2SO4 that have RRMSE lower than 1×10^{-6}, the corresponding RRMSE for $\eta_{t+\Delta t}$ is lower than 2×10^{-10}.

3.2. Computational performance comparison between TChem and E3SM

To compare the computational performance of E3SM and TChem, we measure the wall time for a time-integration step in both the standalone driver and TChem. We first measure the wall time for the following Fortran subroutines: `setrxt()`, `usrrxt()`, `adjrxt()`, `O1D_to_2OH_adj()`, `phtadj()`, `imp_sol()`, `exp_sol()` of the CAMPP-generated code running within an E3SM simulation (see Table 2-2 for subroutine details). We plot the total sum of the wall times for these subroutines in Figure 3-6, and we can see that the standalone driver's wall times lie within the same range as those of E3SM. Because the E3SM built-in timer reports summary statistics (min, max, and total) for wall times over processor and time steps, we cannot directly compare E3SM's wall time measurements with those of TChem. Thus, we employ wall time measurements from the standalone driver to compare the performance of the E3SM chemistry solver and TChem. In both solvers, we use the same initial conditions (kinetic model, and η_t) and the same number of time integration evaluations. The initial conditions were extracted from an E3SM simulation and corresponded to a chunk with 15 columns, each composed of 80 levels. Note that the wall-time measurements include both the troposphere and stratosphere.

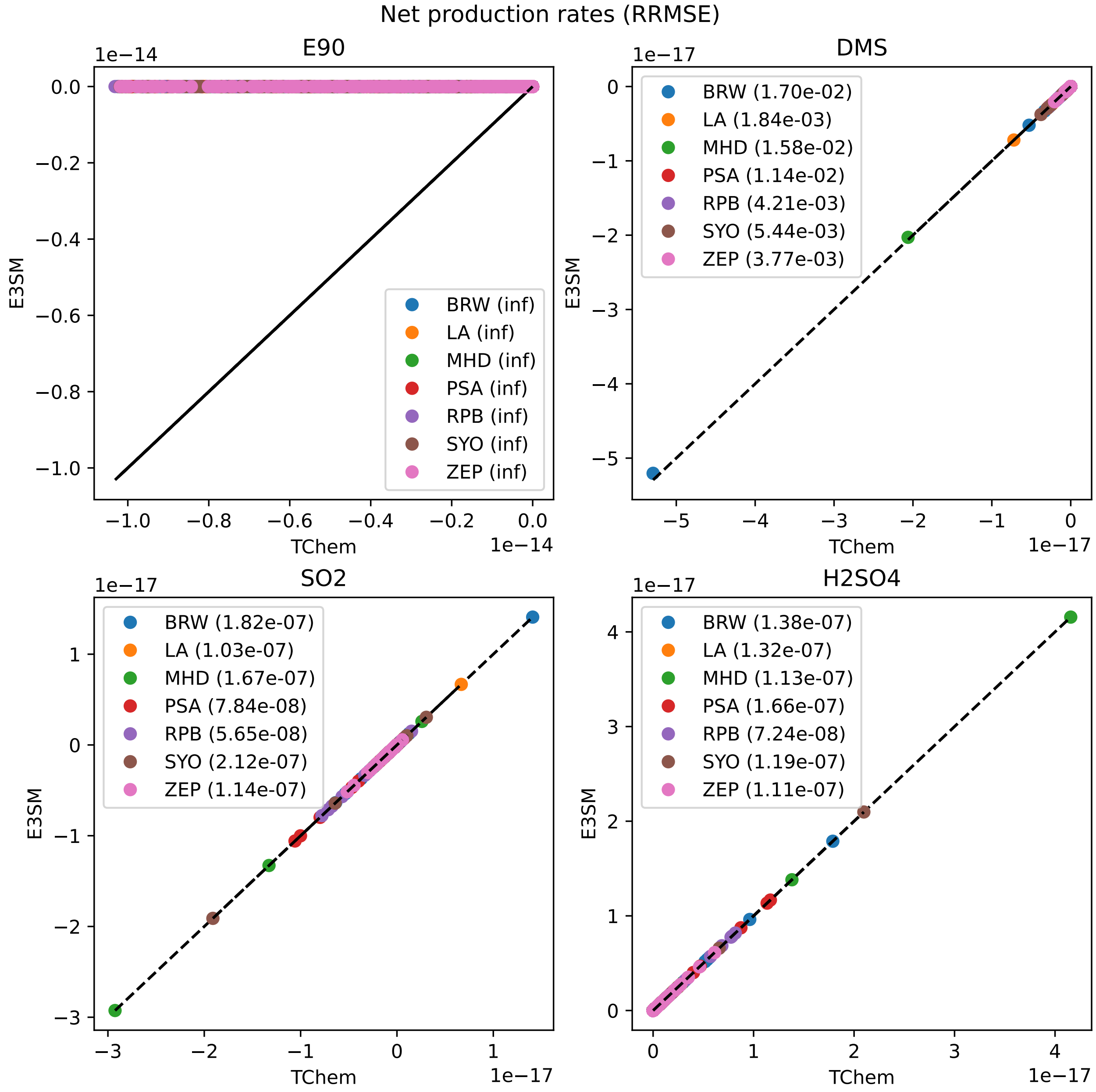

Figure 3-4. Parity plot of net production rates(RHS) in the stratosphere.

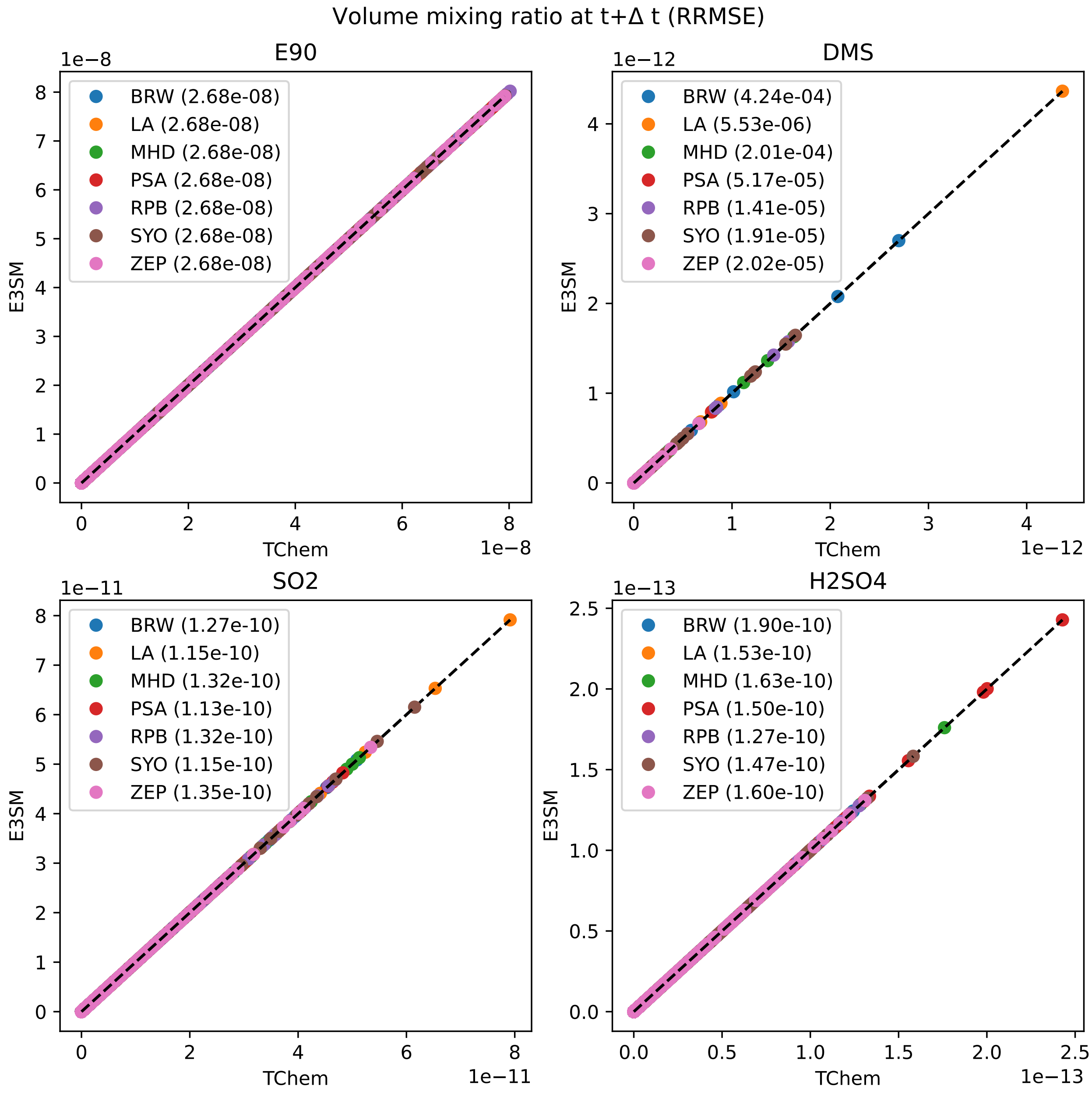

Figure 3-5. Parity plot for updated volumetric mixing ratio in the stratosphere.

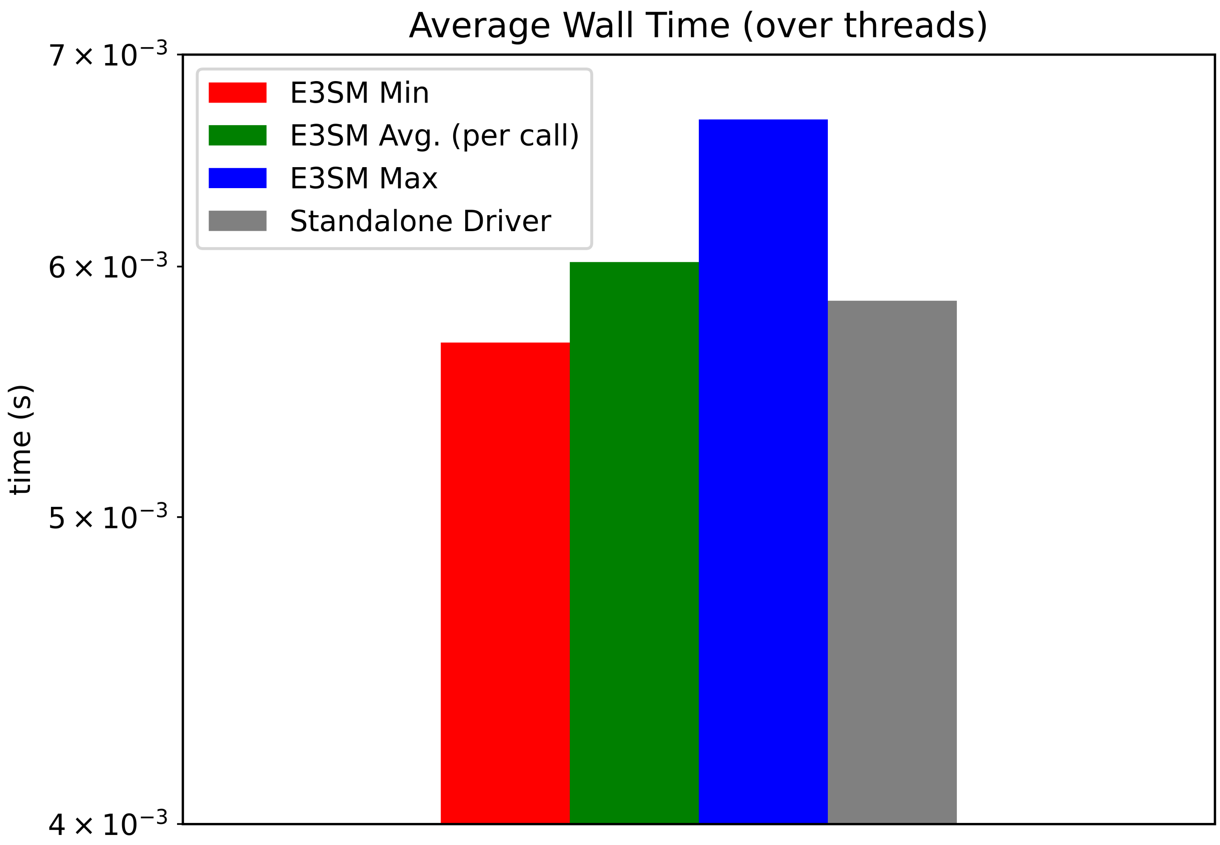

Figure 3-6. Wall time measurements for CAMPP-generated code in both an E3SM simulation and the standalone driver.

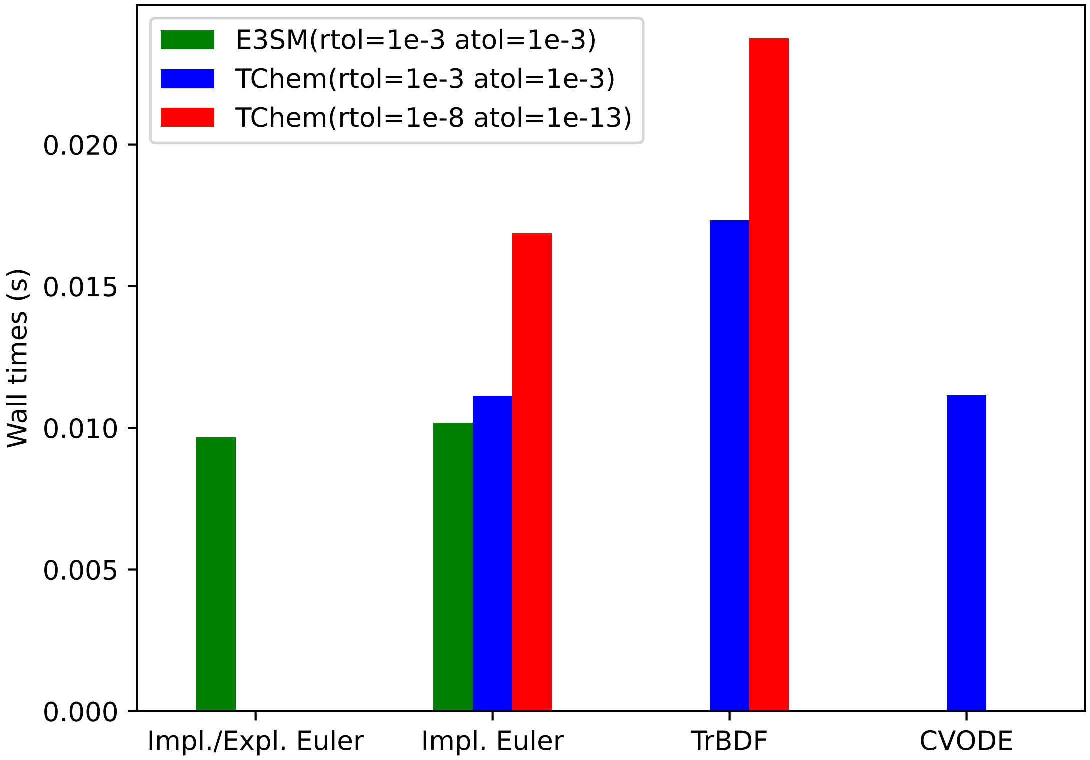

Figure 3-7. Wall time measurements on a POWER 9 architecture (CPU) for a chunk of 15 columns and 80 levels for TChem vs. CAMPP-generated code (E3SM). TChem runs on 160 threads, and the E3SM-standalone driver employs one thread. The implicit/explicit Euler solvers are used in the E3SM simulation when species are split into implicit and explicit classes.

We compare the wall times for TChem and the standalone driver (E3SM) in Figure 3-7. In this Figure, we present the total wall time over a time step for a chunk-size of 15 columns and 80 levels. Therefore, in TChem, we also measure the wall time for a time step in the troposphere and stratosphere. In this Figure, we present wall times for the different ODE solvers.

One fact to note is that TChem is configured at runtime, as opposed to the E3SM/CAMPP generated code, so there is added overhead due to conditional statements, memory access, and parallelism. Despite these additional contributions to run time, the TChem implicit Euler wall times are quite close to E3SM's decoupled implicit/explicit Euler results. The wall time measurements are significant for TChem with the TrBDF solver because its implementation requires additional computation to estimate the time step. However, CVODE's wall times are also quite close to those of E3SM. Finally, as expected, if we decrease the solver tolerance in TChem, wall times increase as the number of Newton iterations increases; however, the slowdown is modest for the appreciable gain in accuracy.

3.2.1. *TChem performance scaling*

We measure the computational performance of TChem for calculating updated volumetric mixing ratios in the UCI mechanism, and we vary the number of evaluations from ≈ $100 - 10^6$. In Figure 3-8, we see the results of measuring wall time for the UCI troposphere calculations. In TChem, we employ the TrBDF2 solver, implicit Euler, with and without an analytical Jacobian computed via the Sacado library [1] for both CPU (Power9) and GPU (NVIDIA V100). We also profile results using the CVODE solver, though only for CPU. We see in this Figure that superior performance can be achieved on the GPU for $\geq 10^4$ evaluations, particularly for the numerical Jacobian cases. On CPU, the implicit-Euler/TrBDF2 with Sacado performs better than the implicit Euler/TrBDF2 without Sacado. Finally, the CVODE case displays the best performance when 160 threads are used on Power9, due to its natural affinity for hyperthreading.

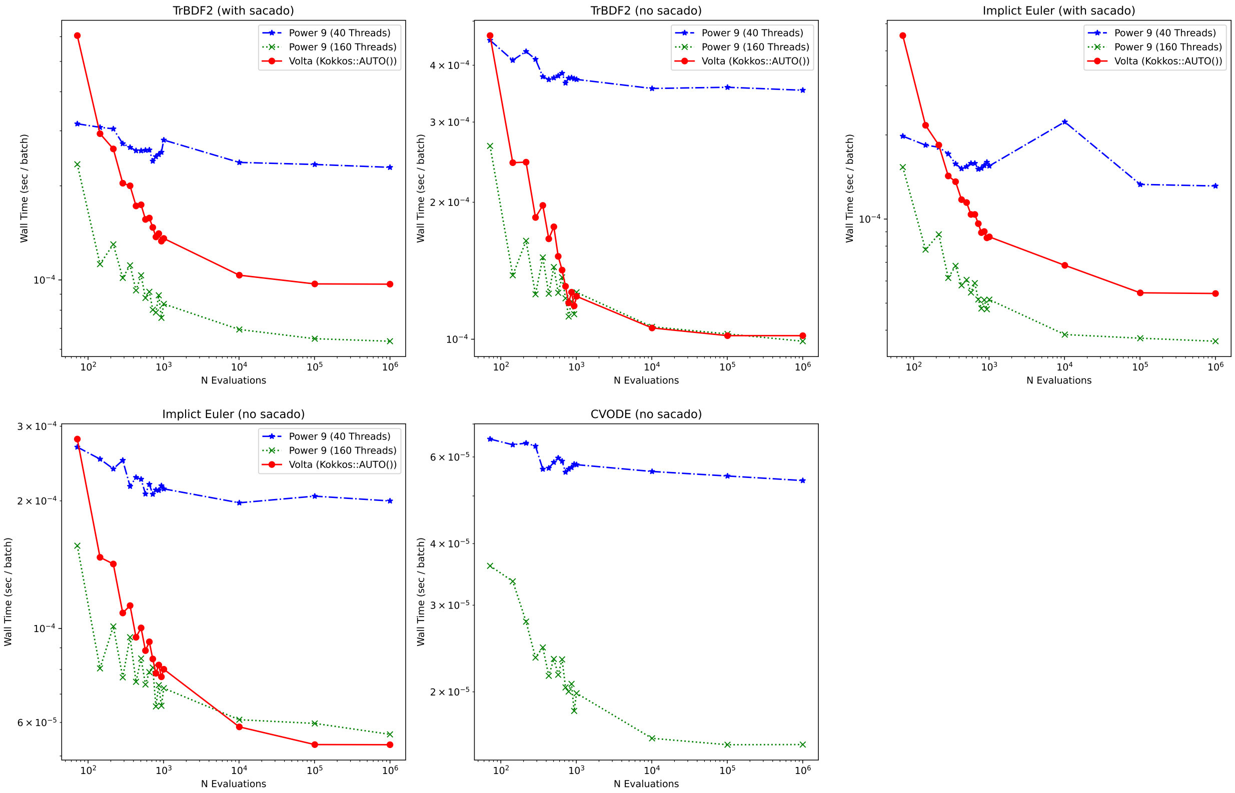

Figure 3-8. Scaling plots for TChem on Power9 (CPU) and NVIDIA V100 (GPU). Here, wall time corresponds to a single time step for of troposphere chemistry.

We also compare wall time profiles for POWER9 (CPU) runs for all ODE solvers in Figure 3-9. The best performance is obtained using CVODE, and the worst is for TrBDF2 with a numerical Jacobian (no Sacado). On CPU, calculating an analytical Jacobian (with Sacado) improves TChem performance results.

We compare the wall time profiles for runs on NVIDIA V100 GPUs for all ODE solvers in Figure 3-9. In the GPU case, the implicit solver displays better performance than TrBDF2. Also, employing an analytical Jacobian, via Sacado, does not result in significant speedup for the GPU case.

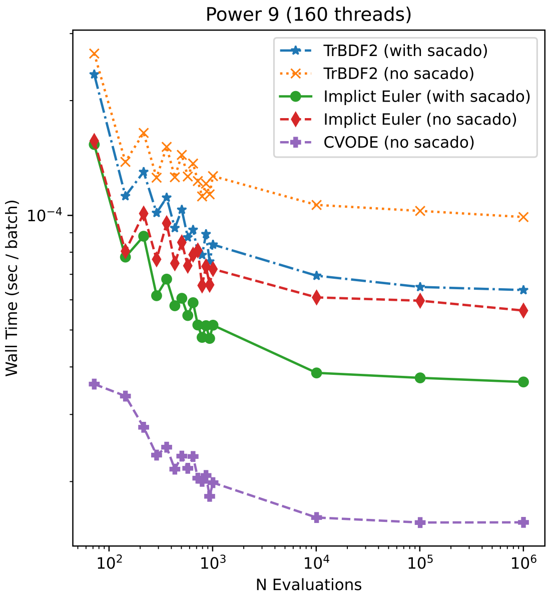

Figure 3-9. Scaling plots for TChem with different ODE solvers on Power9 architecture (CPU) for the UCI mechanism in the troposphere.

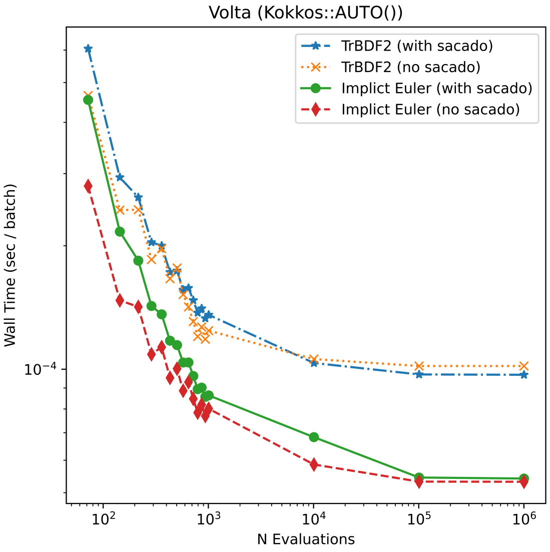

Figure 3-10. Scaling plots for TChem with different ODE solvers on the NVIDIA V100 architecture (GPU) for the UCI mechanism in the troposphere.

This page intentionally left blank.

4. ANTICIPATED OUTCOMES AND IMPACTS

This project generated critical performance data to assess TChem's viability to replace CAMPP in E3SM. In pursuit of this goal, we produced the following outcomes:

- We designed a standalone driver for the CAMPP-generated solver to facilitate direct comparison with TChem results.
- We generated a TChem input file for the UCI chemistry mechanism in both the troposphere and stratosphere.
- We upgraded TChem with features required to model the UCI mechanism that will be used in E3SM v3.
- We verified the TChem/UCI implementation against the E3SM chemistry solver using E3SM datasets.
- We ran TChem on modern computing platforms (Power9 and NVIDIA V100) to generate performance benchmarks.
- We compared the performance of TChem against E3SM solvers with favorable results.
- We upgraded TChem to provide a tool that can be readily added to E3SM for gas chemistry calculations.
    - TChem is a chemistry solver written in C++ /Kokkos that runs on GPUs.
    - TChem accepts runtime-configurable kinetic parameters (not hard-coded).
    - TChem provides a flexible approach for switching reaction mechanisms without writing new code or recompiling.
    - TChem provides the ability to choose among different ODE solvers, and can adapt to future developments.
        - Additionally, the user has precise, arbitrary control over error tolerances in these solvers.

Finally, we will release an upgraded version of TChem in an open-source library that is custom-tailored for atmospheric chemistry and includes all necessary features to solve current and forthcoming E3SM chemistry mechanisms.

This page intentionally left blank.

5. CONCLUSION

This quick targeted investigation (QTI) demonstrates that TChem is a viable replacement for E3SM's current atmospheric chemistry solver, CAMPP. We showed that TChem can reproduce E3SM's gas chemistry outputs, particularly for the UCI mechanism, the current mechanism for E3SM v3. After implementing the needed reaction types and creating the TChem inputs file for the UCI mechanism, we compared the outputs of TChem and E3SM for the volumetric mixing ratio at the next time step ($\eta_{t+\Delta t}$), rate of progress for each reaction, and net production rate of each species (or RHS of vmr for each species) using data inputs produced by E3SM simulations of several locations around the globe. We verified that the relative root means square error (RRMSE) of the difference between the TChem and E3SM's outputs are lower than $\approx 2\times10^{-7}$ for the net production rate and rate of progress. Even though we implemented an implicit Euler solver in TChem, the RRMSE for $\eta_{t+\Delta t}$ is larger than RRMSE for net production rates and rate of progress, which is due to differences (e.g., error tolerances) in the internal newton solver between these two approaches.

Second, the computational performance of TChem is similar to E3SM's chemistry solver. As we mentioned in the report, the E3SM's chemistry solver relies on CAMPP's generated Fortran kinetic files. It is well known that chemistry solvers with pre-generated kinetic files have faster wall times than solvers like TChem, which constructs RHSs and Jacobians at simulation time. Therefore, the performance of these two approaches is similar due to the more granular parallelization in TChem. Future work could include a module in TChem to produce generated C++/Kokkos code for gas chemistry mechanisms. However, the necessity/likelihood of this work is conditional to the relative computational cost of the gas chemistry in E3SM; to our knowledge, gas chemistry computations in E3SM are not one of the most expensive tasks. Third, we also report performance measurements for two additional solvers, TrBDF2 and CVODE. We note that the computational performance TChem with CVODE is close to that of E3SM. Finally, we present scaling plots for TChem, run on an NVIDIA V100 GPU, where superior performance is obtained for more than 10^4 evaluations. In summary, we have presented evidence that TChem is a reasonable choice of chemistry solver for E3SM because it can reproduce current chemistry solver outputs, performs similarly to the current chemistry solver, and executes on GPUs.

This page intentionally left blank.

REFERENCES

[1] SACADO. `https://docs.trilinos.org/dev/packages/sacado/doc/html/index.html`, 2021. Accessed: 2021-04-02.

[2] D. J. Gardner, D. R. Reynolds, C. S. Woodward, and C. J. Balos. Enabling new flexibility in the SUNDIALS suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 2022.

[3] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396, 2005.

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Amanda Barry	01901	anbarry@sandia.gov
Andrew Salinger	01446	agsalin@sandia.gov
Technical Library	1911	sanddocs@sandia.gov

Hardcopy—Internal

Number of Copies	Name	Org.	Mailstop
1	L. Martin, LDRD Office	1910	0359

This page intentionally left blank.

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.