

**Integration of Rosenbrock-type solvers into CAM4-Chem and
evaluation of its performance in the perspectives of science and
computation**

**A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville**

**Jian Sun
May 2018**

Copyright © 2018 by Jian Sun
All rights reserved.

DEDICATION

To my beloved family who provide me endless love and support,
To my advisor and all committee members who make this dissertation happen,
To all my dear friends who accompany me through the past years,
And most especially to our Almighty Lord our God,
This dissertation is dedicated to you.

ACKNOWLEDGMENTS

I would like to first thank my advisor, Dr. Joshua S. Fu, for his help in my research since 2012. Encouraged by his motto “read & search”, I started to learn how to be a qualified researcher in this area. He is willing to find useful resources to support my study and promote the results when possible. In addition, his communication skills and leadership in a team taught me how to collaborate with others, which is very important for my future career. As one of my committee members, Dr. John B. Drake has been intensively involved in my research study and thus delayed his retirement. His knowledge about model framework, mathematics and scientific computation has provided constructive suggestions, resolved my problems and advanced my research. Dr. Jack Dongarra is a world-known scientist in the area of high performance computing. His knowledge about graphics processing unit (GPU) contributes significantly to part of my dissertation and potentially the development of the next-generation model. Dr. Chris Cox is our department head and he generally has a tight schedule. However, he is always willing to support my research when necessary and his suggestion has greatly improved my dissertation. Dr. Jean-Francois Lamarque from the National Center for Atmospheric Research also served as my committee member remotely. He has kindly helped me on the simulation of chemistry-climate model and resolved related issues. I really appreciate his efforts to promote my research by arranging presentations and forwarding job opportunities. I would like to greatly thank my team members including Yang Gao, Kan Huang, Xinyi Dong, Melissa R. Allen, Cheng-en Yang, Jiani Tan and Qingzhao Zhu for their sincere support in my research and personal life. Last but not the least, I would like to thank all the members from the UTK Bible Study Group for their prayers when I’m weak and their reminding when I’m proud. They made my life more colorful and meaningful during my pursuit of degree here. More importantly, they helped me to know the Lord, who is always a lamp unto my feet and a light unto my path. In him I trust and there is no fear in his love. There are many friends who ever helped, but I just cannot list them all here. However, I bear all their kindness in mind and may God bless them forever!

ABSTRACT

In this study, the perennial problem of overestimation of ozone concentration from the global chemistry-climate model (CAM4-Chem [Community Earth System Model with chemistry activated]) is investigated in the sense of numerics and computation. The high-order Rosenbrock-type solvers are implemented into CAM4-Chem, motivated by its higher order accuracy and better computational efficiency. The results are evaluated by comparing to the observation data and the ROS-2 [second-order Rosenbrock] solver can reduce the positive bias of ozone concentration horizontally and vertically at most regions. The largest reduce occurs at the mid-latitudes of north hemisphere where the bias is generally high, and the summertime when the photochemical reaction is most active. In addition, the ROS-2 solver can achieve $\sim 2\times$ speed-up compared to the original IMP [first-order implicit] solver. This improvement is mainly due to the reuse of the Jacobian matrix and LU [lower upper] factorization during its two-stage calculation. In order to gain further speed-up, we port the ROS-2 solver to the GPU [graphics processing unit] and compare the performance with CPU. The speed-up of the GPU version with the optimized configuration reaches a factor of $\sim 11.7\times$ for the computation alone and $\sim 3.82\times$ considering the data movement between CPU and GPU. The computational time of the GPU version increases more slowly than the CPU version as a function of the number of loop iterations, which makes the GPU version more attractive for a massive computation. Moreover, under the stochastic perturbation of initial input, we find the ROS-3 [third-order Rosenbrock] solver yields better convergence property than the ROS-2 and IMP solver. However, the ROS-3 solver generally provides a further overestimation of ozone concentration when it is implemented into CAM4-Chem. This is due to the fact that more frequent time step refinements are involved by the ROS-3 solver, which also makes the ROS-3 solver less computationally efficient than the IMP and ROS-2 solvers. We also investigate the effect of grid resolution and it shows that the fine resolution can provide relatively better pattern correlation than the coarse resolution, given the same chemical solver.

TABLE OF CONTENTS

CHAPTER I INTRODUCTION.....	1
1.1 Background	1
1.2 Current Challenge.....	1
1.3 Goals	2
CHAPTER II IMPROVED COMPUTATIONAL PERFORMANCE AND PREDICTION OF SURFACE OZONE CONCENTRATION CONTERMINOUS U.S. FROM 2ND-ORDER ROSENBROCK SOLVER	4
Abstract	5
2.1 Introduction	6
2.2 Methodology	9
2.2.1 Chemistry configuration in CAM4-Chem	9
2.2.2 Second-order Rosenbrock (ROS-2) solver description	10
2.2.3 Time step size setting	11
2.2.4 Algorithm and implementation.....	12
2.3 Results and Discussion	14
2.3.1 Numerical analysis of the first-order implicit solver and the ROS-2 solver	14
2.3.2 First-order implicit solver and ROS-2 solver at 1800 seconds time step	18
2.3.3 First-order implicit solver with 180 and 1800 seconds time step	23
2.3.4 ROS-2 solver with 180 and 1800 seconds time step.....	29
2.3.5 Evaluation of surface ozone concentration at global scale.....	31
2.3.6 Computational efficiency	35
2.4 Conclusion	41
References	45
CHAPTER III COMPUTATIONAL BENEFIT OF GPU OPTIMIZATION FOR THE ATMOSPHERIC CHEMISTRY MODELING	52
Abstract	53
3.1 Introduction	54

3.2 Methodology	56
3.2.1 Chemistry configuration in CAM4-Chem	56
3.2.2 Architecture	57
3.2.3 Memory layout.....	58
3.3 Results and Discussion	60
3.3.1 Basic analysis	60
3.3.2 CPU	62
3.3.3 GPU	68
3.4 Conclusion	82
References	85
CHAPTER IV COMPARISON OF HORIZONTAL AND VERTICAL DISTRIBUTION OF	
OZONE BETWEEN THE SECOND-ORDER ROSENBROCK (ROS-2) AND THIRD-	
ORDER ROSENBROCK (ROS-3) SOLVERS IN CAM4-CHEM WITH DIFFERENT	
HORIZONTAL RESOLUTIONS	90
Abstract	91
4.1 Introduction	92
4.2 Methodology	94
4.2.1 Global chemistry-climate model	94
4.2.2 Description of chemical solvers.....	95
4.2.3 Port Validation modules in CAM.....	98
4.3 Results and Discussion	99
4.3.1 Perturbation of initial concentration	99
4.3.2 Perturbation of reaction rates	102
4.3.3 Model evaluation	105
4.3.4 Effect of grid resolution	113
4.3.5 Computational cost	115
4.4 Conclusion	122
References	125
CHAPTER V CONCLUSION	136
5.1 Summary of this study	136

5.2 Summary from CHAPTER II	137
5.3 Summary from CHAPTER III	138
5.4 Summary from CHAPTER IV	139
CHAPTER VI FUTURE WORK	141
VITA	142

LIST OF TABLES

Table 2.1. Numerical Analysis of the First-Order Implicit Solver and Second-Order Rosenbrock (ROS-2) Solver for a Linear System.....	16
Table 2.2. Numerical analysis of the first-order implicit solver and ROS-2 solver for a non-linear system with box-model simulations (units: mol/mol).	19
Table 2.3. Student's t-test for bias (ROS-2_1800s minus ORI_1800s, units: ppb) between the ROS-2 solver and the first-order implicit solver at 1800 seconds time step in each state.	26
Table 2.4. Statistics for the first-order implicit solver at 180 seconds time step (ORI_180s) and 1800 seconds time step (ORI_1800s) in each state, respectively.	32
Table 2.5. Statistics for the ROS-2 solver at 180 seconds time step (ROS-2_180s) and 1800 seconds time step (ROS-2_1800s) in each state, respectively.	36
Table 2.6. Summary of computational time (units: seconds) per processor for each solver (ORI_180s, ORI_1800s, ROS-2_180s and ROS-2_1800s) with various simulation periods.	42
Table 3.1. The floating point operations (FLOP) and data copy (unit: KB) per loop iteration for the main functions of the chemistry box model.	62
Table 4.1. Coefficients for the third-order Rosenbrock method.	98
Table 4.2. Summary of summed computational time (units: hours) for each solver (IMP: first-order implicit solver, ROS-2: second-order Rosenbrock solver, ROS-3: third-order Rosenbrock solver) with different horizontal resolutions.	121

LIST OF FIGURES

Figure 2.1. The sparsity pattern of left-hand side system matrix from the second-order Rosenbrock method (ROS-2) for the TROP_MOZART mechanism (a) before and (b) after permutation.....	14
Figure 2.2. Difference of 10 year averaged (a) annual and (b) summertime (June, July, and August, JJA) mean surface ozone concentration (units: ppb) over the conterminous U.S. (CONUS) between the ROS-2 solver (ROS-2_1800s) and the first-order implicit solver (ORI_1800s) at 1800 s time step (ROS-2_1800s minus ORI_1800s).....	25
Figure 2.3. 10 year averaged (a) annual and (c) summertime mean surface ozone concentration bias (units: ppb) over the CONUS between the ORI_1800s and the AQS observation data (ORI_1800s minus AQS). (b, d) The same but for the ROS-2_1800s.	28
Figure 2.4. Difference of 10 year averaged (a) annual and (b) summertime mean surface ozone concentration (units: ppb) over the CONUS between the first-order implicit solver at 180 seconds time step (ORI_180s) and the ORI_1800s (ORI_180s minus ORI_1800s).	30
Figure 2.5. Difference of 10 year averaged (a) annual and (b) summertime mean surface ozone concentration (units: ppb) over the CONUS between the ROS-2 solver at 180 seconds time step (ROS-2_180s) and the ROS-2_1800s (ROS-2_180s minus ROS-2_1800s).....	39
Figure 2.6. Difference of 10 year averaged (a) annual and (b) JJA mean surface ozone concentration (units: ppb) between the ROS-2_1800s and the ORI_1800s (ROS-2_1800s minus ORI_1800s) over the global continents.....	39
Figure 3.1. Memory bandwidth (unit: GB/s) for (a-c) GNU, (d-f) Intel and (g-i) PGI compilers using different memory layouts (left panel: strided memory layout; middle panel: fully interleaved memory layout; right panel: block interleaved memory layout). Different colors refer to the different functions in the chemistry box model (All: the whole chemistry box model; Init: data initialization; Jacob: formation of the	

Jacobian matrix; LF: LU factorization; LS: LU solve; Others: update of intermediate and final solutions; RHS: formation of the right hand side source term).....	63
Figure 3.2. The same as Figure 3.1 but for the computational rate (unit: GFLOPS).	65
Figure 3.3. The total wall-clock time (Y-axis, log scale, unit: second) of the chemistry box model with different number of loop iterations (X-axis, log scale) for (a) GNU, (b) Intel, and (c) PGI compiler. Different colors refer to different memory layouts (blue: strided, red: fully interleaved, black: block interleaved).	67
Figure 3.4. The percent contribution (Y-axis) of each function in the chemistry box model to the total wall-clock time with different number of loop iterations (X-axis) for different compilers ((a-c): GNU, (d-f): Intel and (g-i): PGI) and memory layouts (left-panel: strided, middle-panel: fully interleaved, right-panel: block interleaved).	70
Figure 3.5. The total wallclock time (Y-axis, log scale, unit: second) of the GPU version of chemistry box model with different number of loop iterations (X-axis, log scale) using the (a) strided, (b) fully interleaved and (c) block interleaved memory layouts. Different colors refer to different metrics of time (red: computational time, blue: data movement time between CPU and GPU, green: data deallocation time on the GPU, black: summed time from the three above).	72
Figure 3.6. The percent contribution (Y-axis) of each function in the GPU version of chemistry box model to the total wall-clock time with different number of loop iterations (X-axis) for the (a) strided, (b) fully interleaved and (c) block interleaved memory layout.....	72
Figure 3.7. The computational time (Y-axis, log scale, unit: second) of the GPU version of chemistry box model with different number of threads per block (X-axis), using the fully interleaved memory layouts (blue: 448 loop iterations; red: 7,168 loop iterations).	73
Figure 3.8. The percent contribution (Y-axis) of each function in the GPU version of chemistry box model to the total wall-clock time with different number of threads per block (X-axis) for (a) 448 and (b) 7,168 loop iterations.	74

Figure 3.9. The NVIDIA visual profiler (NVVP) results for the GPU version of chemistry box model using the fully interleaved memory layout for (a) multiple kernels and (b) one kernel.	75
Figure 3.10. The computational time (Y-axis, log scale, unit: second) of the GPU version of chemistry box model with different number of loop iterations (X-axis) for the multiple kernels (blue) and one kernel (red).	75
Figure 3.11. The computational time (Y-axis, log scale, unit: second) of the GPU version of chemistry box model with different number of loop iterations (X-axis) using the shared memory for the solution vector (blue), the shared memory for the intermediate solution vectors (red) and no shared memory (black).....	77
Figure 3.12. The computational time (Y-axis, log scale, unit: second) of the GPU version of chemistry box model with different number of loop iterations (X-axis) using the shared memory for the intermediate solution vectors (blue), shared memory plus constant memory (green), no shared memory (black) and no shared memory but constant memory (red).	78
Figure 3.13. The computational time (Y-axis, log-scale, unit: second) of the GPU version of chemistry box model with different number of loop iterations (X-axis) for the multiple kernels (blue), one kernel with constant memory (black) and streamed multiple kernels with constant memory (red).	80
Figure 3.14. The NVVP results for the GPU version of chemistry box model using the streamed multiple kernels + constant memory with (a) 448 and (b) 7,168 loop iterations.	80
Figure 3.15. The data movement time (Y-axis, log scale, unit: second) of the GPU version of chemistry box model with different number of loop iterations (X-axis) using the separate memory allocation (blue), the contiguous memory allocation (red) and the pinned memory (black).	83
Figure 3.16. The total wall-clock time (Y-axis, log scale, unit: second) of the chemistry box model with different number of loop iterations (X-axis). Different colors refer to different metrics of time (black: total wall-clock time of CPU version, red: total wall-	

clock time of GPU version, blue: time of computation alone, green: time of data movement between CPU and GPU, cyan: time of data deallocation on the GPU). 83

Figure 4.1. Boxplot of ozone concentrations after one climate time step integration (1800 seconds) from the first-order implicit (IMP) solver (black), the second-order Rosenbrock (ROS-2) solver (blue) and the third-order Rosenbrock (ROS-3) solver (red) with different time step sizes. X-axis shows the time step size (unit: second) and Y-axis shows the ozone concentration (unit: ppb). The dots refer to the mean values. (a) to (i) correspond to the scenarios with $\pm 10\%$, $\pm 20\%$, $\pm 30\%$, $\pm 40\%$, $\pm 50\%$, $\pm 60\%$, $\pm 70\%$, $\pm 80\%$ and $\pm 90\%$ perturbation of initial species concentrations, respectively. 103

Figure 4.2. The same as Figure 1 but for the perturbation of initial reaction rates. 107

Figure 4.3. Vertical distributions (median) of ozone concentration (unit: ppb) averaged for the whole year (solid line) and June-July-August (JJA, dash line) from 1995 to 2004 for the troposphere. Different colors mean the ozone soundings (black), IMP solver (green), ROS-2 solver (blue) and ROS-3 solver (red). 111

Figure 4.4. Taylor diagram for regions at the High Latitudes (left panel), Mid-Latitudes (middle panel), and Tropics (right panel) with three different altitudes (top row: 1000 hPa, middle row: 500 hPa, bottom row: 250 hPa) in the troposphere. The 10-year results for the three chemical solvers generated at the $0.47^\circ \times 0.625^\circ$ (0.5 degree), $0.94^\circ \times 1.25^\circ$ (1.0 degree) and $1.9^\circ \times 2.5^\circ$ (2.0 degree) horizontal resolution are compared to the ozonesonde climatology data. The radial distance from the origin is the ratio of the standard deviation of simulated ozone concentrations over the standard deviation of ozonesonde climatology data. The azimuthal position is the correlation coefficient of seasonal median ozone concentrations between the simulated ozone concentrations and the ozonesonde climatology data. Different colors mean different chemical solvers (blue: ROS-2 solver, green: IMP solver, red: ROS-3 solver). 116

Figure 4.5. 10-year averaged seasonal median ozone concentration (y-axis, unit: ppb) for regions in the North Mid-Latitudes at 1000 hPa (top row), 500 hPa (middle row), and 250 hPa (bottom row). The x-axis is the season (winter: December-January-

February (DJF), spring: March-April-May (MAM), summer: June-July-August (JJA), fall: September-October-November (SON)). Different line patterns mean different resolutions (solid: 0.5°, long dash: 1.0°, short dash: 2.0°) and different colors mean different chemical solvers (blue: ROS-2 solver, green: IMP solver, red: ROS-3 solver). The vertical error bars refer to the one standard deviation.	118
Figure 4.6. Difference of 10-year averaged seasonal mean temperature (unit: K) between the 0.5° (regridged to 2.0°) and 2.0° at 1000 hPa (top row), 500 hPa (middle row), and 250 hPa (bottom row).	120
Figure 4.7. The same as Figure 6 but for the difference between the 1.0 degree (regridged to 2.0°) and 2.0°.....	120
Figure 4.8. Monthly mean ozone concentrations at different altitudes for different resolutions without lightning NO _x emission.	121

CHAPTER I INTRODUCTION

1.1 Background

Processes including dynamics, physics and chemistry are closely coupled in the earth system modeling. Since most radiatively active compounds (e.g., CH₄, O₃ and black carbon) in the current Earth's atmosphere are chemically active, atmospheric chemistry is highlighted as an essential component of climate [Tian and Chipperfield, 2006; IPCC AR5 Chapter 8, 2013]. It defines the homogeneous (e.g., gas-phase species) and heterogeneous (e.g., gas and aerosol) reactions, black carbon and acid (nitrogen + sulfur) deposition and cloud-aerosol interactions in the atmosphere. The burden and residence time of chemically active compounds is generally affected by the (1) reaction with other species (e.g., gaseous species, aerosols and water) in its immediate vicinity, (2) photolysis (interaction with solar radiation) and (3) deposition process (wet and dry removal).

1.2 Current Challenge

Due to the complex chemical mechanism and strong variability of temporal and spatial pattern, atmospheric chemistry is usually characterized by the existence of significant non-linearity (Kleinman et al., 2001) and a wide range of scales (Isaksen et al., 2009), which makes it a challenge to be well represented by the model. Scientists have made tremendous efforts to address these issues during the development of global chemistry-climate models but observable deficiencies still remain. For example, the bias of surface ozone concentration is a well-known problem and numerous studies are done to explore the potential reasons. The coarse horizontal resolution is partially responsible for the enhanced ozone production [Kumar et al., 1994], since the emissions of ozone precursors are artificially diluted [Sillman et al., 1990; Liang and Jacobsen, 2000] when the model grid size increases. The coarse vertical resolution is also likely to increase

ozone production when the emissions are distributed through the depth of the boundary layer [Murazaki, K. and P. Hess, 2006]. The failure to resolve the steep sub-grid land-to-sea gradients in mixing depth and complex topography that controls the airflow and ventilation could lead to an overestimate of surface ozone concentration along the coastal regions [Fiore et al., 2002]. The existence of heterogeneous reactions of aerosols can enable efficient gas-particle mass transfer and thus influence the ozone production as well [Martin et al., 2003]. Kim et al. [2015] indicate that an underestimate of the cloud fraction would contribute to the overestimate of photolysis rate and thus higher prediction of surface ozone concentration.

In addition, the atmospheric chemistry determines the formation of aerosols, which will influence the aerosol-cloud-precipitation interaction in the microphysics process [Andreae and Rosenfeld, 2008; Stevens and Feingold, 2009; Rosenfeld et al., 2014; Fan et al., 2016; Seinfeld et al., 2016]. The radiative forcing is strongly influenced by the aerosols through direct (i.e., scattering and absorption of radiation) and indirect (i.e., aerosol-cloud interaction, aerosols serve as condensation and ice nucleation sites to form cloud droplets and ice particles) radiative effect [Stier et al., 2007; Stevens, 2015]. Large uncertainty still exists in the magnitude and sign of aerosol-related radiative forcing [IPCC AR5 Chapter 7, 2013; Lee et al., 2015] and partial uncertainty is attributed to the anthropogenic emission from human activity [Hamilton et al., 2014; Kodros et al., 2015]. The sophisticated mechanism for the formation and evolution of secondary organic aerosol (SOA) in the atmosphere also evidently contributes to the complexity of chemistry-climate model framework [Kroll and Seinfeld, 2008].

1.3 Goals

Overall, it is anticipated that coupled atmospheric chemistry will play a larger role in the future climate simulations and global models will be important for addressing future air quality as well as global warming concerns. Recognizing the poor performance of chemistry in the current earth system modeling and limited studies of investigating the

numerical method used to resolve the atmospheric chemistry, we implement an alternative chemical solver into the state-of-the-art global chemistry-climate model (Community Atmosphere Model with Chemistry (CAM4-Chem)) to examine whether it can provide an improved simulation result. Moreover, as the model has grown in the grid scale and process, the computational requirement to produce long-term simulations (e.g., decadal to century) has significantly increased as well [Slingo et al., 2009; Washington et al., 2009]. With the increased high performance computing power and the coinciding requirement for increased algorithmic parallelism, it will be of great help if the newly implemented method offers better computational efficiency.

The goals of this dissertation include:

- (1) Implement a chemical solver with high order numerical accuracy (e.g., second-order Rosenbrock (ROS-2) solver) into CAM4-Chem and examine its model performance of surface ozone concentration and computational efficiency.
- (2) Port the ROS-2 solver to GPU and compare its computational performance with CPU.
- (3) Investigate the response of first-order implicit solver, ROS-2 solver and third-order Rosenbrock (ROS-3) solver to the stochastic perturbation of initial input.
- (4) Examine the model performance (both horizontal and vertical distribution of ozone concentration) of Rosenbrock-type solvers and the effect of grid resolutions.

**CHAPTER II IMPROVED COMPUTATIONAL PERFORMANCE AND
PREDICTION OF SURFACE OZONE CONCENTRATION
CONTERMINOUS U.S. FROM 2ND-ORDER ROSENBROCK SOLVER**

A version of this chapter was originally published by Jian Sun et al.:

Sun, J., J. S. Fu, J. Drake, J.-F. Lamarque, S. Tilmes, and F. Vitt (2017), Improvement of the prediction of surface ozone concentration over conterminous U.S. by a computationally efficient second-order Rosenbrock solver in CAM4-Chem, *J. Adv. Model. Earth Syst.*, 9, 482–500, doi:10.1002/2016MS000863.

Abstract

The global chemistry-climate model (CAM4-Chem) overestimates the surface ozone concentration over the conterminous US (CONUS). Reasons for this positive bias include emission, meteorology, chemical mechanism and solver. In this study, we explore the last possibility by examining the sensitivity to the numerical methods for solving the chemistry equations. A second-order Rosenbrock (ROS-2) solver is implemented in CAM4-Chem to examine its influence on the surface ozone concentration and the computational performance of the chemistry program. Results show that under the same time step size (1800 seconds), statistically significant reduction of positive bias is achieved by the ROS-2 solver. The improvement is as large as 5.2 ppb in Eastern US during summer season. The ROS-2 solver is shown to reduce the positive bias in Europe and Asia as well, indicating the lower surface ozone concentration over the CONUS predicted by the ROS-2 solver is not a trade-off consequence with increasing the ozone concentration at other global regions. In addition, by refining the time step size to 180 seconds, the first-order implicit solver does not provide statistically significant improvement of surface ozone concentration. It reveals that the better prediction from the ROS-2 solver is not only due to its accuracy, but also to its suitability for stiff chemistry equations. As an added benefit, the computation cost of the ROS-2 solver is almost half of first-order implicit solver. The improved computational efficiency of the ROS-2 solver is due to the reuse of the Jacobian matrix and lower upper (LU) factorization during its multi-stage calculation.

2.1 Introduction

Atmospheric chemistry is important for global climate simulation because of the close coupling of transport, physical, chemical and biological processes. The feedbacks among chemical reactions, climate equilibria, anthropogenic emissions and land-use changes provide a new dynamical perspective on global and regional climate and air quality predictions. Among the atmospheric chemical constituents, tropospheric ozone is a critical pollutant that can significantly affect ecosystems, agriculture productions, public health and climate forcing [Stevenson et al., 1998; Fiscus et al., 2005; Karnosky et al., 2007; Cooper et al., 2010; Sun et al., 2015]. However, ozone is not directly emitted and its complex photochemical reaction mechanism makes its simulation a challenge. Significant bias in the prediction of ozone concentration exists zonally and seasonally for both single model output [Zeng et al., 2008; Lamarque et al., 2012; Val Martin et al., 2015] and multi-model ensemble mean results [Stevenson et al., 2006; Stevenson et al., 2013; Young et al., 2013]. There are also inconsistencies between the estimated ozone concentration from global climate models and the observed seasonal cycle [Fiore et al., 2014]. In the CAM4-Chem model we are using, optimizing the dry deposition scheme based on land use changes has significantly improved the simulation of summertime (June, July and August, JJA) surface ozone concentration over the US [Val Martin et al., 2014; Val Martin et al., 2015]. However, further efforts are required to reduce the remaining bias.

Few studies have investigated the numerical chemical solver itself in the performance of global chemistry-climate models. Shampine [1982] studied initial value problems for stiff systems of ordinary differential equations (ODEs) and proposed an approach to automatically select either an explicit Runge-Kutta formula or Rosenbrock formula at every time step. The Rosenbrock method was competitive with the backward differentiation formulas (BDF) in some circumstances. The sparse matrix vectorized gear code (SMVGEAR) developed by Jacobson and Turco [1994] was also

implemented in global chemistry models such as GMI [Rotman et al, 2001], IMPACT [Rotman et al, 2004], GATOR-GCMOM [Jacobson and Ginnebaugh, 2010; Jacobson et al., 2015] and GEOS-Chem [Zhang et al., 2011]. However, this solver used an iterative method to solve the ODEs and included the re-evaluation of the Jacobian matrix during the iteration, which was intrinsically slower than other solvers using a non-iterative method. Sandu et al. [1997] tested a set of box-model atmospheric chemistry problems (TMK model, CBM-IV model, AL model, a NASA HSRP/AESA stratospheric model and an aqueous model) with different solvers, including LSODE (Livermore Solver for Ordinary Differential Equations) and Variable-coefficient Ordinary Differential Equation solver (VODE) that were expected to perform similarly to SMVGEAR. The benchmark problems covered a wide range of photolytic, homogeneous (gas-phase, liquid-phase) and heterogeneous (gas-liquid) reactions. The results showed that the Rosenbrock solvers were the most cost-effective and performed well for real problems with large variety of conditions that could occur at different grid cells. Verwer et al. [1999] applied a second-order, L-stable Rosenbrock (ROS-2) method to the three-dimension atmospheric reaction and transport problem including photochemistry, advection and diffusion. Three chemistry models (RIVM, CBM-IV and WET) using the ROS-2 method were examined and the ROS-2 method was proven to be an excellent candidate for global air quality modeling with large time steps on the order of minutes. Blom and Verwer [2000] examined different operator splitting methods for the atmospheric transport-chemistry problems and the test results revealed that the Rosenbrock W-method, split at the linear algebra level, was a better option than Strang operator splitting or source splitting. The W-method avoided the artificial stiff transients during the chemistry computation and boundary condition issues for integration in time, though its implementation was complex. Long et al. [2013] developed a coupled chemistry and climate system model by linking the NCAR Community Atmosphere Model (modal-CAM v3.6.33, the atmospheric component of Community Climate System Model (CCSM3)) and the Max Planck Institute for Chemistry's Module Efficiently Calculating the Chemistry of the Atmosphere (MECCA; v2.5) to investigate the multiphase process in the atmosphere. The Kinetics PreProcessor (KPP, Sandu and Sander, 2006) package

was used to provide user-defined solvers and the benchmark inter-comparison among three solvers from the Rosenbrock family, namely ROS-2, ROS-3 and RODAS-3, showed good agreement for ozone and OH radical prediction. Note that only the atmospheric component was considered in the work of Long et al. [2013] and the effect of different solvers in a long-term simulation was not explored.

In this paper we do not achieve improvement by a better design of the sparse matrix. Breaking the data structure in order to introduce some external packages would disable existing optimizations. Instead, we seek to obtain improved performance of chemistry and computation from the algorithm itself. Therefore, based on the literature above, the numerical solvers from the Rosenbrock family are good alternatives for full atmospheric chemistry and climate simulation. These methods have already been incorporated into some regional models such as the Weather Research and Forecasting model coupled with Chemistry (WRF-Chem) and the Community Multi-scale Air Quality Model (CMAQ) [Linford et al., 2009; Sarwar et al., 2013]. But their use in global models is very limited. In this study, the global climate model we use is the Community Earth System Model (CESM) with online chemistry activated (CAM4-Chem [Lamarque et al., 2012]). Currently the chemical solver uses a fully implicit Euler method that gives unconditional stability but only first order accuracy in time [Kinnison et al, 2007]. This could be part of the reason for poor performance of ozone in previous work with CAM4-Chem. To bridge the gap between the need for better estimate of ozone from a scientific perspective and the limitation of a low order accuracy solver in the current CAM4-Chem, the ROS-2 solver is implemented to replace the original chemical solver and tested to see whether there is any benefit for the global climate and chemistry simulation. Unlike the implicit Euler method, the ROS-2 method avoids the re-evaluation and decomposition of the Jacobian matrix. These are the most time-consuming parts in a chemical solver [Daescu et al., 2000]. Hence, we are also interested in the improvement of computational performance of chemistry with the ROS-2 solver. The mathematical formula of the ROS-2 method and its implementation in CAM4-Chem are described in the methodology section. The predictions of surface ozone concentration over the

conterminous US (CONUS) between the ROS-2 solver and the original first-order implicit solver are compared, as well as their computational efficiencies with varying numbers of processors on a massively parallel supercomputer. Finally, we discuss the major differences we have observed, and make further recommendations.

2.2 Methodology

2.2.1 Chemistry configuration in CAM4-Chem

The Community Earth System Model (CESM version 1.2.2) is a state-of-the-art global climate model. It consists of four components: atmosphere, land and land-ice, ocean and sea ice. CAM4-Chem is an implementation of atmospheric chemistry in CESM and its chemistry is fully coupled with the radiative absorption processes of the Community Atmosphere Model, the atmospheric component of CESM (CAM4, [Neale et al., 2013]). The chemistry mechanism in the current CAM4-Chem version is adapted from the standard Tropospheric Model for Ozone and Related chemical Tracers (MOZART-4) [Emmons et al., 2010]. However, more chemistry-specific parameterization such as dry/wet deposition schemes and species types are expanded in the current CAM4-Chem version. The details about their differences, together with CAM4-Chem's representation of atmospheric chemistry in the global model, are well evaluated in previous work [Lamarque et al., 2012]. In total the chemical mechanism includes 212 reactions, with 40 photochemical reactions and 172 gas phase reactions. There are 103 chemical species: 8 species (CH_4 , N_2O , CO , Rn , Pb , H_2 , HCN and CH_3CN) are solved explicitly by the first-order (forward) Euler method and the remaining 95 species are solved implicitly by a backward Euler method with a Newton-Raphson iteration for quick convergence. These 95 species include all the chemically active species such as ozone and OH radicals, which contribute most to the stiffness of the system. In this study the ROS-2 solver will be developed to replace the implicit solver mentioned above. The finite-volume (FV) dynamical core [Neale et al., 2010] is used with a global horizontal

resolution of 0.9 degree (latitude) by 1.25 degree (longitude) and 26 vertical layers top to approximately 3 hPa.

2.2.2 Second-order Rosenbrock (ROS-2) solver description

In the state-of-the-art CAM4-Chem version, an operator-splitting approach is used and the atmospheric chemical reactions are integrated by the stiff ODE solver separately from other processes like dynamics and physics. Thus for the chemical update, each control volume behaves like a box model. The equations for chemical species conservation and reaction in the atmosphere take the form of an autonomous system:

$$\frac{dy}{dt} = F(y) = P(y) - L(y) + I(y) \quad (2.1)$$

where $y = (y_1, y_2, \dots, y_N)^T$ is the vector of volume mixing ratios for N species (N here represents the 95 implicit species) at given latitude, longitude and vertical coordinate; the source term $F(y)$ represents the atmospheric chemical reactions. It can be further decomposed into three components: production $P(y)$, loss $L(y)$ and independent forcing $I(y)$ terms. $P(y)$ and $L(y)$ are mainly calculated by the species mixing ratios and reaction rates, while $I(y)$ is evaluated based on the external forcing (i.e., aircraft and lightning emission). The two-stage, linear-implicit Rosenbrock scheme for the ODE above can be written as Verwer et al. [1999]:

$$(I - h\gamma A)k_1 = F(y^n) \quad (2.2)$$

$$(I - h\gamma A)k_2 = F(y^n + hk_1) - 2k_1 \quad (2.3)$$

$$y^{n+1} = y^n + \frac{3}{2}hk_1 + \frac{1}{2}hk_2 \quad (2.4)$$

where I is an $N \times N$ identity matrix; h is the time step size; $A = \left. \frac{\partial F(y)}{\partial y} \right|_{y=y^n}$ is the Jacobian matrix at time $t = t^n$; y^n and y^{n+1} are the solution vectors of species mixing ratios at time $t = t^n$ and t^{n+1} , respectively. Vectors k_1 and k_2 are the intermediate solutions at each stage. Parameter γ is a constant and appears in the stability function with $z = h\alpha$ for problem $y' = \alpha y$ as:

$$R(z) = \frac{1 + (1-2\gamma)z + (\frac{1}{2} - 2\gamma + \gamma^2)z^2}{(1-\gamma z)^2} \quad (2.5)$$

A solver is defined as A-stable if $|R(z)| \leq 1$ as $z \rightarrow \infty$ and it is further called L-stable if $R(z) \rightarrow 0$ as $z \rightarrow \infty$. The ROS-2 solver proposed above is A-stable for Eq. (2.5) if and only if $\gamma \geq \frac{1}{4}$. Furthermore, by choosing $\gamma = 1 \pm \frac{1}{\sqrt{2}}$ this scheme becomes L-stable ($R(\infty) = 0$), which is good for simulating some chemical species with a short life span in the atmosphere. In practice, $\gamma = 1 + \frac{1}{\sqrt{2}}$ is usually chosen due to its better nonlinear stability behavior under large time steps [Verwer et al., 1999] and this value is thus used in this study. An obvious advantage to the implementation of the ROS-2 solver is that it does not require the re-evaluation of the Jacobian matrix at each stage, while the Newton method requires the re-evaluation of the Jacobian matrix during each iteration. Therefore, it can utilize the same 'lower upper' (LU) factorization result at each stage of the solution update. Since updating the Jacobian matrix and conducting the LU factorization are the most time-consuming operation, this benefit should speed up the chemistry update and save much computation time [Daescu et al., 2000].

2.2.3 Time step size setting

In this work, $h=180$ and 1800 seconds are chosen as two different time step sizes to examine whether the first-order implicit solver and the ROS-2 solver can benefit from the time refinement. It is noted that the previous literature claimed that the ROS-2 solver was able to work under time step size of 600 and 900 seconds [Verwer et al., 1999; Blom and Verwer, 2000]. However, those studies were either working on a simple chemical mechanism or a benchmark simulation without considering the complexity of an earth system model. Furthermore, they agreed that the necessity to form a better conditioned system, resolve the initial transients accurately and handle the non-linear chemistry in the real atmosphere might require a more restricted time step size [Shampine, 1982; Sandu et al., 1997; Verwer et al., 1999; Blom and Verwer, 2000]. In addition, the concepts of A-stability and L-stability are defined for an idealized linear system. The real atmospheric chemistry system is nonlinear and can be very stiff with the existence of fast and slow reacting species. Thus a small time step is usually recommended to make the chemical

reaction system valid for “linearization” (keep the property of linear system) and avoid impacting the quality of numerical integration. Based on those concerns, we implement an adaptive time step method for the ROS-2 solver when the time step size equals 1800 seconds. If the program detects a location where a negative solution is generated by the ROS-2 solver, it refines the time step by a factor of 2 for that location and re-computes the chemistry. This procedure is repeated until the refined time step size is less than 180 seconds, in which case we simply set the time step size to 180 seconds to save computation time. We have found that the ROS-2 solver can run continuously with a fixed 180 seconds time step for the standard full tropospheric chemistry mechanism TROP_MOZART [Emmons et al., 2010].

2.2.4 Algorithm and implementation

The steps of the algorithm are as follows:

- a.** Calculate the independent forcing term, which is treated as invariant as in the original first-order implicit solver design.
- b.** Calculate the linear and nonlinear components of Jacobian matrix to form the left hand side system in Eq. (2.2) and Eq. (2.3). CAM4-Chem uses the same chemical preprocessor as MOZART-4 [Lamarque et al., 2012], which reads the specific chemical mechanism file and converts it into Fortran code to provide the input for the calculation of linear and non-linear parts. For the TROP_MOZART mechanism, there are 737 non-zero matrix entries of the system matrix and they are computed explicitly and exactly. The sparsity pattern reveals a classic “arrow matrix” ordering with the arrow pointing up as shown in Figure 2.1(a). If the LU factorization is performed directly on this matrix, almost all the entries in the upper-right matrix will be changed from an initial zero to a non-zero value during the execution. This is known as a fill-in issue [Lee, 2010] and it will significantly increase the cost of computation, which is determined by the total number of non-zero entries rather than by the size of the sparse matrix. Therefore, a permutation operator is applied before doing the LU factorization and the system matrix is flipped over to point down (see Figure 2.1(b)). The LU factorization starts with the

nearly diagonal part of the matrix and all fill-ins occur down the right hand side column where there are already a lot of non-zero entries. Note that although the system is different between the first-order implicit solver and the ROS-2 solver, the formation of the Jacobian matrix is exactly the same. Hence, the routines from the original first-order implicit solver can be used directly with only minor modification. It is also worth noting that in other solvers like SMVGEAR, the JSPARSE algorithm developed by Jacobson and Turco [1994] is used to reorder the Jacobian matrix based on the combined production and loss terms. In addition, the partial pivoting process is removed from the decomposition process to further reduce the number of matrix calculation. Instead, the chemical preprocessor of CAM4-Chem uses a diagonal Markowitz scheme [Lee, 2010] to reorder the Jacobian matrix, which searches on diagonal elements and chooses the pivot with the smallest Markowitz weight.

c. Form the LU factorization for the system matrix. As shown in Figure 2.1, the system matrix is very sparse with about 90% zeroes. In the current version of CAM4-Chem, the LU factorization is hardwired with a fixed pivoting order by knowing exactly the non-zeroes of system matrix. There are a total of 824 non-zero matrix entries after performing the LU factorization. Since the hardwired LU factorization does not need to test for pivots, the routine is highly optimized and efficient. It should be mentioned that extensive testing is done to ensure that the pivot strategy is robust even with the modification of the system in the ROS-2 formulas. We compared the accuracy of LU solution by capturing selected system matrices (examples used in Section 2.3.1) and analyzing them in MATLAB. The results showed that either full pivoting (including both row and column pivoting) or partial pivoting (only including row pivoting) had no appreciable impact on the accuracy of solution as compared with the fixed pivoting strategy.

d. Calculate the source term F on the right hand side.

e. Solve for the first-stage solution vector $\mathbf{k}_1$ with explicitly programmed steps of reduction.

f. Update source term F for the second stage with intermediate approximation $\mathbf{y}^n + \mathbf{h}\mathbf{k}_1$.

- g. Solve for the second-stage solution vector $\mathbf{k}_2$ using the same LU factorization result from the first stage.
- h. Update species mixing ratio vector from $\mathbf{y}^n$ to $\mathbf{y}^{n+1}$ for the next time step.

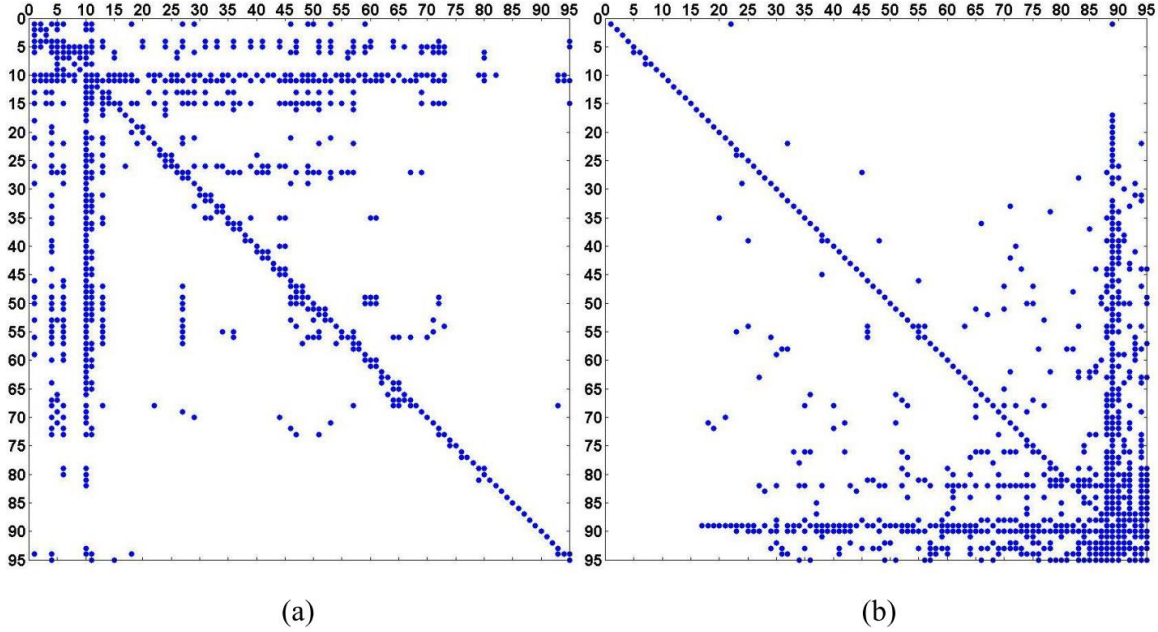


Figure 2.1. The sparsity pattern of left-hand side system matrix from the second-order Rosenbrock method (ROS-2) for the TROP_MOZART mechanism (a) before and (b) after permutation.

2.3 Results and Discussion

2.3.1 Numerical analysis of the first-order implicit solver and the ROS-2 solver

Before comparing the model simulation results, it is necessary to verify that both solvers converge to the same solution if the time step is small enough and they provide the numerical convergence rate as we expect. In order to verify this, a test problem is chosen for a linear system $\frac{du}{dt} = u$. Given the initial condition $u_0 = 1$ at $t_0 = 0$, it is trivial

to derive the analytical solution, $u(t) = e^t$. By arbitrarily setting $t_{\text{end}} = 2$, separate numerical simulations with different time step sizes are performed to integrate from t_0 to t_{end} and the numerical results for both solvers are summarized in Table 2.1. Note that for the ROS-2 solver, we only use the given fixed time step size for verification tests and no refinement of time step is involved. In Table 2.1, Δt is the time step size for a numerical simulation; Abs (Error1) is the absolute error between the exact solution and the numerical solution generated by the first-order implicit solver at time $= t_{\text{end}}$; Rate1 is the ratio of current absolute error over the previous absolute error with twice time step size for the first-order implicit solver; Abs (Error2) and Rate2 are the same but for the numerical solution generated by the ROS-2 solver. The decimal precision of numerical results is shown as 6 digits so that the absolute error less than $1.E-06$ is not included in the table. For the time step sizes finer than $1.25E-02$, the error between exact solution and the numerical solution generated by the first-order implicit solver is cut by half when the time step size is reduced by a factor of two, indicating a linear convergence of the first-order implicit solver and thus first-order numerical accuracy. On the other hand, a quadratic convergence (second-order accuracy) is observed for the ROS-2 solver as the error between exact solution and the numerical solution generated by the ROS-2 solver is cut by four when the time step size is reduced by a factor of two. Therefore, for a linear system, it proves that the convergence rate and accuracy of both solvers behave as we expect. Due to the quicker convergence of the ROS-2 solver, its absolute error with time step size equal to $3.91E-04$ is already around $3.10E-06$, while the absolute error for the first-order implicit solver with the same time step size is about $2.89E-03$. However, Table 2.1 reveals that both solvers will eventually converge to the same solution as long as the time step size is small enough.

For the non-linear system, the CAM4-Chem chemistry modules are first isolated such that other processes like dynamics and physics are excluded and the update of chemistry behaves exactly like a box model. Then input from some arbitrarily selected grid cells are used to provide the species concentrations, chemical reaction rates and independent forcings as initial condition. Lastly, a series of box-model simulations are

Table 2.1. Numerical Analysis of the First-Order Implicit Solver and Second-Order Rosenbrock (ROS-2) Solver for a Linear System.

Δt	Abs(Error1)	Rate1	Abs(Error2)	Rate2
2.00E-01	1.924170E+00		1.580427E+00	
1.00E-01	8.362072E-01	0.434581	2.806979E-01	0.177609
5.00E-02	3.923089E-01	0.469153	5.946621E-02	0.211851
2.50E-02	1.902726E-01	0.485007	1.372136E-02	0.230742
1.25E-02	9.372968E-02	0.492607	3.297921E-03	0.240349
6.25E-03	4.652076E-02	0.496329	8.085571E-04	0.245172
3.13E-03	2.317529E-02	0.498171	2.001870E-04	0.247585
1.56E-03	1.156648E-02	0.499087	4.980504E-05	0.248793
7.81E-04	5.777966E-03	0.499544	1.242119E-05	0.249396
3.91E-04	2.887666E-03	0.499772	3.101549E-06	0.249698
1.95E-04	1.443504E-03	0.499886	-	-
9.77E-05	7.216697E-04	0.499943	-	-
4.88E-05	3.608143E-04	0.499972	-	-
2.44E-05	1.804020E-04	0.499986	-	-
1.22E-05	9.019972E-05	0.499993	-	-
6.10E-06	4.509954E-05	0.499996	-	-
3.05E-06	2.254969E-05	0.499998	-	-
1.53E-06	1.127483E-05	0.499999	-	-
7.63E-07	5.637408E-06	0.500000	-	-
3.81E-07	2.818702E-06	0.500000	-	-
1.91E-07	1.409351E-06	0.500000	-	-

performed for one climate time step (1800 seconds by default) with different numerical time step sizes. We have tested input from different grid cells and they generally provide consistent results of numerical analysis. Hence one grid at Eastern US (latitude = 34.4°, longitude = -90.0°, level = 26 (bottom layer)) is used as an example for illustration (see Table 2.2(a)). Compared to Table 2.1, one additional column named “Ozone” is added to show the ozone concentration (units: mol/mol) from the box-model simulation after one climate time step. Note that the chemistry used in CAM4-Chem is a non-linear system and therefore no analytical solution exists. A common approach to obtain the “exact” solution in this case is to solve the system with a very tiny numerical time step size. We find out that for the ROS-2 solver, when the time step size is about 1.76 seconds, the ratio of absolute difference between ozone solution calculated by 1.76 seconds and ozone solution calculated by 3.52 seconds over the ozone solution calculated by 3.52 seconds is smaller than 1.E-12, which is beyond the decimal precision in CAM4-Chem. Hence 6.034342167442E-13 mol/mol is chosen as the “exact” solution here for numerical analysis. It turns out that the first-order implicit solver still converges linearly for this non-linear system while a quadratic convergence is again observed for the ROS-2 solver. Table 2.2 also shows that if the time step size is small enough, the first-order implicit solver will still converge to the “exact” solution with relative error (“Abs(Error2)” column divided by “Ozone” column) smaller than 1.E-12. However, the ROS-2 solver behaves better here since the ROS-2 solver using a numerical time step size equal to 1800 seconds produces a solution with relative error smaller than 1.E-7.

It is worth noting that the initial condition for the convergence test above is from the initial data of CAM4-Chem. The species concentration (e.g., ozone) is a little away from the realistic chemistry state of the atmosphere due to the spin-up effect. Therefore, we further examine another case with the initial condition from the output of CAM4-Chem after three-month simulation. It turns out that when the time step size is refined to 3.43E-3 second, the relative error of the ROS-2 solver is smaller than 1.E-12 and thus 5.973488706086E-08 mol/mol can be used as the “exact” solution for ozone (see Table

2.2(b)). For the ROS-2 solver, quadratic convergence is observed after the time step size is refined to 7.03 seconds and smaller, while the first-order implicit solver converges linearly at the similar scale. The first-order implicit solver at time step size equal to 1800 seconds surprisingly shows smaller relative error ($7.95\text{E-}4$) than that for the ROS-2 solver at the time step size equal to 1800 seconds ($2.52\text{E-}3$). However, both relative errors can be thought small enough when doing the analysis of ozone concentration in the realistic atmosphere. Overall, based on the box-model analysis here, it seems appropriate to use the ROS-2 solver with 1800 seconds time step for the real simulation of atmospheric chemistry.

2.3.2 First-order implicit solver and ROS-2 solver at 1800 seconds time step

We focus our analysis on annual and summertime mean surface ozone concentrations generated from a decadal simulation from 2001 to 2010. Using the 1800 seconds time step, the output from the ROS-2 solver (ROS-2_1800s) and the first-order implicit solver (ORI_1800s) are compared. Figure 2.2(a) shows a nationwide decrease of ten-year averaged annual mean surface ozone concentration between the ROS-2_1800s and the ORI_1800s (ROS-2_1800s minus ORI_1800s). The largest and smallest differences are -3.12 ppb and -0.58 ppb at the grid-cell level. By averaging the whole corresponding grids inside each state, the state-level differences are summarized in Table 2.3 and the Student's t-test suggests that the state-level differences of annual mean surface ozone concentration between the ROS-2_1800s and the ORI_1800s are statistically significant at $\alpha=0.05$ for 47 states except Washington. Considering the ten-year averaged surface ozone concentration during the summer season when the photolytic reaction is the most active during the year, Figure 2.2(b) clearly presents an even larger difference than the annual difference, especially over the Northeastern US. The largest and smallest differences at the grid-cell level are -5.62 ppb and -0.18 ppb, respectively. For the state level, the ROS-2_1800s again shows a lower prediction of summertime mean ozone concentration over the CONUS than the ORI_1800s, similar to that for the annual mean ozone concentration. According to Table 2.3, the Student's t-test suggests that the

Table 2.2. Numerical analysis of the first-order implicit solver and ROS-2 solver for a non-linear system with box-model simulations (units: mol/mol).

Δt	Ozone	Abs(Error)	Rate
(a) Input from initial data for CAM4-Chem			
First-order implicit solver			
1.80E+03	6.033875081040E-13	4.670864013899E-17	-
9.00E+02	6.034108571651E-13	2.335957908294E-17	0.500113
4.50E+02	6.034225356959E-13	1.168104830493E-17	0.500054
2.25E+02	6.034283759124E-13	5.840831817003E-18	0.500026
1.13E+02	6.034312962523E-13	2.920491926016E-18	0.500013
5.63E+01	6.034327564793E-13	1.460264834934E-18	0.500006
2.81E+01	6.034334866071E-13	7.301370989840E-19	0.500003
1.41E+01	6.034338516745E-13	3.650696949433E-19	0.500002
7.03E+00	6.034340342091E-13	1.825351129328E-19	0.500001
3.52E+00	6.034341254766E-13	9.126759599780E-20	0.500000
1.76E+00	6.034341711104E-13	4.563378492274E-20	0.500000
8.79E-01	6.034341939273E-13	2.281686398665E-20	0.499999
4.39E-01	6.034342053358E-13	1.140840099425E-20	0.499999
2.20E-01	6.034342110400E-13	5.704177929390E-21	0.499998
1.10E-01	6.034342138921E-13	2.852068971804E-21	0.499996
5.49E-02	6.034342153182E-13	1.425979000582E-21	0.499981
2.75E-02	6.034342160312E-13	7.129810184583E-22	0.499994
1.37E-02	6.034342163877E-13	3.564590052800E-22	0.499956
6.87E-03	6.034342165661E-13	1.780309368838E-22	0.499443
3.43E-03	6.034342166553E-13	8.885597969955E-23	0.499104
1.72E-03	6.034342166987E-13	4.546201610256E-23	0.511637
8.58E-04	6.034342167198E-13	2.442293167725E-23	0.537216

Table 2.2. Continued.

Δt	Ozone	Abs(Error)	Rate
4.29E-04	6.034342167270E-13	1.716399771048E-23	0.702782
2.15E-04	6.034342167344E-13	9.780966456978E-24	0.569854
1.07E-04	6.034342167385E-13	5.651929639532E-24	0.577850
5.36E-05	6.034342167414E-13	2.744983514679E-24	0.485672
2.68E-05	6.034342167436E-13	5.820152649848E-25	0.212029
ROS-2 solver			
1.80E+03	6.034341703907E-13	4.635350394929E-20	-
9.00E+02	6.034342027413E-13	1.400284692707E-20	0.302088
4.50E+02	6.034342129412E-13	3.802968924664E-21	0.271585
2.25E+02	6.034342157557E-13	9.885179905049E-22	0.259933
1.13E+02	6.034342164924E-13	2.518129837532E-22	0.254738
5.63E+01	6.034342166807E-13	6.349792599436E-23	0.252163
2.81E+01	6.034342167283E-13	1.590192123632E-23	0.250432
1.41E+01	6.034342167402E-13	3.937993638863E-24	0.247643
7.03E+00	6.034342167432E-13	9.399687893379E-25	0.238692
3.52E+00	6.034342167440E-13	1.899324624283E-25	0.202063
1.76E+00	6.034342167442E-13	-	-
(b) Input from CAM4-Chem output after three-month simulation			
First-order implicit solver			
1.80E+03	5.978245102732E-08	4.756396646170E-11	
9.00E+02	5.976140850864E-08	2.652144778010E-11	0.557595
4.50E+02	5.975076715046E-08	1.588008960000E-11	0.598764
2.25E+02	5.974356744260E-08	8.680381741498E-12	0.546620
1.13E+02	5.973906259478E-08	4.175533919096E-12	0.481031
5.63E+01	5.973668923211E-08	1.802171253901E-12	0.431603

Table 2.2. Continued.

Δt	Ozone	Abs(Error)	Rate
2.81E+01	5.973561178778E-08	7.247269202004E-13	0.402141
1.41E+01	5.973517535312E-08	2.882922579047E-13	0.397794
7.03E+00	5.973500657783E-08	1.195169678004E-13	0.414569
3.52E+00	5.973493966948E-08	5.260862329855E-14	0.440177
1.76E+00	5.973491143495E-08	2.437409320730E-14	0.463310
8.79E-01	5.973489874541E-08	1.168454599514E-14	0.479384
4.39E-01	5.973489277449E-08	5.713632806852E-15	0.488991
2.20E-01	5.973488988499E-08	2.824131301198E-15	0.494279
1.10E-01	5.973488846466E-08	1.403796901733E-15	0.497072
5.49E-02	5.973488776067E-08	6.998111962483E-16	0.498513
2.75E-02	5.973488741024E-08	3.493800075629E-16	0.499249
1.37E-02	5.973488723541E-08	1.745559012606E-16	0.499616
6.87E-03	5.973488714810E-08	8.723879520881E-17	0.499776
3.43E-03	5.973488710448E-08	4.361949686608E-17	0.500001
1.72E-03	5.973488708267E-08	2.180870287674E-17	0.499976
8.58E-04	5.973488707195E-08	1.109250524923E-17	0.508627
4.29E-04	5.973488706696E-08	6.096307879626E-18	0.549588
2.15E-04	5.973488706502E-08	4.160096441321E-18	0.682396
1.07E-04	5.973488706356E-08	2.696807320270E-18	0.648256
ROS-2 solver			
1.80E+03	5.958452613181E-08	1.503609290504E-10	-
9.00E+02	5.968294173170E-08	5.194532915820E-11	0.345471
4.50E+02	5.971188939336E-08	2.299766750360E-11	0.442728
2.25E+02	5.972455856641E-08	1.032849445230E-11	0.449111
1.13E+02	5.973019079380E-08	4.696267055103E-12	0.454690

Table 2.2. Continued.

Δt	Ozone	Abs(Error)	Rate
5.63E+01	5.973286765394E-08	2.019406917293E-12	0.430003
2.81E+01	5.973413490840E-08	7.521524604925E-13	0.372462
1.41E+01	5.973465434590E-08	2.327149588001E-13	0.309399
7.03E+00	5.973482583434E-08	6.122652349968E-14	0.263097
3.52E+00	5.973487236949E-08	1.469137010049E-14	0.239951
1.76E+00	5.973488354320E-08	3.517655699798E-15	0.239437
8.79E-01	5.973488616374E-08	8.971160996013E-16	0.255032
4.39E-01	5.973488682017E-08	2.406849016532E-16	0.268287
2.20E-01	5.973488699653E-08	6.432889656107E-17	0.267274
1.10E-01	5.973488704421E-08	1.664560031187E-17	0.258758
5.49E-02	5.973488705667E-08	4.190801385659E-18	0.251766
2.75E-02	5.973488705982E-08	1.043994577271E-18	0.249116
1.37E-02	5.973488706060E-08	2.619978784976E-19	0.250957
6.87E-03	5.973488706078E-08	7.629913970189E-20	0.291220
3.43E-03	5.973488706086E-08	-	-

differences are statistically significant over 42 out of 48 conterminous states at $\alpha=0.05$, ranging from -5.17 ppb to -0.75 ppb. We further evaluate both model results with the ground-level observation data obtained from the Air Quality System (AQS) archived by U.S. EPA (<https://www.epa.gov/aqs>, ~1200 observation sites over the CONUS). For the difference of ten-year averaged annual mean surface ozone concentration between model and monitor data, Figure 2.3(a-b) show that both ORI_1800s and ROS-2_1800s are likely to produce more than 20 ppb overestimate at the Eastern and Western US coastal areas. The ROS-2_1800s reduces the bias mainly over the Eastern US, consistent with the plot in Figure 2.2(a). The differences of ten-year averaged summertime mean surface ozone concentration between both solvers and AQS observation data are depicted in Figure 2.3(c-d). The bias is even higher and could be larger than 30 ppb, especially over the Central (e.g., West Virginia) and Northeast (e.g., Pennsylvania). The overestimate of ozone concentration in the Eastern US is a well-known issue from the previous literature about global chemistry-climate models [Murazaki and Hess, 2006; Reidmiller et al., 2009; Lapina et al., 2014], and potential reasons include coarse global resolution that fails to represent the steep topographic gradients in mixing depths [Fiore et al., 2009] and disproportionate sensitivity of models at the high ozone concentration level [Hollaway et al., 2012]. The spatial distribution of surface ozone concentration for the ROS-2_1800s presents an evident reduction of ozone bias at Southeast (e.g., Georgia), Central (e.g., Illinois) and Northeast (e.g., Pennsylvania). In addition, recalling the results of Student's t-test, most of these reductions are statistically significant at $\alpha=0.05$. This reveals that the ROS-2_1800s is likely to significantly reduce the overestimate of surface ozone concentration to some extent, especially during the summer season.

2.3.3 First-order implicit solver with 180 and 1800 seconds time step

Since a bias of surface ozone concentration is observed between the ORI-1800s and AQS monitor data, it is worth investigating whether refining the time step size is able to reduce the bias for the first-order implicit solver. The time step size in this study is

refined to 180 seconds and the output from first-order implicit solver (ORI_180s) are compared with that from the ORI_1800s. Figure 2.4(a) shows a nationwide decrease of ten-year averaged annual mean surface ozone concentration between the ORI_180s and the ORI_1800s (ORI_180s minus ORI_1800s). The largest and smallest differences at the grid-cell level are -0.53 ppb and -0.05 ppb, respectively. The state-level differences are listed in Table 4 and the mean normalized gross error EMNGE is also calculated based on the Eq. (2.6):

$$\text{Mean normalized gross error } E_{\text{MNGE}} = \frac{1}{N} \sum_{i=1}^N \left(\frac{|M_i - O_i|}{O_i} \right) \times 100\% \quad (2.6)$$

where N is the number of observations from AQS by time and space in each state; O_i and M_i are the i-th values of observation and model in each state, respectively. Albeit the ORI_180s gives a widely lower estimate of surface ozone concentration and EMNGE, the Student's t-test shown in Table 2.4 suggests that the state-level differences of annual mean surface ozone concentration over the CONUS between the ORI_180s and the ORI_1800s are not statistically significant at $\alpha = 0.05$. Considering the ten-year averaged summertime mean surface ozone concentration, Figure 2.4(b) presents that both positive and negative differences between the ORI_180s and the ORI_1800s appear, with the highest positive value 0.26 ppb and negative value -0.54 ppb. However, the Student's t-test shown in Table 2.4 indicates that the difference of summertime mean surface ozone concentration between the ORI_180s and the ORI_1800s is again not statistically significant at $\alpha = 0.05$. As stated before, the summertime mean surface ozone concentration is strongly overestimated by the current version of CAM4-Chem. By keeping the same first-order implicit solver but using the 180 seconds rather than 1800 seconds time step, the prediction bias seems to be reduced nationwide except at 5 states (Alabama: 0.16 ppb, Mississippi: 0.12 ppb, Georgia: 0.13 ppb, South Carolina: 0.04 ppb and Louisiana: 0.16 ppb). Nevertheless, the benefit is less visible due to the small difference (difference of EMNGE < 1% for most states) and neither the positive nor negative difference is statistically significant. Therefore, we cannot conclude that a better prediction is obtained by simply refining the time step size. On the other hand, the ROS-2_1800s does provide statistically

significant reduction of both annual and summertime mean surface ozone concentration bias over most conterminous states, as described in the previous section. Furthermore, the magnitude of surface ozone concentration decrease is also much higher than that here. This reflects the fact that the improvement from the ROS-2_1800s may not follow its second order accuracy as we supposed, but instead from its suitability for stiff problems [Shampine, 1982; Verwer et al., 1999]. Based on the results of numerical analysis in Section 3.1, the first-order solver may converge to the “exact” solution at tiny time step size but the computation will be unaffordable in the real simulation of an earth system model. However, the ROS-2 solver is likely to converge to the “exact” solution with relatively large time step size and the ROS-2 solver with 1800 seconds time step has already provided a solution with small relative error. We think the ROS-2_1800s solver could handle stiff system better than the original first-order implicit solver and thus provide a better estimate of surface ozone concentration, especially for the summer season.

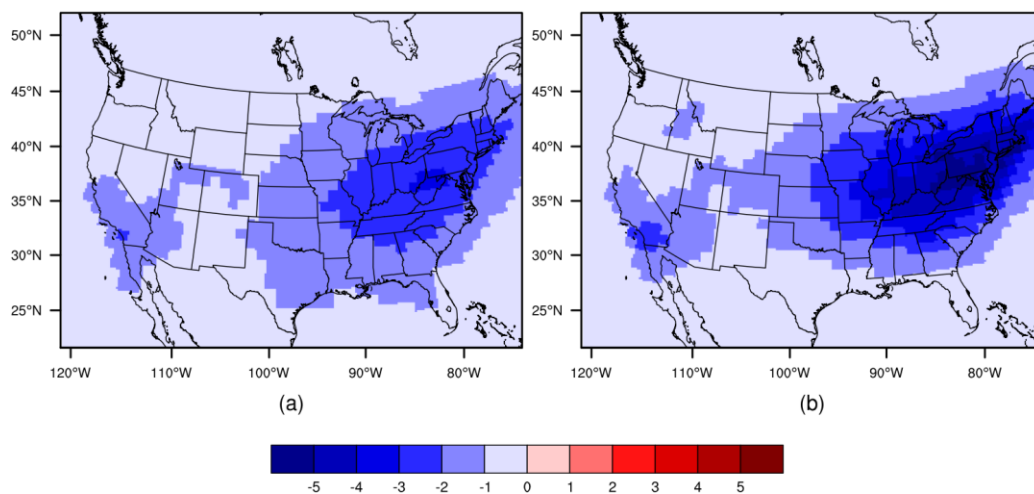


Figure 2.2. Difference of 10 year averaged (a) annual and (b) summertime (June, July, and August, JJA) mean surface ozone concentration (units: ppb) over the conterminous U.S. (CONUS) between the ROS-2 solver (ROS-2_1800s) and the first-order implicit solver (ORI_1800s) at 1800 s time step (ROS-2_1800s minus ORI_1800s).

Table 2.3. Student's t-test for bias (ROS-2_1800s minus ORI_1800s, units: ppb) between the ROS-2 solver and the first-order implicit solver at 1800 seconds time step in each state.

State	Annual mean bias	P-value	Summertime mean bias	P-value
AL	-1.74	< 0.001	-1.73	0.007
AZ	-1.10	< 0.001	-1.32	< 0.001
AR	-1.78	< 0.001	-2.02	< 0.001
CA	-1.20	< 0.001	-1.28	0.044
CO	-0.99	< 0.001	-1.09	< 0.001
CT	-2.66	0.001	-4.84	< 0.001
DE	-2.67	0.003	-4.63	< 0.001
FL	-1.01	0.021	-0.62	0.400
GA	-1.67	< 0.001	-1.79	0.035
ID	-0.76	0.008	-0.90	0.005
IL	-2.32	< 0.001	-3.63	< 0.001
IN	-2.60	< 0.001	-4.20	< 0.001
IA	-1.61	< 0.001	-2.52	< 0.001
KS	-1.27	< 0.001	-1.66	< 0.001
KY	-2.78	< 0.001	-4.21	< 0.001
LA	-1.24	0.002	-0.72	0.348
ME	-1.49	< 0.001	-2.40	0.001
MD	-2.67	0.030	-4.68	< 0.001
MA	-2.46	0.001	-4.54	< 0.001
MI	-1.79	< 0.001	-2.86	< 0.001
MN	-1.06	< 0.001	-1.41	< 0.001
MS	-1.49	< 0.001	-1.32	0.118
MO	-2.00	< 0.001	-2.85	< 0.001

Table 2.3. Continued.

State	Annual mean bias	P-value	Summertime mean bias	P-value
MT	-0.71	0.003	-0.80	< 0.001
NE	-1.10	< 0.001	-1.57	< 0.001
NV	-0.85	< 0.001	-0.90	0.001
NH	-2.38	< 0.001	-4.18	< 0.001
NJ	-2.51	0.002	-4.89	< 0.001
NM	-0.85	< 0.001	-0.75	0.022
NY	-2.25	< 0.001	-4.03	< 0.001
NC	-2.21	< 0.001	-3.13	< 0.001
ND	-0.72	0.008	-0.83	< 0.001
OH	-2.72	< 0.001	-4.63	< 0.001
OK	-1.47	< 0.001	-1.44	< 0.001
OR	-0.68	0.045	-0.67	0.405
PA	-2.85	< 0.001	-5.17	< 0.001
RI	-2.39	0.019	-4.68	0.018
SC	-1.83	< 0.001	-2.30	0.002
SD	-0.90	< 0.001	-1.22	< 0.001
TN	-2.40	< 0.001	-3.21	< 0.001
TX	-1.14	< 0.001	-0.67	0.058
UT	-1.00	< 0.001	-1.07	< 0.001
VT	-2.04	< 0.001	-3.51	< 0.001
VA	-2.78	< 0.001	-4.56	< 0.001
WA	-0.67	0.075	-0.48	0.531
WV	-3.01	< 0.001	-5.12	< 0.001
WI	-1.50	< 0.001	-2.23	< 0.001
WY	-0.87	< 0.001	-0.95	< 0.001

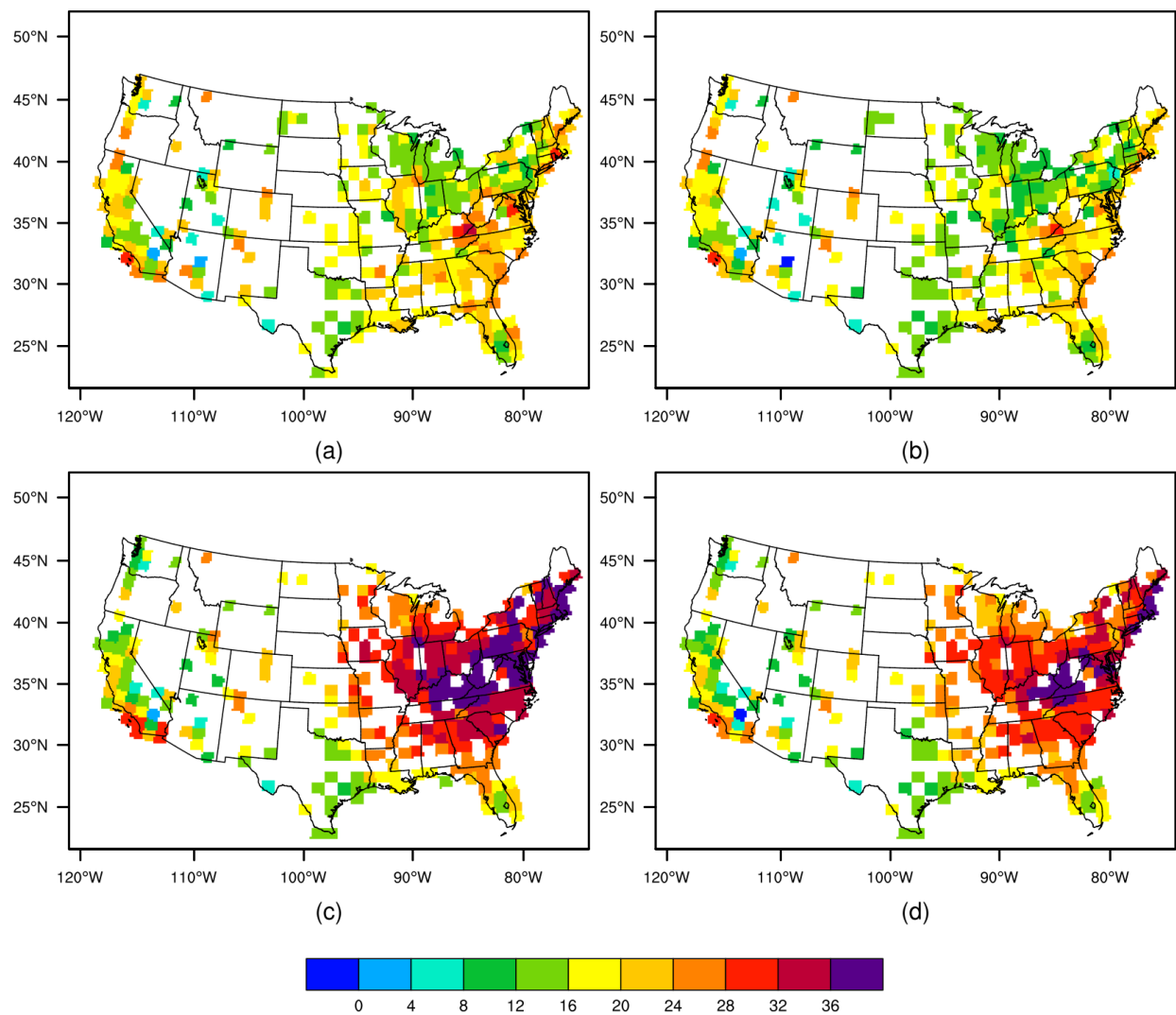


Figure 2.3. 10 year averaged (a) annual and (c) summertime mean surface ozone concentration bias (units: ppb) over the CONUS between the ORI_1800s and the AQS observation data (ORI_1800s minus AQS). (b, d) The same but for the ROS-2_1800s.

2.3.4 ROS-2 solver with 180 and 1800 seconds time step

As stated above, the ROS-2_1800s reduces the surface ozone concentration bias between simulation and observation data. A further question is raised: Can the ROS-2 solver benefit from refining the time step size? To answer this question, another time-slice simulation is conducted by using the ROS-2 solver with 180 seconds time step (ROS-2_180s). The difference of ten-year averaged annual mean surface ozone concentration between the ROS-2_180s and the ROS-2_1800s (ROS-2_180s minus ROS-2_1800s) is plotted in Figure 2.5(a). Surprisingly, the ROS-2_180s provides a nationwide higher estimate of surface ozone concentration, ranging from 0.16 ppb to 2.15 ppb. The primary difference locates in the Eastern US with the largest and smallest state-level differences in West Virginia and New Mexico, respectively (see Table 2.5). The Student's t-test suggests that the difference between the ROS-2_180s and the ROS-2_1800s is statistically significant over 32 states at $\alpha=0.05$, varying from 0.42 ppb to 2.07 ppb. The difference of ten-year averaged summertime mean surface ozone concentration between the ROS-2_180s and the ROS-2_1800s is shown in Figure 2.5(b). Instead of a consistently higher estimate of surface ozone concentration over the CONUS in Figure 2.5(a), some slight decreases are observed at parts of Texas and New Mexico. It presents a wider range of difference at grid-cell level from -0.19 ppb to 4 ppb. The state-level differences shown in Table 2.5 indicate that the ROS-2_180s still generates a higher estimate of summertime ozone concentration over most parts of the CONUS, similar to the results in Figure 2.5(a). However, the Student's t-test suggests that these differences are statistically significant over 28 states at $\alpha=0.05$, varying from 0.41 ppb to 3.71 ppb. As demonstrated in Figure 2.3, the ROS-2_1800s is still likely to overestimate the surface ozone concentration over the CONUS. Therefore, the ROS-2_180s seems to increase the surface ozone concentration bias and no benefit is obtained for the ROS-2 solver from refining the chemistry time step size in this case. However, compared to the ORI_180s and ORI_1800s (see Figure 2.2, Figure 2.4 and Figure 2.5), the ROS-2_180s still reduces the surface ozone concentration bias to some extent (up to 2 ppb for summer season). Therefore, the ROS-2_180s at least doesn't negatively impact the chemistry calculation compared with the original first-order implicit

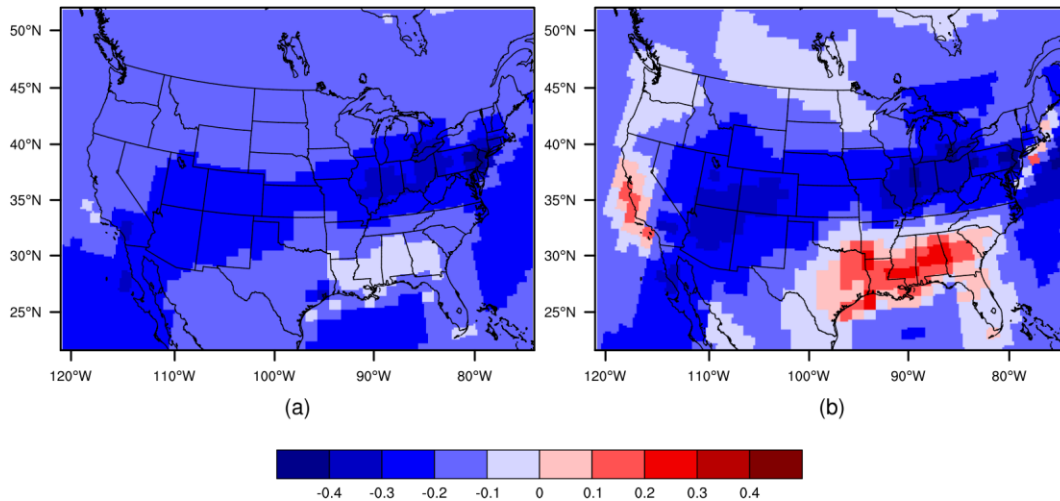


Figure 2.4. Difference of 10 year averaged (a) annual and (b) summertime mean surface ozone concentration (units: ppb) over the CONUS between the first-order implicit solver at 180 seconds time step (ORI_180s) and the ORI_1800s (ORI_180s minus ORI_1800s).

solver. From Section 2.3.1, both ROS-2_180s and ROS-2_1800s provide small relative errors and don't differ too much from each other (see Table 2.1 and Table 2.2). Thus we think some other factors may play a role here. We output the system matrices from the selected grid cells used in Section 2.3.1 and calculate the condition numbers of the matrices using the function `cond()` in MATLAB. All of them are larger than 10^{12} , which means the system matrices are strongly ill-conditioned and a minor change of left hand side system matrix or right hand side vector could lead to a major difference in the final solution. For the box model, the solution of the current time step will be used as the input for the next time step integration. Nevertheless, in the real simulation of CAM4-Chem, other processes like advection and diffusion will bring some additional mass from the neighboring grid cells or transfer some mass from the current grid cell, leading to a “discontinuity” of the solution between the end of the current time step and the beginning of the next time step. This “discontinuity” may be amplified by the ill-conditioned system and partially contributes to the difference between the ROS-2_180s and the ROS-2_1800s. It reminds us that the earth system model is very complex and

the improvement from the ROS-2 solver may not be limited to its numerical properties, but also related to its interaction with other processes (e.g., dynamics) and components (e.g., land and ocean). The exact reason can only be explored by additional case studies but that is beyond the scope of this study.

2.3.5 Evaluation of surface ozone concentration at global scale

Though this work mainly focuses on the bias of surface ozone concentration over the CONUS, model comparisons for variables at the global scale are necessary since CAM4-Chem is a global chemistry-climate model. The difference of ten-year averaged global surface ozone concentration between the ROS-2_1800s and the ORI_1800s is shown in Figure 2.6. The clear decrease of surface ozone concentration predicted by the ROS-2_1800s is observed for both annual mean (Figure 2.6(a)) and JJA mean (Figure 2.6(b)) results. In particular, greater than 3 ppb reduction occurs in Western Europe and Northeast China during the summer season. We further compare the model results to the observation data in Europe from the European Monitoring and Evaluation Program (EMEP, <http://www.emep.int/>) and Asia from the Acid Deposition Monitoring Network in East Asia (EANET, <http://www.eanet.asia/>). For Europe, the observed ten-year averaged annual mean ozone concentration is (mean $\pm$ standard deviation over space) 32.83 ± 8.12 ppb while those for the ORI_1800s and the ROS-2_1800s are 48.69 ± 7.83 ppb and 47.32 ± 7.76 ppb, respectively. Considering the ten-year averaged JJA mean ozone concentration, the observation value is 37.21 ± 10.08 ppb while those for the ORI_1800s and the ROS-2_1800s are 57.74 ± 12.27 ppb and 55.65 ± 11.76 ppb, respectively. For Asia, the observed ten-year averaged annual mean ozone concentration is 34.06 ± 11.51 ppb while the values from the ORI_1800s and the ROS-2_1800s are 44.91 ± 5.39 ppb and 43.99 ± 5.17 ppb, respectively. For the ten-year averaged JJA mean ozone concentration, the observed value is 27.91 ± 12.43 ppb while the values from the ORI_1800s and the ROS-2_1800s are 37.43 ± 11.01 ppb and 36.55 ± 10.57 ppb, respectively. Similar to the US, both solvers overestimate the surface ozone concentration over the European and Asian continents, especially for the

Table 2.4. Statistics for the first-order implicit solver at 180 seconds time step (ORI_180s) and 1800 seconds time step (ORI_1800s) in each state, respectively.

State	Annual mean bias	P-value	Summer mean bias	P-value	Mean Normalized Gross Error, %			
					ORI_1800s		ORI_180s	
					Annual	Summer	Annual	Summer
AL	-0.07	0.770	0.16	0.810	81.31	104.00	81.05	104.55
AZ	-0.24	0.246	-0.31	0.341	44.06	43.22	43.44	42.50
AR	-0.15	0.624	-0.03	0.942	64.43	81.66	63.95	81.63
CA	-0.18	0.549	-0.07	0.910	65.09	46.81	64.51	46.74
CO	-0.22	0.298	-0.30	0.053	57.51	49.84	56.89	49.13
CT	-0.33	0.673	-0.28	0.822	55.35	97.68	54.23	96.88
DE	-0.32	0.686	-0.29	0.769	88.26	104.58	87.03	103.77
FL	-0.14	0.758	0.00	0.999	67.31	86.78	66.81	86.76
GA	-0.09	0.770	0.13	0.881	78.54	102.13	78.22	102.61
ID	-0.15	0.591	-0.16	0.619	45.54	44.83	45.10	44.43
IL	-0.28	0.398	-0.33	0.259	70.47	96.98	69.44	95.99
IN	-0.33	0.397	-0.39	0.317	49.21	100.04	48.17	98.93
IA	-0.20	0.485	-0.24	0.477	67.42	98.33	66.70	97.55
KS	-0.21	0.402	-0.25	0.280	65.71	60.78	64.98	60.13
KY	-0.26	0.567	-0.23	0.615	76.22	120.02	75.34	119.31
LA	-0.09	0.826	0.16	0.843	76.44	84.65	76.10	85.23
ME	-0.18	0.637	-0.16	0.828	67.74	117.68	67.10	117.17
MD	-0.27	0.819	-0.24	0.838	88.58	116.22	87.62	115.56
MA	-0.27	0.698	-0.21	0.867	77.15	108.69	76.19	108.07
MI	-0.23	0.457	-0.26	0.667	51.34	86.75	50.58	85.92
MN	-0.14	0.564	-0.10	0.777	61.55	85.35	61.02	84.99
MS	-0.08	0.827	0.12	0.892	65.70	83.33	65.41	83.74

Table 2.4. Continued.

State	Annual mean bias	P- value	Summer mean bias	P- value	Mean Normalized Gross Error, %			
					ORI_1800s		ORI_180s	
					Annual	Summer	Annual	Summer
MO	-0.22	0.536	-0.25	0.488	59.57	84.60	58.85	83.98
MT	-0.15	0.533	-0.14	0.427	91.86	85.99	91.30	85.55
NE	-0.20	0.399	-0.25	0.154	38.78	42.86	38.27	42.32
NV	-0.18	0.406	-0.22	0.445	58.92	45.87	58.44	45.48
NH	-0.26	0.679	-0.20	0.851	74.45	126.73	73.48	126.05
NJ	-0.52	0.509	-0.52	0.660	46.58	84.76	44.66	83.25
NM	-0.21	0.372	-0.26	0.437	62.04	45.99	61.40	45.36
NY	-0.23	0.607	-0.21	0.713	65.25	109.30	64.47	108.64
NC	-0.18	0.529	-0.11	0.834	74.54	101.68	73.95	101.33
ND	-0.13	0.641	-0.09	0.684	52.50	57.42	52.07	57.10
OH	-0.31	0.281	-0.36	0.247	56.65	98.21	55.62	97.18
OK	-0.19	0.542	-0.11	0.778	50.69	48.86	50.11	48.63
OR	-0.13	0.703	-0.07	0.932	88.31	66.15	87.80	66.06
PA	-0.31	0.407	-0.30	0.462	66.68	108.20	65.58	107.30
RI	-0.23	0.811	-0.16	0.935	116.77	118.70	115.82	118.24
SC	-0.13	0.735	0.04	0.961	85.54	116.14	85.09	116.27
SD	-0.16	0.530	-0.17	0.437	43.84	41.09	43.34	40.59
TN	-0.18	0.552	-0.09	0.819	66.94	97.24	66.38	97.01
TX	-0.18	0.315	-0.05	0.889	52.15	51.04	51.51	51.16
UT	-0.22	0.358	-0.29	0.182	38.64	37.14	38.05	36.50
VT	-0.21	0.656	-0.15	0.855	43.96	78.53	43.32	78.10
VA	-0.24	0.490	-0.23	0.543	93.04	135.11	92.19	134.38
WA	-0.14	0.713	-0.04	0.954	96.26	63.51	95.53	63.29

Table 2.4. Continued.

State	Annual mean bias	P- value	Summer mean bias	P- value	Mean Normalized Gross Error, %			
					ORI_1800s		ORI_180s	
					Annual	Summer	Annual	Summer
WV	-0.27	0.565	-0.28	0.541	91.46	135.65	90.44	134.77
WI	-0.18	0.494	-0.16	0.675	50.81	82.15	50.25	81.67
WY	-0.18	0.444	-0.23	0.282	47.29	46.10	46.78	45.56

JJA season. However, the comparison between model simulation results and observation data suggests that the ROS-2_1800s is able to reduce the bias more at the European sites. One main reason is that most of the Asian observation stations are in Japan, Korea and Thailand, where less difference is seen from Figure 2.6. It is also worth noting that the observed JJA mean ozone concentration is lower than the annual mean value. This can be explained by the fact that more than half of the available ozone observation data comes from sites in Japan and the concentrations there are highest in spring but lowest in summer at all stations [Chatani and Sudo, 2011; Li et al., 2016]. This comparison reveals that the ozone concentration is globally decreased by the ROS-2_1800s instead of a random fluctuation. Therefore, the lower surface ozone concentration over the CONUS predicted by the ROS-2_1800s is not a trade-off against increasing the surface ozone concentration in other global regions.

2.3.6 Computational efficiency

Since the ROS-2 solver utilizes the same Jacobian matrix and LU factorization structure between the two stages, it is expected to speed up the computation compared to the original first-order implicit solver. Using the default simulation period setting (5 days), Table 2.6 summarizes the global statistics of average computational time per processor as a function of different numbers of threads. The percent of saved computational time and the factor of speedup are calculated by the following equations:

$$\text{Time per processor} = \frac{\text{Total time}}{\text{NTASKS_ATM} \times \text{NTHRDS_ATM}} \quad (2.7)$$

$$\text{Percent of saved time} = \frac{T_{\text{ORI}} - T_{\text{ROS-2}}}{T_{\text{ORI}}} \quad (2.8)$$

$$\text{Factor of speedup} = \frac{T_{\text{ORI}}}{T_{\text{ROS-2}}} \quad (2.9)$$

where “Total time” means the total summed wall-clock time consumed over all the processors. “NTASKS_ATM” and “NTHRDS_ATM” are variables that set the number of MPI tasks and the number of OpenMP threads per task, respectively. The product of these two variables specifies the total computational processors for the atmospheric component of CESM. TORI and TROS-2 refer to the computational time of the first-

Table 2.5. Statistics for the ROS-2 solver at 180 seconds time step (ROS-2_180s) and 1800 seconds time step (ROS-2_1800s) in each state, respectively.

State	Annual mean bias	P-value	Summer mean bias	P-value	Mean Normalized Gross Error, %			
					ROS-2_1800s		ROS-2_180s	
					Annual	Summer	Annual	Summer
AL	1.11	< 0.001	1.02	0.102	74.69	97.49	79.02	101.39
AZ	0.42	0.039	0.34	0.272	41.12	39.86	42.30	40.91
AR	1.06	< 0.001	1.02	0.001	58.72	75.94	62.16	78.99
CA	0.69	0.019	0.82	0.189	61.11	43.47	63.44	45.80
CO	0.39	0.064	0.14	0.377	54.71	47.23	55.83	47.59
CT	1.77	0.026	3.50	0.004	46.39	83.66	52.34	93.79
DE	1.74	0.035	3.24	0.003	78.10	91.86	84.72	100.76
FL	0.44	0.312	0.14	0.850	63.72	84.24	65.26	84.80
GA	1.06	< 0.001	1.12	0.182	72.31	95.82	76.32	99.83
ID	0.38	0.173	0.42	0.177	43.39	42.63	44.48	43.65
IL	1.36	< 0.001	1.94	< 0.001	62.14	86.09	67.02	91.96
IN	1.57	< 0.001	2.42	< 0.001	40.81	87.68	45.93	94.86
IA	0.90	0.001	1.23	< 0.001	61.69	90.25	64.88	94.19
KS	0.61	0.016	0.59	0.007	60.76	55.57	63.28	57.69
KY	1.84	< 0.001	2.71	< 0.001	66.61	107.21	72.98	115.42
LA	0.74	0.062	0.47	0.538	71.82	82.01	74.50	83.62
ME	1.00	0.007	1.68	0.018	61.72	107.22	65.89	114.97
MD	1.81	0.134	3.34	0.006	79.03	103.12	85.52	112.49
MA	1.69	0.016	3.39	0.005	68.28	95.03	74.37	105.24
MI	1.11	< 0.001	1.72	0.002	45.19	77.31	49.03	83.09
MN	0.57	0.018	0.67	0.044	57.37	79.85	59.70	82.64
MS	0.89	0.012	0.63	0.439	60.63	79.00	63.63	81.06

Table 2.5. Continued.

State	Annual mean bias	P-value	Summer mean bias	P-value	Mean Normalized Gross Error, %			
					ROS-2_1800s		ROS-2_180s	
					Annual	Summer	Annual	Summer
MO	1.15	0.001	1.36	< 0.001	53.11	76.74	56.82	80.61
MT	0.34	0.152	0.32	0.064	89.13	83.27	90.51	84.60
NE	0.50	0.032	0.52	0.002	36.21	40.13	37.29	40.74
NV	0.39	0.060	0.26	0.354	56.16	43.42	57.54	44.41
NH	1.66	0.011	3.14	0.002	65.65	112.08	71.78	123.12
NJ	1.32	0.085	3.02	0.012	37.26	70.68	42.18	79.40
NM	0.28	0.237	-0.05	0.881	59.49	44.28	60.28	44.02
NY	1.58	0.001	3.00	< 0.001	57.52	96.48	62.96	106.03
NC	1.46	< 0.001	2.01	< 0.001	67.22	92.39	72.10	98.39
ND	0.33	0.213	0.29	0.182	50.20	55.01	51.22	55.76
OH	1.77	< 0.001	3.02	< 0.001	47.72	85.08	53.52	93.62
OK	0.80	0.008	0.65	0.084	45.85	44.93	48.57	46.82
OR	0.37	0.279	0.39	0.622	85.48	63.72	87.04	65.35
PA	1.93	< 0.001	3.71	< 0.001	56.79	93.03	63.48	103.91
RI	1.65	0.089	3.53	0.062	106.95	105.15	113.74	115.37
SC	1.18	0.001	1.46	0.042	79.02	108.52	83.21	113.33
SD	0.41	0.099	0.41	0.046	41.42	38.59	42.39	39.09
TN	1.59	< 0.001	1.95	< 0.001	59.23	87.88	64.38	93.73
TX	0.57	0.002	0.24	0.484	47.54	48.86	50.07	50.08
UT	0.40	0.098	0.14	0.522	35.92	34.69	37.01	35.04
VT	1.44	0.002	2.65	0.001	37.76	68.61	42.13	76.09
VA	1.93	< 0.001	3.19	< 0.001	83.14	120.85	90.00	130.78
WA	0.35	0.352	0.28	0.712	92.88	61.39	94.63	62.61

Table 2.5. Continued.

State	Annual mean bias	P-value	Summer mean bias	P-value	Mean Normalized Gross Error, %			
					ROS-2_1800s		ROS-2_180s	
					Annual	Summer	Annual	Summer
WV	2.07	< 0.001	3.58	< 0.001	80.47	119.75	88.00	130.81
WI	0.91	< 0.001	1.28	< 0.001	46.08	75.47	48.94	79.29
WY	0.38	0.114	0.19	0.373	44.83	43.83	45.89	44.27

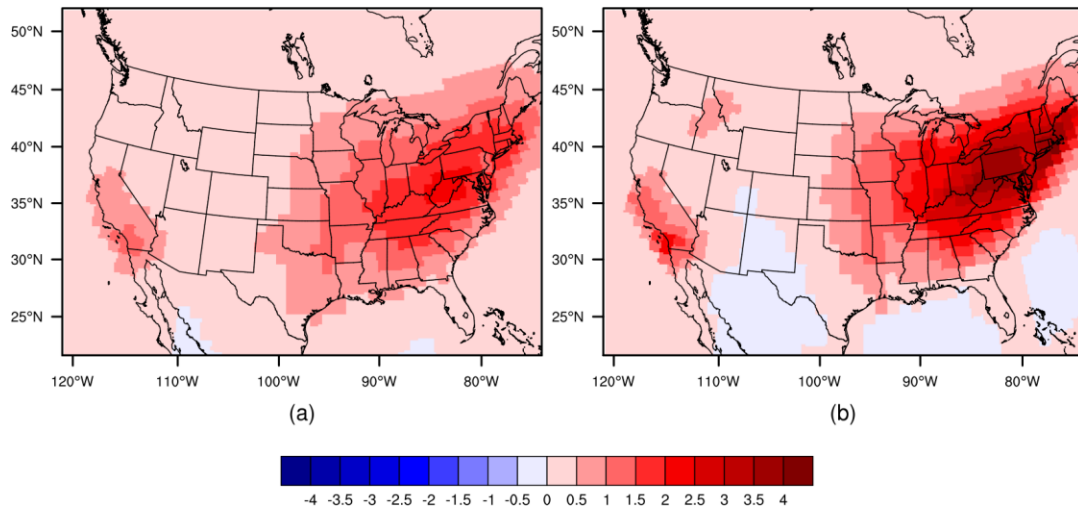


Figure 2.5. Difference of 10 year averaged (a) annual and (b) summertime mean surface ozone concentration (units: ppb) over the CONUS between the ROS-2 solver at 180 seconds time step (ROS-2_180s) and the ROS-2_1800s (ROS-2_180s minus ROS-2_1800s).

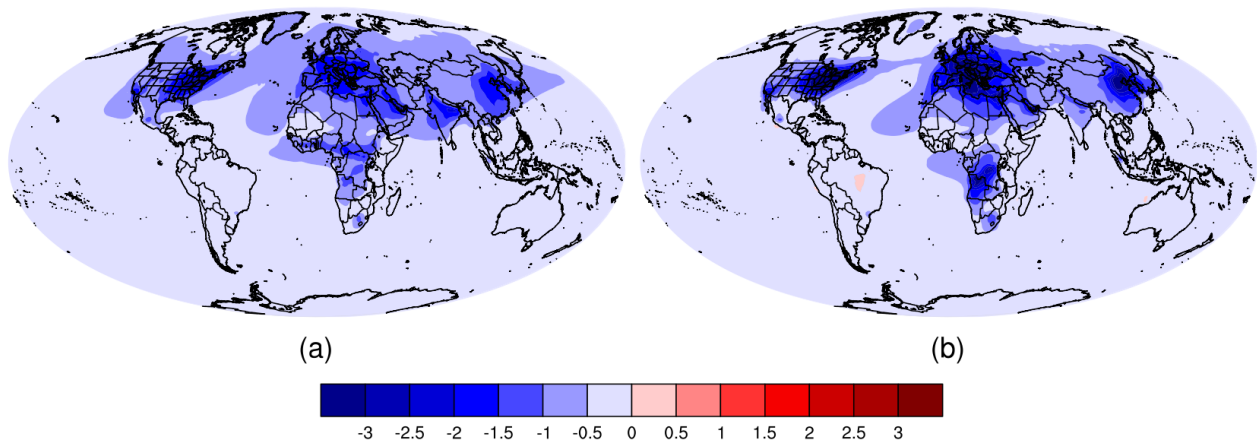


Figure 2.6. Difference of 10 year averaged (a) annual and (b) JJA mean surface ozone concentration (units: ppb) between the ROS-2_1800s and the ORI_1800s (ROS-2_1800s minus ORI_1800s) over the global continents.

order implicit solver and the ROS-2 solver with given number of processors, respectively. The Titan supercomputer at Oak Ridge National Laboratory (ORNL) is used for performance testing. Titan is a hybrid-architecture Cray XK7 system with 18,688 compute nodes (16 processors per node) and a theoretical peak performance exceeding 27 petaflops. The chemistry update takes the longest time per processor when only 48 processors are requested, but it drops rapidly as the number of processors increases (see Table 6). The average speedup of computational time for the ROS-2_1800s over that for ORI_1800s is around 1.95 when the total requested processors are less than or equal to 768, but reduces slightly to 1.83 when more processors are used. Similarly, the average speedup of computational time for the ROS-2_180s over that for the ORI_180s is about 1.88 when the total requested processors are less than or equal to 768, but reduces slightly to 1.77 with more processors. For the first-order implicit solver, using one tenth of the time step size doesn't simply result to a slower performance by a factor of ten. This is probably due to the fact that with a finer time step, fewer iteration steps are required for the chemistry solver to converge. For the ROS-2 solver, it is similarly found that the ROS-2_180s is not ten times slower than then ROS-2_1800s. This is mainly caused by the adaptive time step method implemented in the ROS-2_1800s, which introduces additional computation at refined time step level and impedes the performance. Generally, the computational time per processor for the ROS-2_1800s is about 48% less than that for ORI_1800s and the computational time per processor for the ROS-2_180s is about 46% less than that for the ORI_180s. The ORI_1800s and ROS-2_1800s are further used to conduct one-month and one-year simulations separately to examine whether more improvements are possible (see Table 2.6). By requesting 1,536 processors, the ROS-2_1800s takes about 47% less computational time than ORI_1800s for both one-month and one-year simulations, indicating that the performance improvement of the ROS-2_1800s over the ORI_1800s is stable at around 47%, regardless of the simulation period. It is worth noting that we use the same routines to form the Jacobian matrix and conduct the LU factorization for both solvers, which makes the comparison of performance fair. Therefore, the faster computational speed of the ROS-2 solver should

be attributed to the advantage of numerical algorithm itself that avoids the re-evaluation of Jacobian matrix and LU factorization between two stages.

2.4 Conclusion

To the authors' knowledge, this is the first study of implementing the ROS-2 solver in CAM4-Chem and evaluating its impact on the prediction of surface ozone concentration over the CONUS. Compared to the ORI_1800s, the ROS-2_1800s is likely to provide consistently nationwide lower ten-year averaged annual mean surface ozone concentration (see Figure 2.2(a)) and the Student's t-test suggests that the difference is statistically significant over 47 states at $\alpha = 0.05$. For the summertime, the ROS-2_1800s similarly presents a significantly lower estimate of ten-year averaged surface ozone concentration over the CONUS. The absolute difference is much larger (up to 5.2 ppb for one state) and 42 out of 48 states are statistically significant at $\alpha = 0.05$. This implies that the ROS-2_1800s can help improve the performance of surface ozone concentration over the CONUS, especially during the summer season when the photochemical reactions are the most active. The lower global surface ozone concentration generated by the ROS-2_1800s also suggests that its prediction of lower surface ozone concentration over the CONUS is not a random fluctuation and the model evaluation shows that the ROS-2_1800s can reduce the bias in Europe and Asia to some extent as well.

In addition, the time step size for the original first-order implicit solver is refined to 180 seconds (one tenth of the default time step) to examine whether it could help improve the estimate of surface ozone concentration over the CONUS. Figure 2.4 shows that compared to ORI_1800s, ORI-180s does slightly reduce the surface ozone concentration but the improvement is less visible (only around 0.5 ppb as maximum) and not statistically significant, either. This reveals that just refining the time step size by a factor of 10 for the original first-order implicit solver is not able to provide statistically significant improvement of surface ozone concentration prediction. Thus it is necessary to use other chemical

Table 2.6. Summary of computational time (units: seconds) per processor for each solver (ORI_180s, ORI_1800s, ROS-2_180s and ROS-2_1800s) with various simulation periods.

Simulation period	Number of Processors	Computational time per processor for each solver			
		ORI_1800s	ROS-2_1800s	Percent of Saved Time	Speedup
5 days	48	714	370	48.18 %	1.93
	96	359	185	48.47 %	1.94
	192	180	93	48.33 %	1.94
	384	91	47	48.35 %	1.94
	768	46	23	50.00 %	2.00
	1536	22	12	45.45 %	1.83
	3072	11	6	45.45 %	1.83
		ORI_180s	ROS-2_180s	Percent of Saved Time	Speedup
5 days	48	5,133	2,724	46.93 %	1.88
	96	2,568	1,363	46.92 %	1.88
	192	1,287	681	47.09 %	1.89
	384	645	340	47.29 %	1.90
	768	310	167	46.13 %	1.86
	1536	145	82	43.45 %	1.77
	3072	72	41	43.06 %	1.76
		ORI_1800s	ROS-2_1800s	Percent of Saved Time	Speedup
1 month	1536	139	73	47.48 %	1.90
		ORI_1800s	ROS-2_1800s	Percent of Saved Time	Speedup
1 year	1536	1607	844	47.48%	1.90

solvers like the ROS-2 solver in this study. However, the outperformance for the ROS-2 solver may not come from its second order accuracy with respect to time scale, but from its suitability for solving stiff problems. We have also applied the 180 seconds time step to the ROS-2 solver but the ROS-2_180s surprisingly performs worse than the ROS-2_1800s for estimating the surface ozone concentration. By studying some selected grid cells, the system matrix is found to be strongly ill-conditioned and other processes (e.g., diffusion) will cause “discontinuity” for the solution between the end of the current time step and the beginning of the next time step. The ROS-2 solvers with different time step sizes could converge to different solutions after a long-term simulation. In the real application, the CAM4-Chem may accumulate errors from other processes like dynamics and physics. Thus the time step size for the ROS-2 solver in chemistry should be chosen carefully in order to provide a compensation of error.

The computational efficiencies of the ORI_1800s, ORI_180s, ROS-2_1800s and ROS-2_180s are also analyzed. Even if the first-order implicit solver has been optimized to solve the chemical reaction system efficiently, the ROS-2 solver takes about 47 % less computational time than the original first-order implicit solver when using the same time step size. Our analysis indicates this speedup is explained by the fact that the ROS-2 solver utilizes the same Jacobian matrix and LU factorization structure during the two-stage calculations, which could consume 90% of the total computational time for the chemistry update. It is also observed that the ROS-2_180s is not ten times slower than the ROS-2_1800s because of the adaptive time step method implemented in the ROS-2_1800s.

In future work, case studies with different configurations may be explored to examine whether the benefit of the ROS-2 solver observed in this study still persists. Other solvers from the Rosenbrock family (e.g., ROS-3) can also be implemented to see whether further bias reduction is possible for the surface ozone concentration prediction over the CONUS. Since all the constituents except water vapor are treated as radiatively inactive by default, the reduction of ozone concentration from the ROS-2_1800s is not expected to influence

the longwave and shortwave cloud forcings in the current simulation but it is worth investigating the differences in a fully interactive simulation in future work. The chemistry update behaves like a box model in CAM4-Chem, which is an ideal target for parallel implementations. Either NVIDIA CUDA or OpenACC could be used as a programming model on advanced high performance computing platforms to harness the power of the GPU and further improve the computational performance.

References

- Blom, J. G. and J. G. Verwer (2000), A comparison of integration methods for atmospheric transport-chemistry problems, *Journal of Computational and Applied Mathematics*, **126**: 381-396, doi:10.1016/S0377-0427(99)00366-0.
- Chatani, S. and K. Sudo (2011), Influences of the variation in inflow to East Asia on surface ozone over Japan during 1996–2005, *Atmos. Chem. Phys.*, **11**, 8745-8758, doi:10.5194/acp-11-8745-2011.
- Cooper, O. R., D. D. Parrish, A. Stohl, M. Trainer, P. Nédélec, V. Thouret, J. P. Cammas, S. J. Oltmans, B. J. Johnson, D. Tarasick, T. Leblanc, I. S. McDermid, D. Jaffe, R. Gao, J. Stith, T. Ryerson, K. Aikin, T. Campos, A. Weinheimer and M. A. Avery (2010), Increasing springtime ozone mixing ratios in the free troposphere over western North America, *Nature*, **463**, 344-348, doi:10.1038/nature08708.
- Daescu, D., G. R. Carmichael and A. Sandu (2000), Adjoint implementation of rosenbrock methods applied to variational data assimilation problems, *Journal of Computational Physics*, **165**, 496-510, doi:10.1006/jcph.2000.6622.
- Emmons, L. K., S. Walters, P. G. Hess, J.-F. Lamarque, G. G. Pfister, D. Fillmore, C. Granier, A. Guenther, D. Kinnison, T. Laepple, J. Orlando, X. Tie, G. Tyndall, C. Wiedinmyer, S. L. Baughcum and S. Kloster (2010), Description and evaluation of the Model for Ozone and Related chemical Tracers, version 4 (MOZART-4), *Geosci. Model Dev.*, **3**, 43-67, doi:10.5194/gmd-3-43-2010.
- Fiore, A. M., F. J. Dentener, O. Wild, C. Cuvelier, M. G. Schultz, P. Hess, C. Textor, M. Schulz, R. M. Doherty, L. W. Horowitz, I. A. MacKenzie, M. G. Sanderson, D. T. Shindell, D. S. Stevenson, S. Szopa, R. Van Dingenen, G. Zeng, C. Atherton, D. Bergmann, I. Bey, G. Carmichael, W. J. Collins, B. N. Duncan, G. Faluvegi, G. Folberth, M. Gauss, S. Gong, D. Hauglustaine, T. Holloway, I. S. A. Isaksen, D. J. Jacob, J. E. Jonson, J. W. Kaminski, T. J. Keating, A. Lupu, E. Marmer, V. Montanaro, R. J., Park, G. Pitari, K. J. Pringle, J. A. Pyle, S. Schroeder, M. G. Vivanco, P. Wind, G. Wojcik, S. Wu and A. Zuber (2009), Multimodel estimates of intercontinental source-receptor

relationships for ozone pollution, *Journal of Geophysical Research*, **114**, D04301, doi: 10.1029/2008JD010816.

Fiore, A. M., J. T. Oberman, M. Y. Lin, L. Zhang, O. E. Clifton, D. J. Jacob, V. Naik, L. W. Horowitz, J. P. Pinto and G. P. Milly (2014), Estimating North American background ozone in U.S. surface air with two independent global models: Variability, uncertainties, and recommendations, *Atmos. Environ.*, **96**, 284-300, doi: 10.1016/j.atmosenv.2014.07.045.

Fiscus, E. L., F. L. Booker and K. O. Burkey (2005), Crop responses to ozone: uptake, modes of action, carbon assimilation and partitioning, *Plant, Cell and Environment*, **28**, 997–1011, doi: 10.1111/j.1365-3040.2005.01349.x.

Hollaway, M. J., S. R. Arnold, A. J. Challinor and L. D. Emberson (2012), Intercontinental trans-boundary contributions to ozone-induced crop yield losses in the Northern Hemisphere, *Biogeosciences*, **9**(1), 271-292, doi: 10.5194/bg-9-271-2012.

Jacobson, M. Z. and R. P. Turco (1994), SMVGEAR: A sparse-matrix, vectorized gear code for atmospheric models, *Atmospheric Environment*, **28**(2), 273-284, doi: 10.1016/1352-2310(94)90102-3.

Jacobson M. Z. and D. L. Ginnebaugh (2010), Global-through-urban nested three-dimensional simulation of airpollution with a 13,600-reaction photochemical mechanism, *Journal of Geophysical Research*, **115**, D14304, doi:10.1029/2009JD013289.

Jacobson, M. Z., M. A. Delucchi, M. A. Cameron and B. A. Frew (2015), Low-cost solution to the grid reliability problem with 100% penetration of intermittent wind, water, and solar for all purposes, *Proceedings of the National Academy of Sciences of the United States of America*, **112**(49), 15060-15065, doi: 10.1073/pnas.1510028112.

Karnosky, D. F., J. M. Skelly, K. E. Percy and A. H. Chappelka (2007), Perspectives regarding 50 years of research on effects of tropospheric ozone air pollution on US forests, *Environmental Pollution*, **147**(3), 489-506, doi:10.1016/j.envpol.2006.08.043.

Sensitivity of chemical tracers to meteorological parameters in the MOZART-3 chemical transport model

Kinnison, D. E., G. P. Brasseur, S. Walters, R. R. Garcia, D. R. Marsh, F. Sassi, V. L. Harvey, C. E. Randall, L. Emmons, J.-F. Lamarque, P. Hess, J. J. Orlando, X. X. Tie,

W. Randel, L. L. Pan, A. Gettelman, C. Granier, T. Diehl, U. Niemeier and A. J. Simmons (2007), Sensitivity of chemical tracers to meteorological parameters in the MOZART-3 chemical transport model, *J. Geophys. Res.*, **112**, D20302, doi: 10.1029/2006JD007879.

Lamarque, J.-F., L. K. Emmons, P. G. Hess, D. E. Kinnison, S. Tilmes, F. Vitt, C. L. Heald, E. A. Holland, P. H. Lauritzen, J. Neu, J. J. Orlando, P. J. Rasch and G. K. Tyndall (2012), CAM-chem: Description and evaluation of interactive atmospheric chemistry in the Community Earth System Model, *Geosci. Model Dev.*, **5**, 369-411, doi: 10.5194/gmd-5-369-2012.

Lapina, K., D. K. Henze, J. B. Milford, M. Huang, M. Y. Lin, A. M. Fiore, G. Carmichael, G. G. Pfister and K. Bowman (2014), Assessment of source contributions to seasonal vegetative exposure to ozone in the U.S., *Journal of Geophysical Research Atmos.*, **119**, 324-340, doi:10.1002/2013JD020905.

Lee, Ingyu (2010), A robust ILU preconditioner using constraints diagonal Markowitz, *ACMSE'10 Proceedings of the 48th Annual Southeast Regional Conference, Oxford, Mississippi*, **15**, 1-6, doi: 10.1145/1900008.1900031.

Li, J., W. Y. Yang, Z. F. Wang, H. S. Chen, B. Hu, J. J. Li, Y. L. Sun, P. Q. Fu and Y. Q. Zhang (2016), Modeling study of surface ozone source-receptor relationships in East Asia, *Atmospheric Research*, **167**, 77–88, doi:10.1016/j.atmosres.2015.07.010.

Linford, J. C., J. Michalakes, M. Vachharajani and A. Sandu (2009), Multi-core acceleration of chemical kinetics for simulation and prediction, *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, SC '09*, Association for Computing Machinery, New York, NY, **7**, 1-11, doi: 10.1145/1654059.1654067.

Long, M. S., W. C. Keene, R. Easter, R. Sander, A. Kerkweg, D. Erickson, X. Liu and S. Ghan (2013), Implementation of the chemistry module MECCA (v2.5) in the modal aerosol version of the Community Atmosphere Model component (v3.6.33) of the Community Earth System Model, *Geosci. Model Dev.*, **6**, 255-262, doi:10.5194/gmd-6-255-2013.

Murazaki, K. and P. Hess (2006), How does climate change contribute to surface ozone change over the United States? *J. Geophys. Res.*, **111**, D05301, doi:10.1029/2005JD005873.

Neale, R. B., J. H. Richter, S. Park, A. J. Conley, P. H. Lauritzen, A. Gettelman, D. L. Williamson, P. J. Rasch, S. J. Vavrus, M. A. Taylor, W. D. Collins, M. H. Zhang and S. J. Lin (2010), Description of the NCAR Community Atmosphere Model (CAM 4.0), *NCAR/TN-485+STR*, pp13.

Neale, R. B., J. Richter, S. Park, P. H. Lauritzen, S. J. Vavrus, P. J. Rasch and M. H. Zhang (2013), The Mean Climate of the Community Atmosphere Model (CAM4) in Forced SST and Fully Coupled Experiments, *Journal of Climate*, **26**, 5150-5168, doi: 10.1175/JCLI-D-12-00236.1.

Reidmiller, D. R., A. M. Fiore, D. A. Jaffe, D. Bergmann, C. Cuvelier, F. J. Dentener, B. N. Duncan, G. Folberth, M. Gauss, S. Gong, P. Hess, J. E. Jonson, T. Keating, A. Lupu, E. Marmer, R. Park, M. G. Schultz, D. T. Shindell, S. Szopa, M. G. Vivanco, O. Wild and A. Zuber (2009), The influence of foreign vs. North American emissions on surface ozone in the US, *Atmos. Chem. Phys.*, **9**, 5027-5042, doi:10.5194/acp-9-5027-2009.

Rotman, D. A., J. R. Tannahill, D. E. Kinnison, P. S. Connell, D. Bergmann, D. Proctor, J. M. Rodriguez, S. J. Lin, R. B. Rood, M. J. Prather, P. J. Rasch, D. B. Considine, R. Ramarosan and S. R. Kawa (2001), Global Modeling Initiative assessment model: Model description, integration, and testing of the transport shell, *J. Geophys. Res.*, **106**(D2), 1669-1691, doi: 10.1029/2000JD900463.

Rotman, D. A., C. S. Atherton, D. J. Bergmann, P. J. Cameron-Smith, C. C. Chuang, P. S. Connell, J. E. Dignon, A. Franz, K. E. Grant, D. E. Kinnison, C. R. Molenkamp, D. D. Proctor and J. R. Tannahill (2004), IMPACT, the LLNL 3-D global atmospheric chemical transport model for the combined troposphere and stratosphere: Model description and analysis of ozone and other trace gases, *J. Geophys. Res.*, **109**, D04303, doi: 10.1029/2002JD003155.

Sandu, A., J. G. Verwer, M. Van Loon, G. R. Carmichael, F. A. Potra, D. Dabdub and J. H. Seinfeld (1997), Benchmarking stiff ode solvers for atmospheric chemistry problems-

I. Implicit vs explicit, *Atmospheric Environment*, **31**(19), 3151-3166, doi:10.1016/S1352-2310(97)00059-9.

Sandu, A. and R. Sander (2006), Technical note: Simulating chemical systems in Fortran90 and Matlab with the Kinetic PreProcessor KPP-2.1, *Atmos. Chem. Phys.*, **6**(1), 187-195, doi:10.5194/acp-6-187-2006.

Sarwar, G., J. Godowitch, B. H. Henderson, K. Fahey, G. Pouliot, W. T. Hutzell, R. Mathur, D. Kang, W. S. Goliff and W. R. Stockwell (2013), A comparison of atmospheric composition using the Carbon Bond and Regional Atmospheric Chemistry Mechanisms, *Atmos. Chem. Phys.*, **13**, 9695-9712, doi:10.5194/acp-13-9695-2013.

Shampine, L. F. (1982), Implementation of Rosenbrock Methods, *Transactions on Mathematical Software (TOMS)*, Association for Computing Machinery, New York, NY, **8**(2), 93-113, doi: 10.1145/355993.355994.

Stevenson, D. S., C. E. Johnson, W. J. Collins, R. G. Derwent, K. P. Shine and J. M. Edwards (1998), Evolution of tropospheric ozone radiative forcing, *Geophysical Research Letters*, **25**(20), 3819-3822, doi: 10.1029/1998GL900037.

Stevenson, D. S., F. J. Dentener, M. G. Schultz, K. Ellingsen, T. P. C. van Noije, O. Wild, G. Zeng, M. Amann, C. S. Atherton, N. Bell, D. J. Bergmann, I. Bey, T. Butler, J. Cofala, W. J. Collins, R. G. Derwent, R. M. Doherty, J. Drevet, H. J. Eskes, A. M. Fiore, M. Gauss, D. A. Hauglustaine, L. W. Horowitz, I. S. A. Isaksen, M. C. Krol, J.-F. Lamarque, M. G. Lawrence, V. Montanaro, J.-F. Müller, G. Pitari, M. J. Prather, J. A. Pyle, S. Rast, J. M. Rodriguez, M. G. Sanderson, N. H. Savage, D. T. Shindell, S. E. Strahan, K. Sudo and S. Szopa (2006), Multimodel ensemble simulations of present-day and near-future tropospheric ozone, *J. Geophys. Res.*, **111**, D08301, doi: 10.1029/2005JD006338.

Stevenson, D. S., P. J. Young, V. Naik, J.-F. Lamarque, D. T. Shindell, A. Voulgarakis, R. B. Skeie, S. B. Dalsoren, G. Myhre, T. K. Berntsen, G. A. Folberth, S. T. Rumbold, W. J. Collins, I. A. MacKenzie, R. M. Doherty, G. Zeng, T. P. C. van Noije, A. Strunk, D. Bergmann, P. Cameron-Smith, D. A. Plummer, S. A. Strode, L. Horowitz, Y. H. Lee, S. Szopa, K. Sudo, T. Nagashima, B. Josse, I. Cionni, M. Righi, V. Eyring, A. Conley, K. W. Bowman, O. Wild and A. Archibald (2013), Tropospheric ozone changes, radiative

forcing and attribution to emissions in the Atmospheric Chemistry and Climate Model Intercomparison Project (ACCMIP), *Atmos. Chem. Phys.*, **13**, 3063-3085, doi:10.5194/acp-13-3063-2013.

Sun, J., J. S. Fu, K. Huang and Y. Gao (2015), Estimation of future PM_{2.5}- and ozone-related mortality over the continental United States in a changing climate: An application of high-resolution dynamical downscaling technique, *Journal of the Air & Waste Management Association*, **65**(5), 611-623, doi: 10.1080/10962247.2015.1033068.

Val Martin, M., C. L. Heald and S. R. Arnold (2014), Coupling dry deposition to vegetation phenology in the Community Earth System Model: Implications for the simulation of surface O₃, *Geophys. Res. Lett.*, **41**(8), 2988-2996, doi: 10.1002/2014GL059651.

Val Martin, M., C. L. Heald, J.-F. Lamarque, S. Tilmes, L. K. Emmons and B. A. Schichtel (2015), How emissions, climate, and land use change will impact mid-century air quality over the United States: a focus on effects at national parks, *Atmos. Chem. Phys.*, **15**, 2805-2823, doi:10.5194/acp-15-2805-2015.

Verwer, J. G., E. J. Spee, J. G. Blom and W. Hundsdorfer (1999), A second-order rosenbrock method applied to photochemical dispersion problems, *SIAM J. Sci. Comput.*, **20**(4), 1456–1480, doi:10.1137/S1064827597326651.

Young, P. J., A. T. Archibald, K. W. Bowman, J.-F. Lamarque, V. Naik, D. S. Stevenson, S. Tilmes, A. Voulgarakis, O. Wild, D. Bergmann, P. Cameron-Smith, I. Cionni, W. J. Collins, S. B. Dalsøren, R. M. Doherty, V. Eyring, G. Faluvegi, L. W. Horowitz, B. Josse, Y. H. Lee, I. A. MacKenzie, T. Nagashima, D. A. Plummer, M. Righi, S. T. Rumbold, R. B. Skeie, D. T. Shindell, S. A. Strode, K. Sudo, S. Szopa and G. Zeng (2013), Pre-industrial to end 21st century projections of tropospheric ozone from the Atmospheric Chemistry and Climate Model Intercomparison Project (ACCMIP), *Atmos. Chem. Phys.*, **13**, 2063-2090, doi:10.5194/acp-13-2063-2013.

Zeng, G., J. A. Pyle and P. J. Young (2008), Impact of climate change on tropospheric ozone and its global budgets, *Atmos. Chem. Phys.*, **8**, 369-387, doi:10.5194/acp-8-369-2008.

Zhang, H., J. C. Linford, A. Sandu and R. Sander (2011), Chemical Mechanism Solvers in Air Quality Models, *Atmosphere*, **2**(3), 510-532, doi:10.3390/atmos2030510.

CHAPTER III COMPUTATIONAL BENEFIT OF GPU OPTIMIZATION FOR THE ATMOSPHERIC CHEMISTRY MODELING

A version of this chapter was submitted to the journal of advances in modeling earth systems and it was under revision at this moment.

Abstract

Global chemistry-climate models are computationally burdened as the chemical mechanisms become more complex and realistic. Optimization for a GPU may make longer global simulation with regional detail possible, but limited study has been done to explore the potential benefit for the atmospheric chemistry modeling. Hence in this study, the second-order Rosenbrock solver of the chemistry module of CAM4-Chem is ported to the GPU to gauge potential speed-up. We find that on the CPU, the fastest performance is achieved using the Intel compiler with a block interleaved memory layout. Different combinations of compiler and memory layout lead to $\sim 6.15\times$ difference in the computational time. In contrast, the GPU version performs best using a fully interleaved memory layout with block size equal to a multiple of the warp size, CUDA streams for independent kernels and constant memory. Moreover, the most efficient data movement between CPU and GPU is gained by allocating the memory contiguously during the data initialization on the GPU. The speed-up of the GPU version reaches a factor of $\sim 11.7\times$ for the computation alone and $\sim 3.82\times$ when the data movement between CPU and GPU is considered. The computational time of the GPU version increases more slowly than the CPU version as a function of the number of loop iterations.

3.1 Introduction

Physics and chemistry are closely coupled in the framework of earth system modeling. Most radiatively active compounds (e.g., CH₄, O₃, and aerosols) in Earth's current atmosphere are also chemically active. The atmospheric chemistry is an essential component of climate [Tian and Chipperfield, 2006; IPCC AR5 Chapter 8, 2013; Collins et al., 2017], which includes the homogeneous (e.g., gas-phase species) and heterogeneous (e.g., gas and aerosol) reactions, aerosol and acid (nitrogen + sulfur) deposition, and cloud-aerosol interactions in the atmosphere. A robust representation of atmospheric chemistry including the chemical reaction with other species (e.g., gaseous species, aerosols and water) and photolysis (interaction with solar radiation) is thus crucial to determine the burden and lifetime of chemically active compounds [Su et al., 2011; Dameris and Jöckel, 2013; Lamarque et al., 2013; Tilmes et al., 2016].

Due to the complex chemical mechanism and strong variability of temporal and spatial patterns, atmospheric chemistry is usually characterized by the existence of significant non-linearity [Kleinman et al., 2001] and a wide range of scales [Isaksen et al., 2009], which makes it challenging to model. High resolution global models capture some of the same scales as mesoscale weather and regional air quality models so an argument can be made that global models should exercise more comprehensive chemistry. This prompts us to review and improve all the computational methods of global chemical simulation. Currently, a first-order implicit solver is widely used in the global chemistry-climate models [Austin et al., 2003; Horowitz et al., 2003; Schraner et al., 2008; Emmons et al., 2010]. The first-order implicit solver is unconditional stable [Kinnison et al., 2007], but may suffer from low computational efficiency and low accuracy. Sun et al. [2017] implemented a second-order Rosenbrock (ROS-2) solver in the global chemistry-climate model (CAM4-Chem), replacing the original first-order implicit solver. The results showed that utilizing the same optimized subroutine structure, the ROS-2 solver achieved ~2× speed-up on the CPU over the original first-order implicit solver. This

speed-up results from avoiding a re-evaluation of the Jacobian matrix and LU factorization during the two-stage computation.

In addition to improving the numerical method of the solver, new computer architectures demand a review of the optimization strategy. The heterogeneous architecture of supercomputers has developed rapidly now including multi-node parallelism and graphics processing units (GPUs). Optimizing for the GPU, considerable speed-up was achieved for both regional [Michalakes and Vachharajani, 2008; Linford et al., 2009] and global models [Korwar et al., 2013; Xu et al., 2015]. The GPU was highly efficient solving large and dense matrix systems, but the atmospheric chemistry problem was characterized by small (size less than 100×100) and sparse (10% non-zero elements) matrix systems. However, it was still possible to utilize the GPU efficiently when a large number of small matrices were solved simultaneously and independently [Dong et al., 2014]. Alvanos and Christoudias [2017] recently used the GPU accelerators to speed up the chemistry module of the global chemistry-climate model ECHAM/MESSy Atmospheric Chemistry (EMAC) by a factor of 1.75 $\times$. However, the CUDA codes in their study were parsed from the Fortran codes generated by the Kinetic PreProcessor (KPP). It was incompatible with the global chemistry-climate models such as CAM4-Chem that did not use KPP. In addition, little information was provided to explore the optimal configurations of the CUDA kernels. Hence in this work, we will port the ROS-2 chemical solver of CAM4-Chem [Sun et al., 2017] to the GPU to solve the chemistry as a box model and examine a series of optimization strategies. The goal is to investigate whether the chemistry box model can benefit from the GPU and its associated most optimized configuration. This investigation will inform the software engineering choices that developers must make to effectively optimize global chemistry-climate models for high resolution and comprehensive atmospheric chemical mechanisms.

3.2 Methodology

3.2.1 Chemistry configuration in CAM4-Chem

In CAM4-Chem, the default finite volume dynamic core uses a tensor product (latitude × longitude × vertical-level grid) over the global sphere [Lin, 2004; Mirin and Worley, 2011]. In order to achieve parallelization, domain decomposition is involved, which divides the global domain into different subdomains. Each subdomain contains 26 chunks (default for the 1° × 1° horizontal resolution). Inside each chunk there are 16 columns and each column consists of 26 vertical layers (default for the 1° × 1° horizontal resolution). Each subdomain is assigned as an MPI task, and OpenMP directives are used for thread-level parallelism when looping over chunks in each subdomain [Worley and Drake, 2005]. Therefore, each OpenMP thread handles exactly the chemistry computation inside one chunk. Further use of either MPI or OpenMP for the chemistry computation inside a chunk is unlikely to bring more computational benefit. For the chemistry at each grid point inside a chunk, a system of ordinary differential equation (ODE) is solved at each time step, which takes the following form:

$$\frac{dy}{dt} = P(y) - L(y) + I(y) \quad (3.1)$$

where y is the vector of volume mixing ratios for the chemical species at a given grid cell; the right hand side source terms include the production $P(y)$ and loss $L(y)$ due to chemical reactions and the external forcing (i.e., lightning and aircraft emissions). In the real implementation, the dimension of the species concentration array is declared as (ncol, pver, gas_pcnst), and thus the loop structure to solve the ODE above looks like:

```
do lev = 1, pver
```

```
  do i = 1, ncol
```

```
    Chemistry update [Sun et al., 2017]
```

1. Local data initialization
2. Formation of the Jacobian matrix
3. LU factorization

4. Formation of source terms b_1 on the right hand side
5. Solve the $Ak_1 = b_1$ system to obtain the intermediate solution k_1
6. Update of source terms b_2 on the right hand side with k_1
7. Solve the $Ak_2 = b_2$ system to obtain the intermediate solution k_2
8. Update the solution array x for the new time step
9. Local data free

end do

end do

where $ncol$ is the number of columns allocated to a given chunk; $pver$ is the number of vertical layers allocated to a given column; gas_pcnst is the total number of species; A is the Jacobian matrix required by the ROS-2 method [Verwer et al., 1999] to solve the non-linear equations; b_1 and b_2 are the right hand side source terms during the two-state calculations; k_1 and k_2 are the intermediate solutions. Since the chemistry in CAM4-Chem is treated independently among different columns and vertical layers, it behaves exactly like a box model for a given column and vertical layer. Therefore, it is possible to vectorize the two loops on the CPU and further compute the chemistry on the GPU simultaneously. To simplify the codes, the chemistry box model in the rest of this study will merge the two loops inside one chunk as one loop:

do loop = 1, chunk size

Chemistry update [Sun et al., 2017]

end do

where chunk size is number of loop iterations and is examined for a wide range as shown in the sections below.

3.2.2 Architecture

The Titan supercomputer at Oak Ridge National Laboratory is used for the computational performance and analysis. Each Titan compute node contains one AMD Opteron™ 6274 (Interlagos) CPU (16 cores) and one NVIDIA Tesla™ K20X (Kepler) GPU connected through a PCI express 2.0 interface. GPU computation is organized

into thread blocks, where each thread block has one or more warps of 32 threads each, and all the instructions (e.g., addition) are issued at the warp level. This execution model is called single instruction multiple thread (SIMT). A function launched on the GPU is called a kernel. Each Kepler K20X GPU contains 14 streaming multiprocessors (SMX) that can run up to 2,048 threads, 16 thread blocks, or 64 warps. Note that there are several other important limits imposed on the Kepler K20X GPU such as a maximum of 255 registers per thread or 65,536 per SMX, 48 kilobytes shared memory per SMX, and 64 kilobytes constant memory per GPU. Violating any of these limits will lead to kernel launch failure.

3.2.3 Memory layout

The memory layout is known to play a critical role in achieving good computational performance [Dongarra et al., 2017]. The strided and interleaved memory layouts are generally competitive for problems of very small sizes that are available in the fast GPU-accelerated implementation of the standard basic linear algebra subroutines (cuBLAS) and other research [Gates, et al., 2017]. Therefore, their effects on our problems will be investigated. To illustrate the difference between them, consider the following four matrices:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}, D = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}.$$

In Fortran, arrays are stored column-wise, so the strided memory layout (SML) stores the matrices consecutively as below:

$$[a_{11}, a_{21}, a_{12}, a_{22}, b_{11}, b_{21}, b_{12}, b_{22}, c_{11}, c_{21}, c_{12}, c_{22}, d_{11}, d_{21}, d_{12}, d_{22}],$$

while the fully interleaved memory layout (FIML) stores the i -th entries of all matrices consecutively, and looks like:

$$[a_{11}, b_{11}, c_{11}, d_{11}, a_{21}, b_{21}, c_{21}, d_{21}, a_{12}, b_{12}, c_{12}, d_{12}, a_{22}, b_{22}, c_{22}, d_{22}],$$

By storing each matrix contiguously, it is clear that the SML can access two elements in the same matrix quickly. In contrast, the elements in the same matrix are not stored consecutively in the FIML. For example, if the number of matrices is N , moving from the

first element to the second element in the same matrix requires a jump of N memory locations, which could hinder opportunities to reuse cached data, and thus reduce the overall performance when N is large. Despite this drawback, the FIML is expected to benefit from vectorization --- an implicit single instruction multiple data (SIMD) parallelization for a single core processor, where the code is transformed into SIMD vector operations (e.g., addition) that can be executed in parallel as single instructions. On a modern multi-core computational architecture, achieving a high-level vectorization is important for obtaining excellent performance. The SML may not utilize vectorization as effortlessly as the FIML. For instance, consider the following four calculations:

$$a_{11} = a_{12} + a_{21}$$

$$b_{11} = b_{12} + b_{21}$$

$$c_{11} = c_{12} + c_{21}$$

$$d_{11} = d_{12} + d_{21}$$

For the SML, it can only use one Advanced Vector Extensions (AVX) register and thus will take four clock cycles to complete the four addition instructions. However, for the FIML, it only takes one clock cycle instead since it can use four AVX registers due to its data storage. There is an intermediate approach between these two memory layouts, called the block interleaved memory layout (BIML). In the BIML, instead of interleaving all N matrices, the first K matrices are interleaved, then the next K matrices, and so on. For example, take K = 2 and BIML will store the four matrices above as:

$$[a_{11}, b_{11}, a_{21}, b_{21}, a_{12}, b_{12}, a_{22}, b_{22}, c_{11}, d_{11}, c_{21}, d_{21}, c_{12}, d_{12}, c_{22}, d_{22}].$$

The main benefit of the “block interleaved” approach is that when moving from the first element to the second element in the same matrix, it requires a jump of only K memory locations instead of N memory locations for the FIML. Meanwhile, the BIML can readily utilize vector instructions, unlike the SML. K is a tuning parameter and it is chosen as 4 here because the AMD Opteron™ 6274 CPU supports the 4-wide Fused Multiply-Add (FMA) vector instructions. In this study, we will explore the performance of these three memory layouts and see which one fits best for the atmospheric chemistry modeling.

3.3 Results and Discussion

3.3.1 Basic analysis

Before proceeding to investigate the computational performance, it is necessary to understand the computational rate and memory requirements of the chemistry box model. The main functions of the chemistry box model include formation of the Jacobian matrix, LU factorization and solve, and formation of the right hand side source term. The analysis of floating point operations (FLOP) for one loop iteration indicates that formation of the Jacobian matrix and LU (both factorization and solve) should consume roughly 70% of the total computational time (Table 3.1). Note that the calculation of FLOP for LU solve and formation of the right hand side source term has been multiplied by a factor of two, due to the two-stage computation in the ROS-2 method (similarly for the calculation of data copy later). On the other hand, formation of the Jacobian matrix requires a significantly higher amount of data copy (both copy in and copy out) than other functions. This is due to the fact that formation of the Jacobian matrix consists of two sub-functions, which calculate the linear and non-linear components of the Jacobian matrix separately. However, in the real architecture, caching will significantly reduce the cost of accessing data from memory. In order to take the cache effect into account, the Performance Application Programming Interface (PAPI v5.5) is used to measure the L2 cache misses. This count is further multiplied by the cache line size (64 bytes for the processor on Titan) and divided by the time to estimate the memory bandwidth (unit: GB/s) in this study. Besides the L2 cache misses, the computational rate (unit: GFLOP per second (GFLOPS = $1\text{E}+9$ FLOPS)) is also measured by PAPI for 100, 1,000 and 10,000 loop iterations. The results generally vary among different number of loop iterations, compilers and memory layouts (Figure 3.1). The highest bandwidth and computational rate for the whole chemistry box model per loop are 1.58 GB/s (GNU compiler with FIML) and 1.35 GFLOPS (Intel compiler with BIML), respectively. In particular, for the SML (Figure 3.1a, 3.1d and 3.1g), the major bandwidth use comes from the data initialization (maximum 1.6 GB/s) while it is almost negligible for the other

functions. This reflects the fact that most of the input arrays required for functions like formation of the Jacobian matrix and LU (both factorization and solve) are already loaded into the cache and reside there after doing the data initialization. For the FIML (Figure 3.1b, 3.1e and 3.1h), the LU solve function is observed with the highest bandwidth for both GNU (6.31 GB/s) and PGI (2.23 GB/s) compilers. However, the Others function that updates the intermediate solutions from the two-stage computation yields the highest bandwidth for the Intel compiler (3.39 GB/s). For the BIML (Figure 3.1c, 3.1f and 3.1i), the bandwidth is generally smaller than 0.1 GB/s for the individual functions with GNU and PGI, but still reaches up to 1.17 GB/s with Intel. On the other hand, formation of the right hand side source term yields a higher computational rate in most configurations (Figure 3.2), but the highest computational rate is achieved by the LU solve with the Intel compiler and BIML (1.95 GFLOPS). Considering the AMD Opteron™ 6274 processor on Titan, the theoretical peak computational rate of one core is $2.2 \text{ (GHz)} \times 8 \text{ (double precision FLOP per cycle for FMA4)} = 17.6 \text{ GFLOPS}$. The practical peak memory bandwidth is measured by the STREAM benchmark program provided by the University of Virginia (<https://www.cs.virginia.edu/stream/>) and the results show that when running with one thread, the memory bandwidth on Titan varies from 5.8 GB/s to 13.46 GB/s for different operations, such as copy and scale, and compilers. Compared with these hardware limitations, the whole chemistry box model reaches only up to 7.67% of the theoretical peak computational rate (Intel compiler with BIML) and 25.61% of the practical peak memory bandwidth (GNU compiler with FIML). However, these percentages increase to 11.08% (LU solve, Intel compiler and BIML) and 102.35% (LU solve, GNU compiler with FIML) for the individual functions, which has clearly exceeded the practical peak memory bandwidth. It is worth noting that in modern computer processors, there is also a technique named “cache prefetching” that fetches data into cache before it is needed. This will further increase the bandwidth as measured by L2 cache misses for the functions above and exceed the practical peak memory bandwidth more for the functions like LU solve. Therefore, the analysis here reveals that the chemistry box model performance is not limited by the computational rate but by the memory bandwidth. Since the GPU has both more floating point cores

and higher memory bandwidth than the CPU [Mantell et al., 2016; Alvanos and Christoudias, 2017], we believe that it is still promising to gain some computational benefit from the GPU.

Table 3.1. The floating point operations (FLOP) and data copy (unit: KB) per loop iteration for the main functions of the chemistry box model.

Function	FLOP	Copy in	Copy out
Formation of the Jacobian matrix	3,076	25.64	13.18
LU factorization	4,075	6.59	6.59
LU solve	3,106	14.7	1.52
Formation of right hand side source term	4,568	6.69	3.04

3.3.2 CPU

For the chemistry box model of chemistry on the CPU, the two most dominant factors that may affect the computational performance are the compilers and memory layouts. On Titan, three major compilers (GNU: gcc/4.9.3, Intel: intel/16.0.3.210 and PGI: pgi/16.10.0) are examined in this study with the flags that enable AVX vectorization (“-O3 -fopenmp -mavx” for GNU, “-O3 -qopenmp -mavx” for Intel and “-O3 -openmp -Mvect=simd:256” for PGI). The Intel compiler also provides memory alignment directives to further assist vectorization. The results show that the fastest computational time for a given number of loop iterations is achieved by the Intel compiler with BIML (Figure 3.3). Using SML and FIML require $\sim 1.97\times$ and $\sim 6.11\times$ the BIML computational time for the Intel compiler (Figure 3.3b). However, for the GNU compiler (Figure 3.3a), using SML yields the fastest computational time, and using FIML and BIML require $\sim 2.09\times$ and $\sim 3.19\times$ the SML computational time. The PGI compiler (Figure 3.3c) behaves similarly to the GNU compiler, but the ratios reduce to $\sim 2.03\times$ and $\sim 1.16\times$,

Figure 3.1. Memory bandwidth (unit: GB/s) for (a-c) GNU, (d-f) Intel and (g-i) PGI compilers using different memory layouts (left panel: strided memory layout; middle panel: fully interleaved memory layout; right panel: block interleaved memory layout). Different colors refer to the different functions in the chemistry box model (All: the whole chemistry box model; Init: data initialization; Jacob: formation of the Jacobian matrix; LF: LU factorization; LS: LU solve; Others: update of intermediate and final solutions; RHS: formation of the right hand side source term).

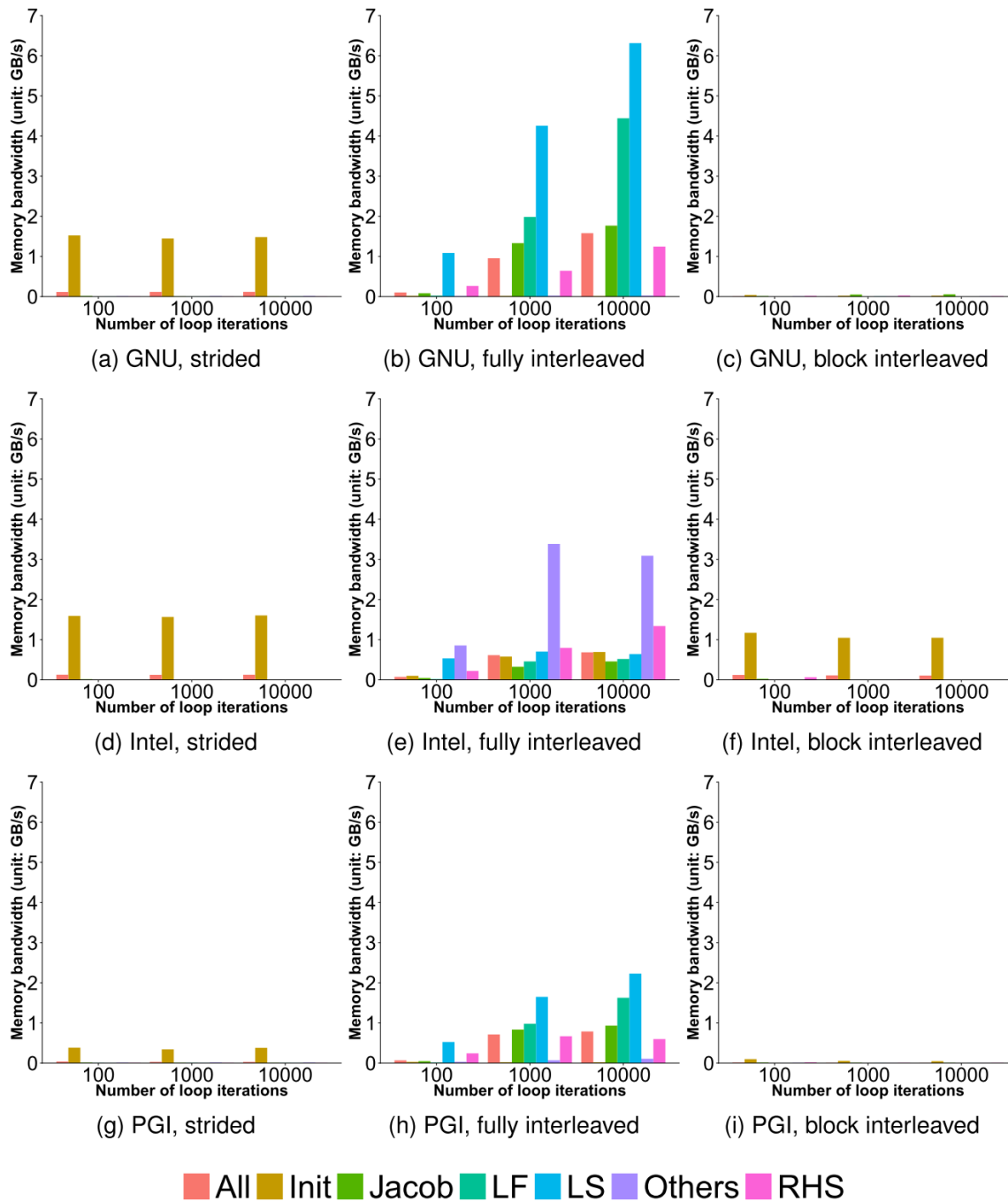


Figure 3.1. Continued.

Figure 3.2. The same as Figure 3.1 but for the computational rate (unit: GFLOPS).

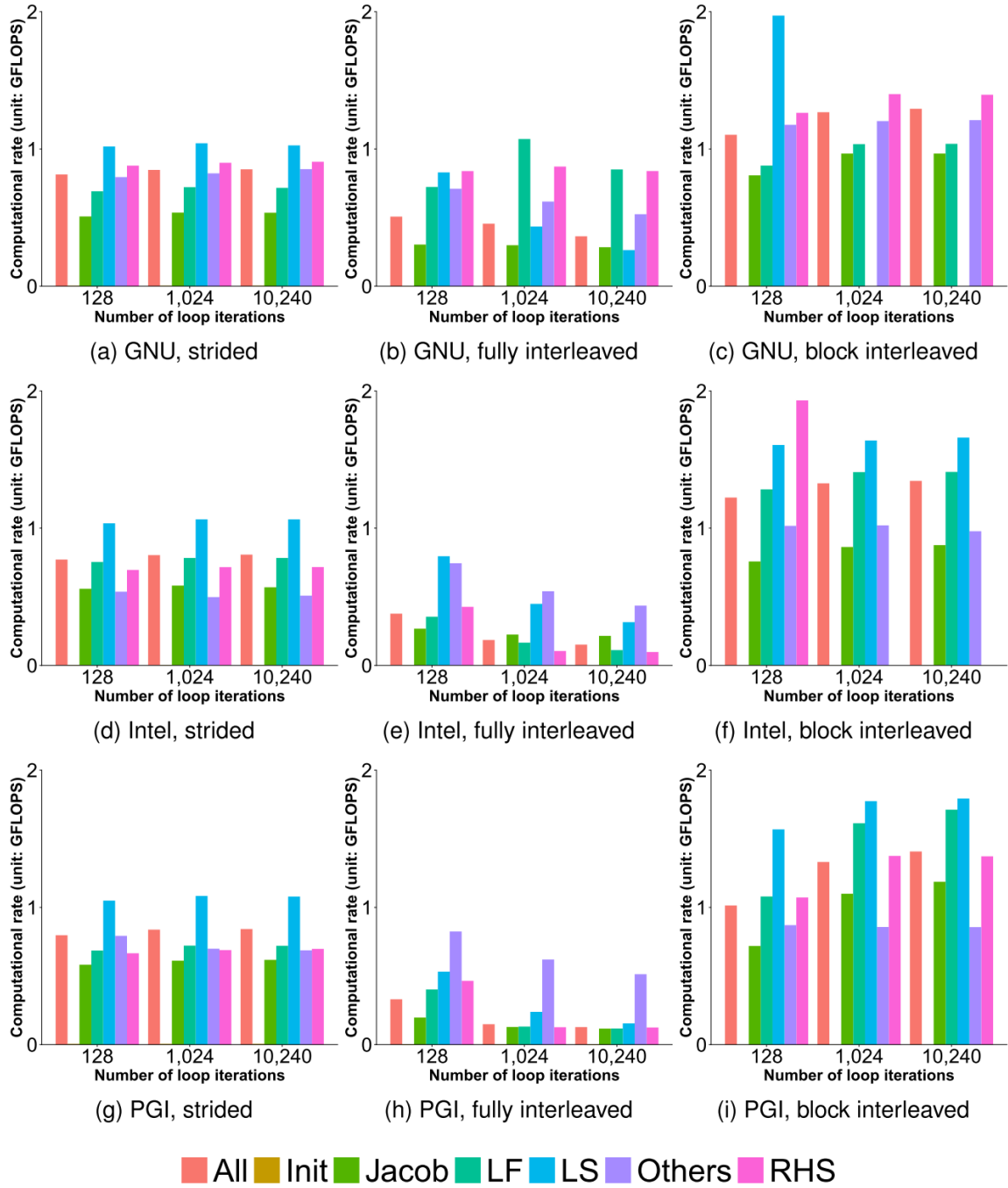


Figure 3.2. Continued.

respectively. Especially, using BIML achieves close computational time as using SML for the PGI compiler when the number of loop iterations grows large (e.g., > 1,000). Using FIML generally yields poor performance here, which may be related to its higher memory bandwidth requirements shown in Figure 3.1. The assembly files (*.s) for the LU factorization are further investigated for the three compilers and memory layouts. Considering that there exist many instructions accessing two far-away elements in the same matrix, it is possible for the SML to perform better compared to the FIML and BIML since the distance in memory between two far-away elements in the same matrix is much shorter in the SML (see Section 3.2.3). On the other hand, the memory alignment directives provided by Intel help generate many “packed double” vector instructions for FIML and BIML, while the instructions are mainly non-vectorized “scalar double” for GNU and PGI. This unique Intel feature assists efficient vectorization and data copy, thus leading to the fastest computational performance for BIML.

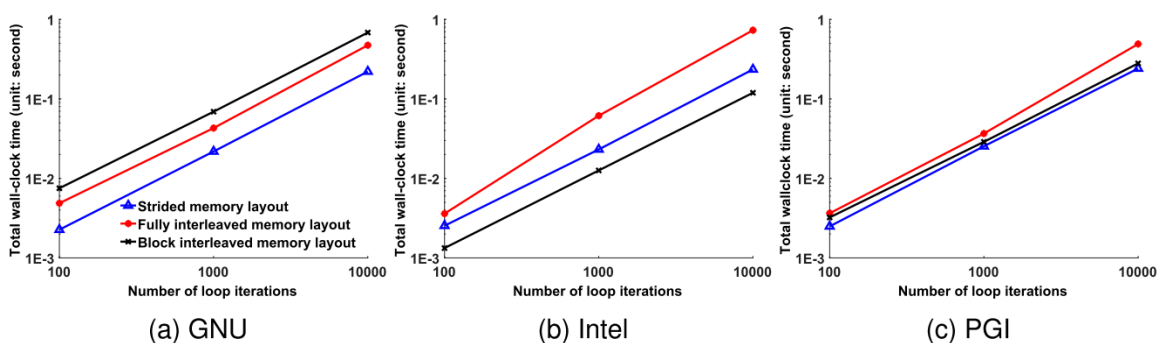


Figure 3.3. The total wall-clock time (Y-axis, log scale, unit: second) of the chemistry box model with different number of loop iterations (X-axis, log scale) for (a) GNU, (b) Intel, and (c) PGI compiler. Different colors refer to different memory layouts (blue: strided, red: fully interleaved, black: block interleaved).

The percent contribution (Y-axis) of each function in the chemistry box model to the total wall-clock time shows that for GNU compiler, formation of the Jacobian matrix consumes the highest amount of time for both SML and FIML (Figure 3.4a and 3.4b)

while LU factorization becomes the most computationally expensive part for BIML (Figure 3.4c). For the Intel compiler (Figure 3.4d, 3.4e and 3.4f), formation of the Jacobian matrix costs the highest amount of time for all three memory layouts, except the FIML with a larger number of loop iterations, where LU factorization dominates the consumption of time. For the PGI compiler (Figure 3.4g, 3.4h and 3.4i), formation of the Jacobian matrix and LU factorization takes similar computational time for both SML and BIML. However, formation of the Jacobian matrix is much more computationally expensive than other functions when using FIML. It is clear that formation of the Jacobian matrix and LU (both factorization and solve) cost more than 70% of the total wall-clock time in most configurations, which is consistent with the previous analysis in Sections 3.3.1. For FIML, the Intel version spends 4× to 7× computational time on the LU factorization and LU solve compared to the GNU and PGI versions when the number of loop iterations is larger than 1,000. This mainly explains the slow computation of FIML with Intel, as shown in Figure 3.3b. Nevertheless, the GNU version with BIML spends 2× to 9× computational time on formation of the Jacobian matrix, LU factorization and LU solve as the Intel and PGI versions. Hence the total wall-clock time of GNU version for BIML is ~5.62× and ~2.39× as that of Intel and PGI versions, even if its memory bandwidth requirement is extremely small (Figure 3.1c). It is clear that the compilers, together with the choice of memory layout, can affect the computational performance significantly. For a given compiler, the largest difference between different memory layouts can be as high as a factor of 6.15 (i.e., Intel: FIML vs. BIML). For a fixed memory layout, the largest difference between different compilers can also be around a factor of 5.75 (i.e., BIML: GNU vs. Intel).

3.3.3 GPU

3.3.3.1 Strided vs. Interleaved vs. Block Interleaved memory layout

The CUDA platform used in this study is cudatoolkit/7.5 on Titan. The NVIDIA visual profiler (NVVP) results of the GPU version of chemistry box model for SML, FIML and BIML show that the LU factorization kernel has already hit the limit of 255 registers per

thread. Therefore, a maximum 256 threads (correspondingly 12.5% occupancy) can be launched simultaneously in each SMX for this kernel in order to not exceed the threshold of 65,536 registers per SMX. Since there are 14 SMXs per GPU on Titan and instructions are executed at the warp (size = 32) level, the number of loop iterations is therefore chosen as a multiple of $14 \times 32 = 448$. For the SML (Figure 3.5a), the computational time is $2.27\times$ to $4.13\times$ the data movement time between CPU and GPU. Except for the data deallocation, the time of both computation and data movement grows linearly with the number of loop iterations. In contrast, for the FIML, the computational time is less than the data movement time except when doing 448 loop iterations (Figure 3.5b). The computational time is $0.47\times$ to $0.76\times$ the data movement time. Similar behavior is also observed for the BIML (Figure 3.5c) but the ratio of computational time over data movement time increases slightly ($0.65\times$ to $0.83\times$). This shows that for the GPU version, using SML requires $1.65\times$ to $2.80\times$ total wall-clock time as that of using BIML and $1.89\times$ to $3.18\times$ total wall-clock time as that of using FIML. This difference is mainly caused by the shorter computational time using BIML and FIML. On the GPU, each thread works on exactly one matrix and the memory bandwidth is much higher than that on the CPU. Hence, FIML and BIML seem to benefit more from the SIMT model than SML. In particular, FIML achieves the fastest computation, which is different from what is observed on the CPU (see Section 3.3.2). The percent contribution of each function shown in Figure 3.6 further confirms that using SML will spend a significant amount of time on formation of the Jacobian matrix and right hand side source term, which can be done very efficiently using FIML. Even for the LU functions (factorization and solve) that seem to consume a large percent of time by using FIML, its actual wall-clock time is still much smaller than the one when using SML. According to Figure 3.6b and 3.6c, the better computational performance of FIML over BIML is mainly attributed to the fast computation of formation of the Jacobian matrix, LU factorization and right hand side source term using FIML. This suggests that FIML is the best choice of memory layout on the GPU for the chemistry box model and thus will be used for the rest examinations in this study.

Figure 3.4. The percent contribution (Y-axis) of each function in the chemistry box model to the total wall-clock time with different number of loop iterations (X-axis) for different compilers ((a-c): GNU, (d-f): Intel and (g-i): PGI) and memory layouts (left-panel: strided, middle-panel: fully interleaved, right-panel: block interleaved).

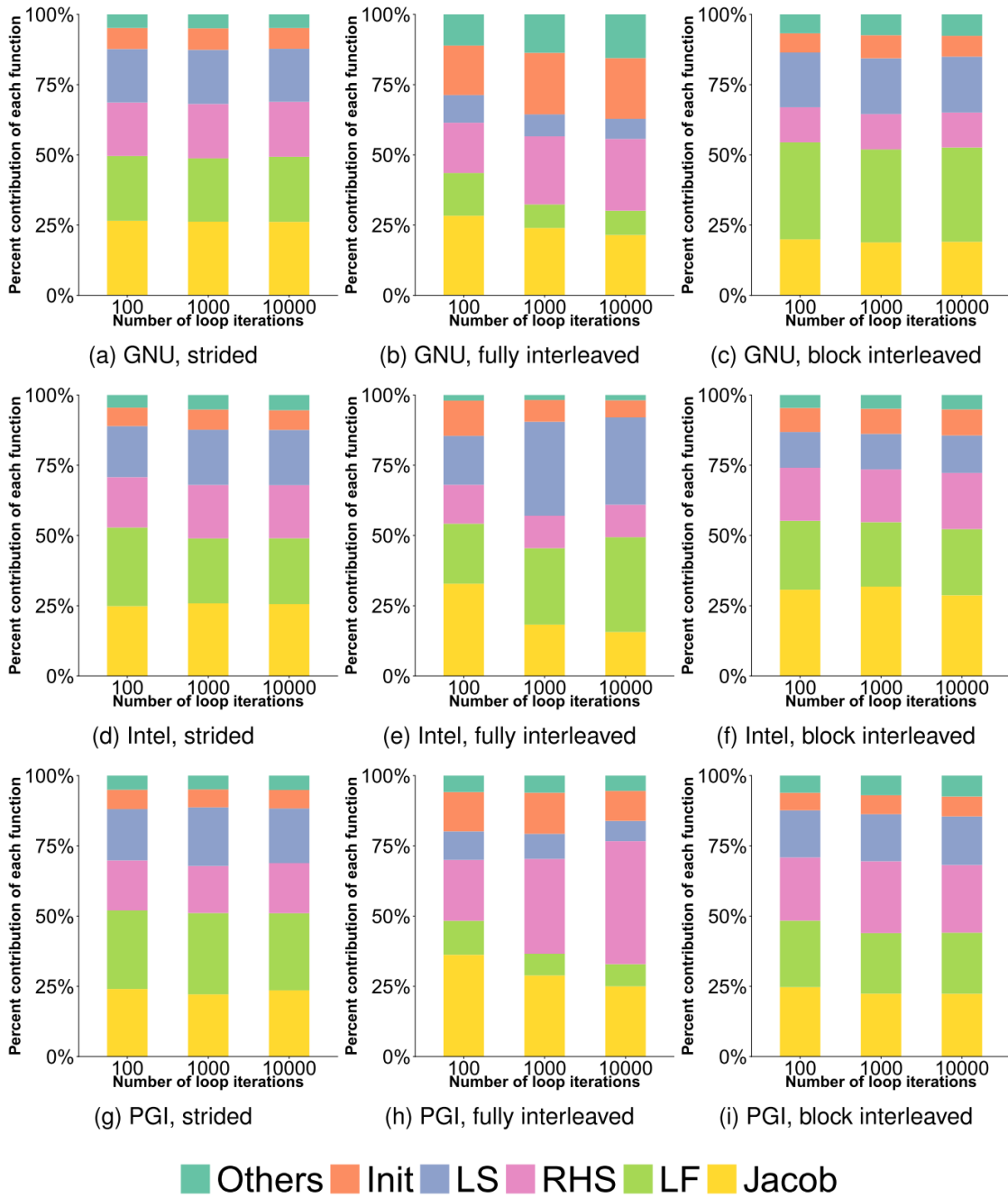


Figure 3.4. Continued.

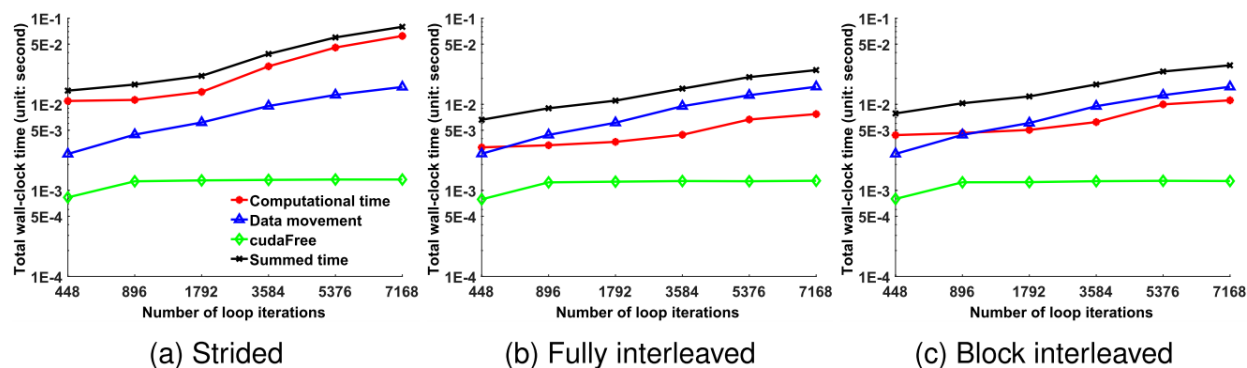


Figure 3.5. The total wallclock time (Y-axis, log scale, unit: second) of the GPU version of chemistry box model with different number of loop iterations (X-axis, log scale) using the (a) strided, (b) fully interleaved and (c) block interleaved memory layouts. Different colors refer to different metrics of time (red: computational time, blue: data movement time between CPU and GPU, green: data deallocation time on the GPU, black: summed time from the three above).

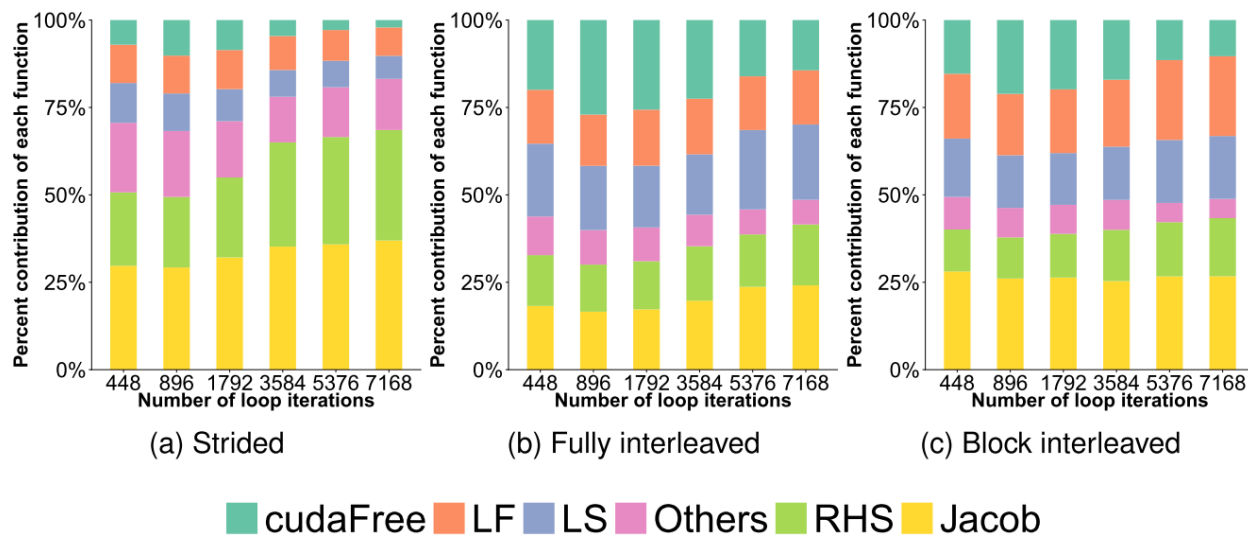


Figure 3.6. The percent contribution (Y-axis) of each function in the GPU version of chemistry box model to the total wall-clock time with different number of loop iterations (X-axis) for the (a) strided, (b) fully interleaved and (c) block interleaved memory layout.

The importance of choosing the thread count wisely is also investigated. For a small number of loop iterations (i.e., 448 loop iterations shown in Figure 3.7), the total wall-clock time using different number of threads, ranging from 4 to 256, differ only slightly from each other. However, when the number of loop iterations increases to 7,168, using a thread count smaller than 32 leads to a clear increase of total wall-clock time. This is reasonable as all the computation on the GPU is executed at the warp level, which equals to 32 threads per block. Using fewer than 32 threads per block will fail to fully use the resources and thus perform worse. On the other hand, using 32, 64, 128 and 256 threads will launch 8, 4, 2 and 1 thread blocks per SMX, respectively, and in this case they behave very close to each other (Figure 3.7). The percent contribution of each function indicates that for 448 loop iterations, using different number of threads per block may take similar time on each function (Figure 3.8a). However, for 7,168 loop iterations, much more time is spent on formation of the Jacobian matrix and LU solve for a thread count smaller than 32 (Figure 3.8b). It appears that using 32 threads per block outperforms slightly and thus is used as the optimal choice for the rest of this study.

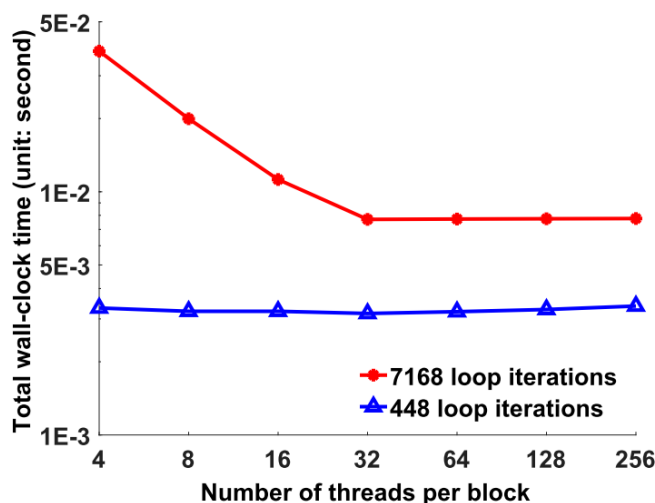


Figure 3.7. The computational time (Y-axis, log scale, unit: second) of the GPU version of chemistry box model with different number of threads per block (X-axis), using the fully interleaved memory layouts (blue: 448 loop iterations; red: 7,168 loop iterations).

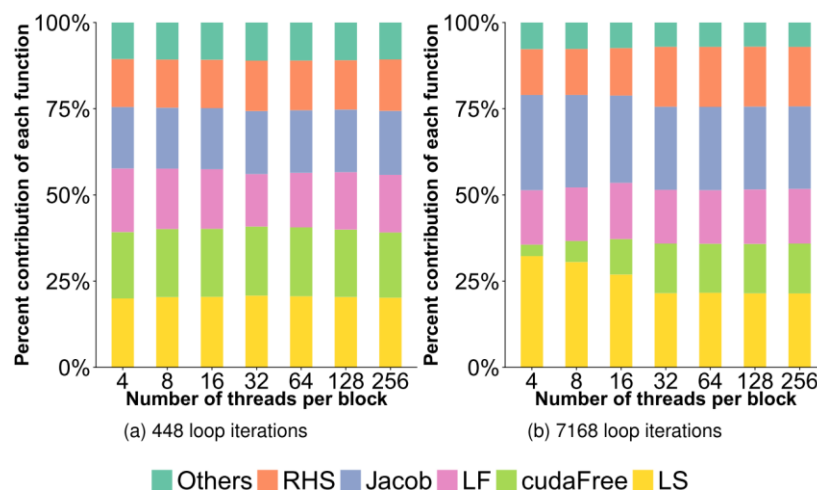


Figure 3.8. The percent contribution (Y-axis) of each function in the GPU version of chemistry box model to the total wall-clock time with different number of threads per block (X-axis) for (a) 448 and (b) 7,168 loop iterations.

3.3.3.2 Multiple kernels vs. one kernel

In the Section 3.3.3.1, the GPU version of the chemistry box model is implemented by launching each function as an individual kernel. According to the NVVP results (Figure 3.9a), there is a clear overhead time between two separate kernels. In order to avoid the overhead time, it is possible to assemble all the functions into one kernel that is launched just once. It seems when the number of loop iterations is smaller than 3,584, using one kernel costs about 96% to 98% of the total wall-clock time as that of using the multiple kernels (Figure 3.10). However, when the number of loop iterations increases to larger than 3,584, using one kernel takes $\sim 1.1\times$ total wall-clock time as that of using the multiple kernels. Although the overhead time between two functions is eliminated by implementing the one kernel version (Figure 3.9b), this does not necessarily always speed up the computation. One potential reason is that the compiler can optimize each kernel separately in the multiple kernels version, while it may not be feasible for the one kernel version with multiple functions. For example, in the one kernel version, the maximum number of threads per SMX is limited by the LU factorization, which is 256 in

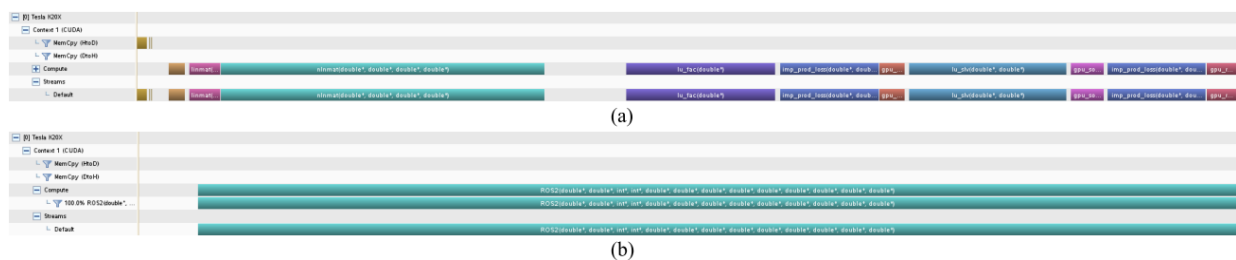


Figure 3.9. The NVIDIA visual profiler (NVVP) results for the GPU version of chemistry box model using the fully interleaved memory layout for (a) multiple kernels and (b) one kernel.

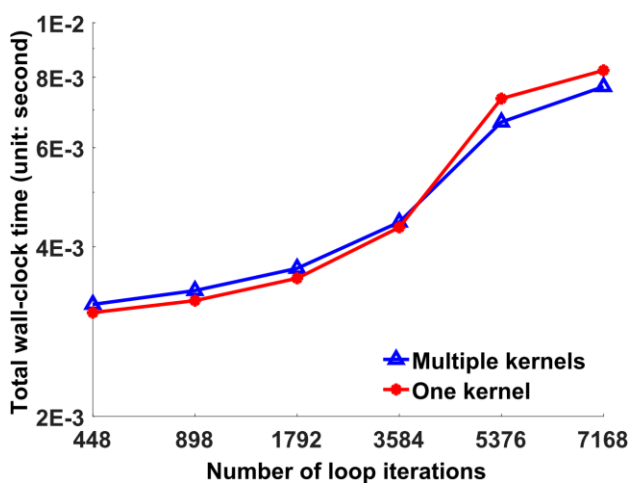


Figure 3.10. The computational time (Y-axis, log scale, unit: second) of the GPU version of chemistry box model with different number of loop iterations (X-axis) for the multiple kernels (blue) and one kernel (red).

this study (see Section 3.3.3.1). However, in the multiple kernels version, the theoretical occupancies for functions like formation of the Jacobian matrix and the right hand side source term can reach as high as 18.8% and 25%, respectively. Therefore, more threads per SMX can be involved for these functions and the computational time is thus reduced, especially for the large number of loop iterations.

3.3.3.3 Shared and constant memory

As mentioned in the Section 3.2.2, there are 48 kilobytes shared memory per SMX and its memory latency is relatively low compared to the GPU's global DRAM memory. Porting some frequently visited arrays to shared memory should increase the memory access speed and thus save some computational time. For the chemistry box model, two arrays are good candidates to port to shared memory: the solution vector ($103 \text{ (number of species)} \times 32 \text{ (number of threads)} \times 8 \text{ (bytes for double precision number)} = 26,368 \text{ bytes or } 25.75 \text{ kilobytes}$) and the intermediate solution vectors ($95 \text{ (number of extracted species)} \times 32 \text{ (number of threads)} \times 2 \text{ (number of stages)} \times 8 \text{ (bytes for double precision number)} = 48,640 \text{ bytes or } 47.5 \text{ kilobytes}$). We port these two vectors separately to shared memory and compare their performances with the one kernel version (note that only the one kernel version is able to effectively exploit shared memory). The NVVP results show that when using shared memory, the theoretical occupancy reaches just 1.6% (corresponding to $1.6\% \times 2,048 \text{ (maximum number of threads per SMX)} = 32 \text{ threads}$) and thus only one thread block can be launched per SMX. The results show that when the number of loop iterations is 448, each SMX will launch only one thread block and using shared memory is faster in this case (Figure 3.11). In particular, porting the intermediate solution vectors to shared memory can save up to 26% of the computational time since it could almost fully use the shared memory. However, when the number of loops increases to 898 or larger, the shared memory version is slower than the no shared memory version by $\sim 4.4\times$ (blue line in Figure 3.11) and $\sim 3.5\times$ (red line in Figure 3.11) when the number of loop iterations equals to 3,584. This is caused by the fact that for the no shared memory version, each SMX can have

256 threads working simultaneously and the total number of working threads is $14 \times 256 = 3,584$. Therefore, the no shared memory version can launch eight thread blocks per SMX (256 (maximum number of threads per SMX) / 32 (number of threads per block) = 8), while the shared memory version can launch only one thread block per SMX as mentioned above. This is also consistent with the observation that the computational time of shared memory version grows linearly with the number of loop iterations, while the computational time of the no shared memory version increases slightly within 3,584 loops (mainly due to the overhead time) but dramatically between 3,584 and 5,376 loop iterations.

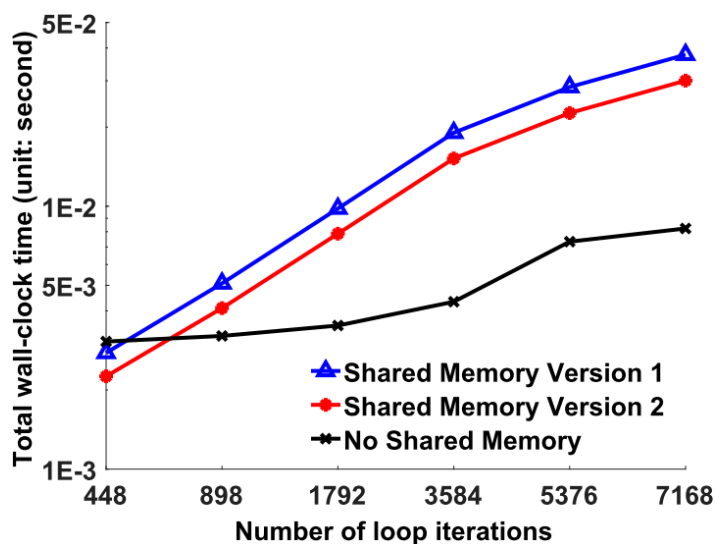


Figure 3.11. The computational time (Y-axis, log scale, unit: second) of the GPU version of chemistry box model with different number of loop iterations (X-axis) using the shared memory for the solution vector (blue), the shared memory for the intermediate solution vectors (red) and no shared memory (black).

Besides the shared memory, there are also 64 kilobytes constant memory that reside in the GPU's global DRAM memory and can be broadcast among all the SMXs. For the chemistry box model in this study, there are two integer mapping arrays used to extract the 95 reaction-active species from the total 103 chemical species and permute it to an

appropriate order with fewer fill-in values [Sun et al., 2017]. In the previous implementation, one copy of these arrays is generated for each thread, which is clearly not necessary since all the values in the mapping arrays are constant. Therefore, some computational time can be saved by storing the mapping arrays in the constant memory where all the SMXs can access them simultaneously. The results show that for the shared memory version which stores the intermediate solution vectors, using constant memory for the mapping arrays could save up to 6.7% of the computational time (blue and green lines in Figure 3.12). For the no shared memory version, using the constant memory can also save 3.6% to 5.2% of the computational time (black and red lines in Figure 3.12). This small improvement is still impressive and worth implementation considering the small size of the mapping arrays ($2 \text{ (number of arrays)} \times 95 \text{ (elements in each array)} \times 4 \text{ (bytes of an integer for a 64-bit system)} = 760 \text{ bytes}$).

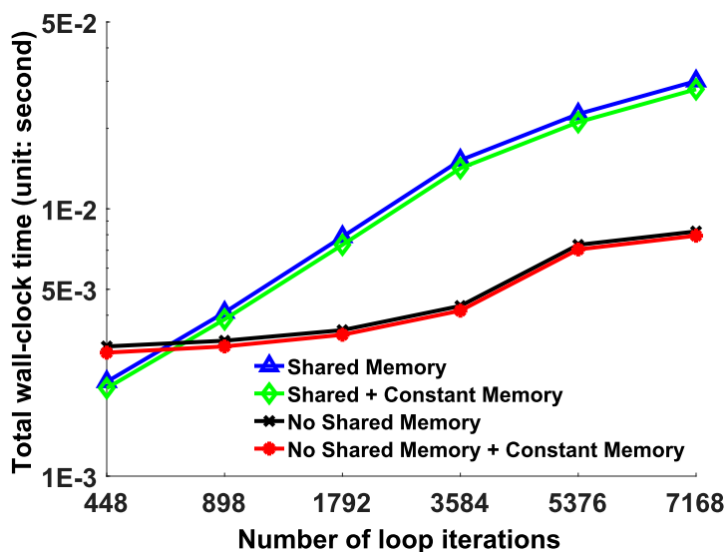


Figure 3.12. The computational time (Y-axis, log scale, unit: second) of the GPU version of chemistry box model with different number of loop iterations (X-axis) using the shared memory for the intermediate solution vectors (blue), shared memory plus constant memory (green), no shared memory (black) and no shared memory but constant memory (red).

3.3.3.4 Stream

A CUDA Stream refers to a queue of work such as kernel launches and memory copies. Operations in the same stream are ordered and can't be overlapped, while operations in different streams can be run in parallel as long as there are no data dependencies between streams. According to the numerical steps in Section 3.2.1, formation of the Jacobian matrix can be done in parallel with the initialization of local data and formation of the right hand side source term in the first stage. Hence, some computational time can be saved by involving CUDA streams here. Note that the streamed version works only with multiple kernels, so we also compare with the computational time of the one kernel version with constant memory (the fastest version so far) to see whether we really benefit from the streamed multiple kernels version. The results indicate that using the streamed multiple kernels with constant memory (red line in Figure 3.13) is likely to save about 4% (7,168 loop iterations) to 16% (448 loop iterations) of the computational time, compared to the previous multiple kernels version (blue line in Figure 3.13). It is also faster than the one kernel version with constant memory (black line in Figure 3.13) but may only save 1.8% (3,584 loop iterations) to 11.2% (5,376 loop iterations) of the computational time. The NVVP results (Figure 3.14) also confirm that two independent streams are successfully created and run in parallel for both the small (448) and large (7,168) number of loop iterations, respectively.

3.3.3.5 Memory copy

In the previous sections, we mainly focus on the optimization of the computational time. As observed in Figure 3.5, the data movement between CPU and GPU can also consume a significant amount of time and its time is even higher than the computational time when using FIML and BIML (Figure 3.5b and 3.5c). Therefore, optimizing the data movement between CPU and GPU is likely to save additional total wall-clock time. Three strategies are investigated here: 1) calling “cudaMalloc” for each array separately; 2) allocating a large space for all the arrays, like mixing ratios of chemical species and reaction rates, so that they are contiguous in the memory locations; 3)

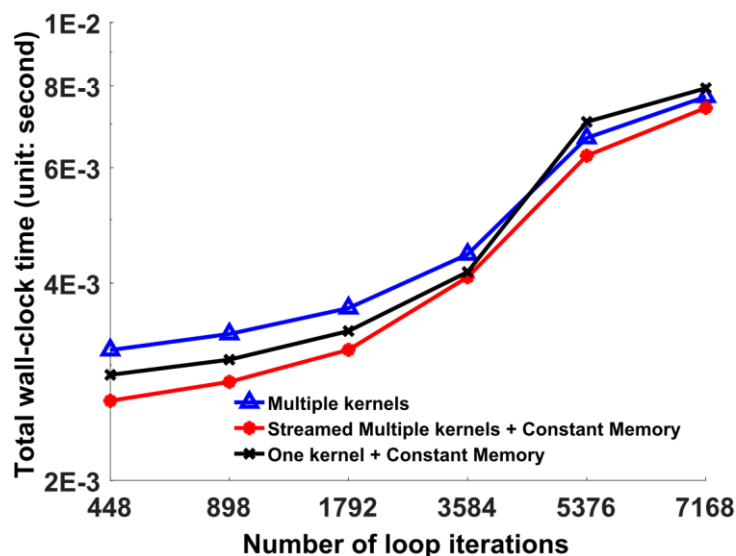


Figure 3.13. The computational time (Y-axis, log-scale, unit: second) of the GPU version of chemistry box model with different number of loop iterations (X-axis) for the multiple kernels (blue), one kernel with constant memory (black) and streamed multiple kernels with constant memory (red).



(a)



(b)

Figure 3.14. The NVVP results for the GPU version of chemistry box model using the streamed multiple kernels + constant memory with (a) 448 and (b) 7,168 loop iterations.

using pinned memory by calling the function “cudaMallocHost” before “cudaMemcpy”. The results show that when allocating all the arrays contiguously in the memory locations (red line in Figure 3.15), it saves about 40% of the time for the 448 loop iterations, compared to the baseline (blue line in Figure 3.15) where each array is allocated separately. When the number of loop iterations increases, the percent of saved time decreases to as low as 10% for the 7,168 loop iterations. When using the pinned memory, it costs $\sim 1.1\times$ data movement time as that of baseline for the 448 loop iterations but increase to $\sim 2.1\times$ for the 7,168 loop iterations (black line in Figure 3.15). The NVVP results indicate that for all the three cases, the copy rate can reach 5.75 GB/s for the 448 loop iterations. When the number of loop iterations grows to 7,168, the copy rate reduces to ~ 3 GB/s for the non-pinned memory but actually increases slightly to 6 GB/s for the pinned memory. However, it also suggests that for the pinned memory, it spends much more time on the “cudaMallocHost” and thus eliminates the benefit of the fast copy rate for different number of loop iterations. On the other hand, it is worth our efforts to allocate a contiguous memory space for all the input arrays, especially for the small number of loop iterations.

Bases on the results above, it seems that the most efficient GPU version of the chemistry box model is using the streamed multiple kernels with constant memory and the contiguous memory allocation for all the arrays. We further compare it with the fastest CPU version (Intel compiler, BIML) and the results show that the CPU version (black line in Figure 3.16) requires $\sim 2.33\times$ computational time as that of GPU version (blue line in Figure 3.16) for the 448 loop iterations. This factor increases rapidly with the number of loop iterations and reaches up to $11.7\times$ for the 7,168 loops. When the time for the data movement between CPU and GPU (green line in Figure 3.16) and “cudaFree” (cyan line in Figure 3.16) is considered, the total wall-clock time of CPU version is still $\sim 1.29\times$ as that of GPU version for the 448 loop iterations and grows to $\sim 3.82\times$ for the 7,168 loop iterations. This clearly shows that the GPU version is superior to the CPU version for the computation alone. When the number of loop iterations is larger than 1,792 and the data movement time is higher than the computational time

alone, the GPU version is still faster than the CPU version and this speed-up is considerable as long as the number of loop iterations is large enough.

3.4 Conclusion

The growing complexity of the global chemistry-climate model increases the computational burden, challenging progress in model development for high resolution simulations. The strong computational power and fast memory access of the GPU in modern supercomputer architectures provides an opportunity to accelerate the computation. The global chemistry-climate model is a natural fit for massive data and instruction parallelism. However, programming for a GPU is difficult and error-prone and limited studies have been done to explore its corresponding benefit to the components of global chemistry-climate model. Therefore, in this study, we port the ROS-2 solver in the chemistry module of CAM4-Chem to the GPU and seek potential speed-up compared with the CPU version. The basic analysis of the chemistry box model reveals that it is not bounded by the computational rate, but by the data access from the CPU to the main memory. Both of these parts can be further accelerated by the GPU.

For the CPU version, different compilers and memory layouts play an important role in the computational time. The Intel compiler with BIML yields the best performance, while the GNU and PGI compilers seem to benefit more from SML. Formation of the Jacobian matrix and LU (both factorization and solve) are shown to be the most time-consuming parts during the chemistry update (around 70%) for most configurations. In contrast, the GPU version benefits more from the FIML. In addition, the computational time of the GPU version increases slowly with the number of loop iterations, opposite to the quick increase of computational time for the CPU version. Tuning the kernel's block size, it is shown that similar performance can be achieved as long as the block is a multiple of the warp size (32 for NVIDIA K20X GPU on Titan). But it will be slower if the block is smaller than the warp size since it doesn't fully utilize all the threads in a warp. The

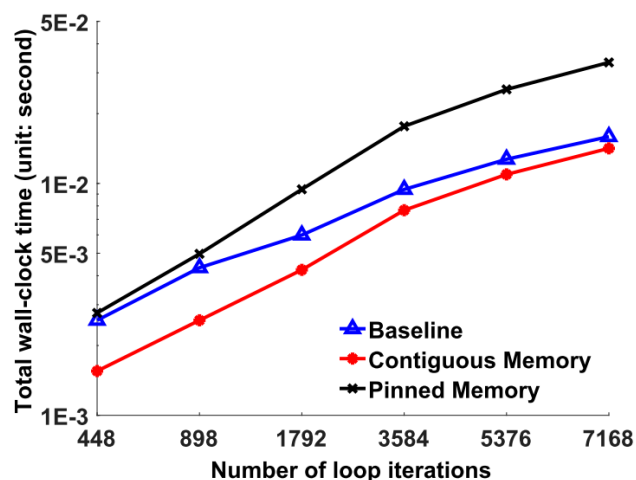


Figure 3.15. The data movement time (Y-axis, log scale, unit: second) of the GPU version of chemistry box model with different number of loop iterations (X-axis) using the separate memory allocation (blue), the contiguous memory allocation (red) and the pinned memory (black).

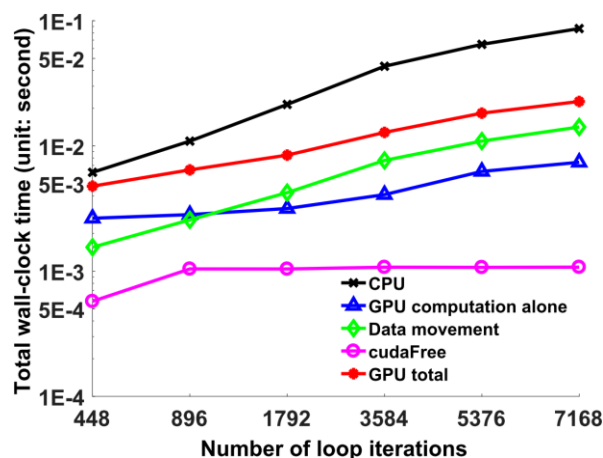


Figure 3.16. The total wall-clock time (Y-axis, log scale, unit: second) of the chemistry box model with different number of loop iterations (X-axis). Different colors refer to different metrics of time (black: total wall-clock time of CPU version, red: total wall-clock time of GPU version, blue: time of computation alone, green: time of data movement between CPU and GPU, cyan: time of data deallocation on the GPU).

multiple kernels version provides better performance for a large number of loop iterations, while the one kernel version runs faster for a small number of loop iterations. The shared memory version yields better performance only for the 448 loop iterations but runs much slower for the larger loop iterations, as it is constrained by the small size of shared memory. Nevertheless, the GPU version can always improve slightly from the usage of constant memory. The best performance of the GPU version is achieved using CUDA streams, which enable the simultaneous execution of independent kernels. The data movement between CPU and GPU, which is also a critical limitation for the overall performance, is done most efficiently by a contiguous allocation when declaring the arrays on the GPU. Using the most optimized configurations from those experiments, the GPU version shows up to 11.7× speed-up of the computational time compared to the CPU version. Even when the data movement between CPU and GPU is considered, the speed-up can be as high as 3.82×. This speed-up is very impressive and makes the GPU version of the chemistry module very promising in the global simulation since 7,168 loop iterations are very close to the loop size in the current configuration of CAM4-Chem (16 columns × 26 levels × 16 cores/node = 6,656 loop iterations). In addition, although we use Titan as a testbed, this optimized configuration can be implemented on other supercomputing platforms as well. However, special attention is required to balance the workload between CPU and GPU. Otherwise the CPU cores are idle while the GPU is busy. Overlapping the computation is beyond the scope of this study since rescheduling for workload balance leads to modification of the main data structure of CAM4-Chem, but we would like to work on that in the near future.

References

- Alvanos, M. and T. Christoudias (2017), GPU-accelerated atmospheric chemical kinetics in the ECHAM/MESSy (EMAC) Earth system model (version 2.52), *Geosci. Model Dev.*, 10(10), 3679-3693, doi: 10.5194/gmd-10-3679-2017.
- Austin, J., Shindell, D., Beagley, S. R., Brühl, C., Dameris, M., Manzini, E., Nagashima, T., Newman, P., Pawson, S., Pitari, G., Rozanov, E., Schnadt, C. and T. G. Shepherd (2003), Uncertainties and assessments of chemistry-climate models of the stratosphere, *Atmos. Chem. Phys.*, 3(1), 1-27, doi: 10.5194/acp-3-1-2003.
- Collins, W. J., Lamarque, J.-F., Schulz, M., Boucher, O., Eyring, V., Hegglin, M. I., Maycock, A., Myhre, G., Prather, M., Shindell, D. and Smith, S. J. (2017), AerChemMIP: quantifying the effects of chemistry and aerosols in CMIP6, *Geosci. Model Dev.*, 10(2), 585-607, doi:10.5194/gmd-10-585-2017.
- Dameris, M. and P. Jöckel (2013), Numerical Modeling of Climate-Chemistry Connections: Recent Developments and Future Challenges, *Atmosphere*, 4(2), 132-156, doi: 10.3390/atmos4020132.
- Dong, T., A. Haidar, P. Luszczek, J. A. Harris, S. Tomov and J. Dongarra (2014), LU factorization of small matrices: Accelerating Batched DGETRF of the GPU, 2014 IEEE Intl. Conf. on High Performance Computing and Communications, 2014 IEEE 6th Intl. Symp. on Cyberspace Safety and Security, 2014 IEEE 11th Intl. Conf. on Embedded Software and Syst. (HPCC, CSS, ICESS), Paris, France, pp. 157-160, doi: 10.1109/HPCC.2014.30.
- Dongarra, J., S. Hammarling, N. J. Higham, S. D. Relton, P. Valero-Lara and M. Zounon (2017), The Design and Performance of Batched BLAS on Modern High-Performance Computing Systems, *Procedia Computer Science*, 108, 495-504, doi: 10.1016/j.procs.2017.05.138.
- Emmons, L. K., S. Walters, P. G. Hess, J.-F. Lamarque, G. G. Pfister, D. Fillmore, C. Granier, A. Guenther, D. Kinnison, T. Laepple, J. Orlando, X. Tie, G. Tyndall, C. Wiedinmyer, S. L. Baughcum and S. Kloster (2010), Description and evaluation of the

Model for Ozone and Related chemical Tracers, version 4 (MOZART-4), *Geosci. Model Dev.*, 3, 43-67, doi:10.5194/gmd-3-43-2010.

Gates, M., J. Kurzak, P. Luszczek, Y. Pei and J. Dongarra (2017), Autotuning batch Cholesky factorization in CUDA with interleaved layout of matrices, 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Lake Buena Vista, FL, pp. 1408-1417, doi: 10.1109/IPDPSW.2017.18.

Horowitz, L. W., S. Walters, D. L. Mauzerall, L. K. Emmons, P. J. Rasch, C. Granier, X. X. Tie, J.-F. Lamarque, M. G. Schultz, G. S. Tyndall, J. J. Orlando and G. P. Brasseur (2003), A global simulation of tropospheric ozone and related tracers: Description and evaluation of MOZART, version 2, *J. Geophys. Res.*, 108(D24), 4784-4812, doi:10.1029/2002JD002853.

Isaksen, I. S. A., Granier, C., Myhre, G., Berntsen, T. K., Dalsøren, S. B., Gauss, M., Klimont, Z., Benestad, R., Bousquet, P., Collins, W., Cox, T., Eyring, V., Fowler, D., Fuzzi, S., Jöckel, P., Laj, P., Lohmann, U., Maione, M., Monks, P., Prevo, A.S.H., Raes, F., Richter, A., Rognerud, B., Schulz, M., Shindell, D., Stevenson, D. S., Storelvmo, T., Wang, W.-C., van Weele, M., Wild, M. and D. Wuebbles (2009), Atmospheric composition change: Climate–Chemistry interactions, *Atmos. Environ.*, 43(33), 5138-5192, doi: 10.1016/j.atmosenv.2009.08.003.

Kinnison, D. E., G. P. Brasseur, S. Walters, R. R. Garcia, D. R. Marsh, F. Sassi, V. L. Harvey, C. E. Randall, L. Emmons, J.-F. Lamarque, P. Hess, J. J. Orlando, X. X. Tie, W. Randel, L. L. Pan, A. Gettelman, C. Granier, T. Diehl, U. Niemeier and A. J. Simmons (2007), Sensitivity of chemical tracers to meteorological parameters in the MOZART-3 chemical transport model, *J. Geophys. Res.*, 112, D20302, doi: 10.1029/2006JD007879.

Kleinman, L. I., Daum, P. H., Lee, Y. N., Nunnermacker, L. J., Springston, S. R., WeinsteinLloyd, J. and J. Rudolph (2001), Sensitivity of ozone production rate to ozone precursors, *Geophys. Res. Lett.*, 28(15), 2903-2906, doi: 10.1029/2000GL012597.

Korwar, S. K., S. Vadhiyar and R. S. Nanjundiah (2013), GPU-enabled efficient executions of radiation calculations in climate modeling, 20th Annual International

Conference on High Performance Computing, Bangalore, 353-361, doi:10.1109/HiPC.2013.6799141.

Lamarque, J.-F., Shindell, D. T., Josse, B., Young, P. J., Cionni, I., Eyring, V., Bergmann, D., Cameron-Smith, P., Collins, W. J., Doherty, R., Dalsoren, S., Faluvegi, G., Folberth, G., Ghan, S. J., Horowitz, L. W., Lee, Y. H., MacKenzie, I. A., Nagashima, T., Naik, V., Plummer, D., Righi, M., Rumbold, S. T., Schulz, M., Skeie, R. B., Stevenson, D. S., Strode, S., Sudo, K., Szopa, S., Voulgarakis, A. and G. Zeng (2013), The Atmospheric Chemistry and Climate Model Intercomparison Project (ACCMIP): overview and description of models, simulations and climate diagnostics, *Geosci. Model Dev.*, 6(1), 179-206, doi: 10.5194/gmd-6-179-2013.

Lin, S. (2004), A "Vertically Lagrangian" Finite-Volume Dynamical Core for Global Models, *Mon. Wea. Rev.*, 132, 2293-2307, doi: 10.1175/1520-0493(2004)132<2293:AVLFDC>2.0.CO;2.

Linford, J. C., J. Michalakes, M. Vachharajani and A. Sandu (2009), Multi-core acceleration of chemical kinetics for simulation and prediction, *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, SC '09*, Association for Computing Machinery, New York, NY, 7, 1-11, doi: 10.1145/1654059.1654067.

Mantell, R. G., C. E. Pitt and D. J. Wales (2016), GPU-Accelerated Exploration of Biomolecular Energy Landscapes, *J. Chem. Theory Comput.*, 12(12), 6182-6191, doi: 10.1021/acs.jctc.6b00934.

Michalakes, J. and M. Vachharajani (2008), GPU Acceleration of Numerical Weather Prediction, *Parallel Processing Letters*, 18(4), 531-548, doi: 10.1142/S0129626408003557.

Mirin, A. A. and P. H. Worley (2011), Improving the performance scalability of the community atmosphere model, *The International Journal of High Performance Computing Applications*, 26(1), 17-30, doi: 10.1177/1094342011412630.

Myhre, G., D. Shindell, F.-M. Bréon, W. Collins, J. Fuglestad, J. Huang, D. Koch, J.-F. Lamarque, D. Lee, B. Mendoza, T. Nakajima, A. Robock, G. Stephens, T. Takemura and H. Zhang, 2013: Anthropogenic and Natural Radiative Forcing. In: *Climate Change*

2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and P.M. Midgley (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.

Schraner, M., Rozanov, E., Schnadt Poberaj, C., Kenzelmann, P., Fischer, A. M., Zubov, V., Luo, B. P., Hoyle, C. R., Egorova, T., Fueglistaler, S., Brönnimann, S., Schmutz, W. and T. Peter (2008), Technical Note: Chemistry-climate model SOCOL: version 2.0 with improved transport and chemistry/microphysics schemes, *Atmos. Chem. Phys.*, 8(19), 5957-5974, doi: 10.5194/acp-8-5957-2008.

Su, H., Cheng, Y. F., Oswald, R., Behrendt, T., Trebs, I., Meixner, F. X., Andreae, M. O., Cheng, P., Zhang, Y. H. and U. Pöschl (2011), Soil Nitrite as a Source of Atmospheric HONO and OH Radicals, *Science*, 333(6049), 1616-1618, doi: 10.1126/science.1207687.

Sun, J., J. S. Fu, J. Drake, J.-F. Lamarque, S. Tilmes, and F. Vitt (2017), Improvement of the prediction of surface ozone concentration over conterminous U.S. by a computationally efficient second-order Rosenbrock solver in CAM4-Chem, *J. Adv. Model. Earth Syst.*, 9, doi:10.1002/2016MS000863.

Tian, W. S. and M. P. Chipperfield (2006), A new coupled chemistry–climate model for the stratosphere: The importance of coupling for future O₃-climate predictions, *Quarterly Journal of the Royal Meteorological Society*, 131(605), 281-303, doi: 10.1256/qj.04.05.

Tilmes, S., Lamarque, J.-F., Emmons, L. K., Kinnison, D. E., Marsh, D., Garcia, R. R., Smith, A. K., Neely, R. R., Conley, A., Vitt, F., Val Martin, M., Tanimoto, H., Simpson, I., Blake, D. R. and N. Blake (2016), Representation of the Community Earth System Model (CESM1) CAM4-chem within the Chemistry-Climate Model Initiative (CCMI), *Geosci. Model Dev.*, 9(5), 1853-1890, doi: 10.5194/gmd-9-1853-2016.

Verwer, J. G., E. J. Spee, J. G. Blom and W. Hundsdorfer (1999), A second-order rosenbrock method applied to photochemical dispersion problems, *SIAM J. Sci. Comput.*, 20(4), 1456–1480, doi:10.1137/S1064827597326651.

Worley, P. H. and J. B. Drake (2005), Performance Portability in the Physical Parameterizations of the Community Atmospheric Model, *The International Journal of High Performance Computing Applications*, 19(3), 187-201, doi: 10.1177/1094342005056095.

Xu, S., Huang, X., Oey, L.-Y., Xu, F., Fu, H., Zhang, Y. and Yang, G. (2015), POM.gpu-v1.0: a GPU-based Princeton Ocean Model, *Geosci. Model Dev.*, 8(9), 2815-2827, doi:10.5194/gmd-8-2815-2015.1

**CHAPTER IV COMPARISON OF HORIZONTAL AND VERTICAL
DISTRIBUTION OF OZONE BETWEEN THE SECOND-ORDER
ROSENBROCK (ROS-2) AND THIRD-ORDER ROSENBROCK (ROS-3)
SOLVERS IN CAM4-CHEM WITH DIFFERENT HORIZONTAL
RESOLUTIONS**

A version of this chapter was prepared to be submitted to the journal of advances in modeling earth systems for review.

Abstract

The second-order Rosenbrock (ROS-2) solver is computationally efficient to reduce the positive bias of surface ozone concentration from CAM4-Chem and it is natural to explore whether we can further benefit from the higher order Rosenbrock-type solver (e.g., third-order Rosenbrock, ROS-3). Based on the study of the chemistry box model, the ROS-3 solver shows better convergence than the ROS-2 solver the first-order implicit (IMP) solver under the stochastic perturbation of initial input. When implemented into CAM4-Chem and compared to the ozonesonde climatology, the ROS-2 solver could reduce the positive bias of vertical ozone concentration from the IMP solver at most regions, while the ROS-3 solver generally enlarges the overestimation. All the three solvers show similar pattern correlation and seasonal variation for a given grid resolution, while the fine resolution (e.g., 0.5°) may provide relatively better pattern correlation than the coarse resolution (e.g., 2.0°) for a given chemical solver. The ROS-2 solver is more computationally efficient than the IMP solver, especially for the 2° resolution. In contrast, the ROS-3 solver is much slower than the IMP solver, especially for the 0.5° resolution. This is mainly attributed to the more frequent time step refinements triggered by the ROS-3 solver, which helps explain its further overestimation of ozone concentration. The ROS-2 solver is suggested as a computationally efficient solver to be adopted in CAM4-Chem with improved prediction of vertical ozone concentration. The high-resolution simulation is also necessary for a better pattern correlation but careful treatment is requested ahead for the lightning NO_x emission.

4.1 Introduction

The ground-level ozone is known as one of the six criteria air pollutants defined by US EPA, which can markedly reduce the crops production, threaten the human health and accelerate the global warming [Bell et al., 2007; Sitch et al., 2007; Wilkinson et al., 2012; Sun et al., 2015]. Numerous researches have been done to investigate the formation of ozone from its precursors including carbon monoxide (CO), methane (CH₄), nitrogen oxides (NO_x), volatile organic compounds (VOCs) in the presence of sunlight [Finlayson-Pitts and Pitts, 1993; Sillman, 1995; Collins et al., 2002; Monks et al., 2005; Monks et al., 2015] and the representation of ozone chemistry is always a key component in the development of global chemistry-climate models [Donner et al., 2011; Lamarque et al., 2011; Shindell et al., 2013] and the regional air quality models [Luecken et al., 2008; Sarwar et al., 2008; Tie et al., 2010]. Although the complexity of global chemistry-climate models has grown exponentially together with our understanding of the ozone mechanism, there still exists significant positive bias for the simulated tropospheric ozone both horizontally [Fiore et al., 2003; Aghedo et al., 2011; Yan et al., 2016] and vertically [Stevenson et al., 2006; Fiore et al., 2009; Tilmes et al., 2012]. The coarse horizontal resolution is partially responsible for the enhanced ozone production [Kumar et al., 1994], since the emissions of ozone precursors are artificially diluted [Sillman et al., 1990; Liang and Jacobsen, 2000] when the model grid size is increased. The coarse vertical resolution is also likely to increase ozone production when the emissions are distributed through the depth of the boundary layer [Murazaki, K. and P. Hess, 2006]. The failure to resolve the steep sub-grid land-to-sea gradients in mixing depth and complex topography that controls the airflow and ventilation could lead to an overestimate of surface ozone concentration along the coastal regions [Fiore et al., 2002]. The existence of heterogeneous reactions of aerosols can enable efficient gas-particle mass transfer and thus influence the ozone production as well [Martin et al., 2003]. Kim et al. [2015] indicate that an underestimate of the cloud fraction would contribute to the overestimate of photolysis rate and thus higher prediction of surface ozone concentration.

Updating the emission inventory of precursors [Fiore et al., 2008; Pan et al., 2014; Travis et al., 2016] and using the multi-model mean results [Eyring et al., 2013; Stevenson et al., 2013; Young et al., 2013] are the common approaches to reduce the biased prediction of ozone, but they are mostly regionally beneficial and the evident bias still remains globally. Developing the model itself is another scientifically sound solution to improve the model performance. One of the main flaws for the global chemistry-climate models is its coarse horizontal resolution and using a finer horizontal resolution turns out to be effective at reducing the bias [Park et al., 2004; Wild and Prather, 2006; Yan et al., 2016] but its computational cost may not be affordable for a long-term simulation. Val Martin et al. [2014, 2015] optimized the dry deposition scheme in the CAM4-Chem and it had significantly reduced the bias of summertime (June, July and August, JJA) surface ozone concentration over the US. Recently, Sun et al. [2017] replaced the original first-order implicit solver in the CAM4-Chem with the second-order Rosenbrock (ROS-2) solver. It reduced bias of surface ozone concentration globally and saved about 50% of the computational time for the update of chemistry, compared with the original first-order implicit solver. These scientific and computational benefits motivate us to seek other solvers in the Rosenbrock family that may further improve the model performance numerically.

The third-order Rosenbrock (ROS-3) solver has been implemented and well evaluated in the global 3-D chemical transport model GEOS-Chem [Henze et al., 2007; Eller et al., 2009] and the regional air quality models such as the Weather Research and Forecasting model coupled with Chemistry (WRF-Chem, [Linford et al., 2009]) and the Community Multiscale Air Quality (CMAQ) modeling system [Sarwar et al., 2013]. Considering the expense of calculating the exact Jacobian matrix at each integration step, the modified Rosenbrock-Wanner (ROW) method [Nørsett and Wolfbrandt, 1979; Steihaug and Wolfbrandt, 1979; Shintani, 1982; Zedan, 1990; Rang and Angermann, 2005] is developed to approximate the Jacobian matrix and use the fixed Jacobian matrix for the next few time integrations. This method may reduce the computational cost but suffer from the order reduction, conservation property and loss of accuracy [Lang, 2001]. Since

it is less likely to carry the information of Jacobian matrix for each grid cell in the global chemistry-climate models, we will not explore the properties of ROW-method for the real simulation of atmospheric chemistry. But it is still natural for us to examine whether we can further benefit from the ROS-3 solver to achieve a better representation of both the distribution and concentration of chemical species (e.g., ozone) in the real atmosphere. Especially, the bias of the vertical profile of ozone between the global chemistry-climate models and ozonesonde data can vary in both sign and magnitude at different regions [Lamarque et al., 2005; Jonson et al., 2010; Zhang et al., 2016], unlike to the systematic overestimation of surface ozone concentration over the continents [Lamarque et al., 2012; Val Martin et al., 2015; Sun et al., 2017]. It is important for us to quantitatively evaluate the performance of the vertical distribution of ozone for the ROS-2 and ROS-3 solver, respectively. In addition, the high-resolution simulation is becoming more feasible for the global climate models [Gent et al., 2010; Bacmeister et al., 2014; Small et al., 2014] and improved performance is mainly observed at the tropical zone and other regions with complex orography. It is also worth investigating the sensitivity of ROS-2 and ROS-3 solver to different horizontal resolutions and its impact on the prediction of ozone concentration.

4.2 Methodology

4.2.1 Global chemistry-climate model

In this study, the global chemistry-climate model, CAM4-Chem, is used for the simulation and comparison. The chemistry in CAM4-Chem can interact with the radiation (no cloud-aerosol feedback) in the Community Atmosphere Model (CAM4, [Neale et al., 2013]), the atmospheric component of the Community Earth System Model (CESM version 1.2.2). The standard TROP_MOZART chemistry mechanism in the CAM4-Chem is inherited from the standard Tropospheric Model for Ozone and Related chemical Tracers (MOZART-4) [Emmons et al., 2010] with the expanded

species types and chemistry-specific parameterization. The further details about CAM4-Chem, together with the evaluation of its representation of atmospheric chemistry, are well documented in Lamarque et al. [2012]. The CAM4-Chem is run with the fully coupled mode, which means the atmosphere, land, ocean and sea-ice components are all active and the chemistry can communicate with the biogeochemical processes in the land and ocean models. The horizontal resolution in this study consists of $1.9^\circ \times 2.5^\circ$, $0.9^\circ \times 1.25^\circ$ and $0.47^\circ \times 0.63^\circ$ (latitude x longitude), respectively. For each horizontal resolution, three case studies using the original first-order implicit (IMP) solver, the ROS-2 solver and the ROS-3 solver are performed separately. Hence there are totally nine case studies required for the intercomparison.

4.2.2 Description of chemical solvers

4.2.2.1 First-order implicit solver

In CAM4-Chem, the equations for chemical species conservation and reaction in the atmosphere take the following autonomous form:

$$\frac{Dy}{Dt} = F(y) \quad (4.1)$$

where $y = [y_1, y_2, \dots, y_N]^T$ is the vector of volume mixing ratios for N species (N here represents the 95 implicit species for the TROP_MOZART mechanism) and the source term $F(y)$ represents the atmospheric chemical reactions (40 photochemical reactions and 172 gas phase reactions). The current first-order implicit (IMP) solver in CAM4-Chem [Kinnison et al, 2007] discretizes the Equation (4.1) by Equation (4.2):

$$\frac{y^{n+1} - y^n}{\Delta t} = F(y^{n+1}) \quad (4.2)$$

Moving all the items to the left hand side, we have:

$$G(y) = \frac{y^{n+1} - y^n}{\Delta t} - F(y^{n+1}) = 0 \quad (4.3)$$

Now solving the Equation (4.1) is simply converted to the problem of finding a root for Equation (4.3). The CAM4-Chem uses the Newton-Raphson method to search for a root

with relatively quick convergence. In other words, it needs to calculate the Jacobian matrix for $G(y)$:

$$J(y) = \frac{dG(y)}{dy} = \frac{1}{\Delta t} - \frac{dF(y)}{dy} \quad (4.4)$$

To begin the iteration, the IMP solver uses $y^0 = y^n$ as an initial guess and inserts it into the Equation (4.5):

$$-J(y^0)\Delta y = R(y^0) = \frac{y^0 - y^n}{\Delta t} - F(y^0) \quad (4.5)$$

By solving Equation (4.5), the updated solution is $y^1 = y^0 + \Delta y$. Assume that the solution is y^k after k^{th} iteration, the IMP solver similarly solves the Equation (4.6) for Δy and then updates $y^{k+1} = y^k + \Delta y$.

$$-J(y^k)\Delta y = R(y^k) = \frac{y^k - y^n}{\Delta t} - F(y^k) \quad (4.6)$$

This iteration is repeated until the absolute value of solution Δy is smaller than the user-defined tolerance.

4.2.2.2 Rosenbrock-type solver

The Rosenbrock-type methods, however, solve the Equation (4.1) in another idea [Hairer and Wanner, 1996]. It originally comes from the idea of the diagonally implicit s-stage Runge-Kutta method that solves the Equation (4.1) by:

$$k_i = hF(y_0 + \sum_{j=1}^{i-1} \alpha_{ij} k_j + \alpha_{ii} k_i), \quad i = 1, \dots, s \quad (4.7)$$

$$y_1 = y_0 + \sum_{i=1}^s \beta_i k_i \quad (4.8)$$

where $h = \Delta t$ is the time step size; k_i is known as the intermediate solution at each stage. Nevertheless, instead of continuing the iterations for Equation (4.7) until convergence, the Rosenbrock-type methods linearize the Equation (4.7) and yield the formula below:

$$k_i = hF(y_0 + \sum_{j=1}^{i-1} a_{ij} k_j) + hA(\sum_{j=1}^i \gamma_{ij} k_j), \quad i = 1, \dots, s \quad (4.9)$$

$$y_1 = y_0 + \sum_{i=1}^s b_i k_i \quad (4.10)$$

where $A = \frac{dF(y_0)}{dy}$ is the Jacobian matrix of $F(y)$ at $y = y_0$. Each stage now contains a system of linear equations with unknown k_i . Re-organize the Equation (4.9) by moving the unknown k_i from right hand side to the left, we obtain:

$$(I - h\gamma_{ii}A)k_i = hF(y_0 + \sum_{j=1}^{i-1} a_{ij}k_j) + hA(\sum_{j=1}^{i-1} \gamma_{ij}k_j) \quad (4.11)$$

Clearly, with the assumption of $\gamma_{11} = \dots = \gamma_{ss} = \gamma$, only one LU-decomposition is needed per step, which would improve the computational efficiency of Rosenbrock-type methods.

In particular, the ROS-2 method takes the following formula [Verwer et al., 1999]:

$$(I - h\gamma A)k_1 = F(y^n) \quad (4.12)$$

$$(I - h\gamma A)k_2 = F(y^n + hk_1) - 2k_1 \quad (4.13)$$

$$y^{n+1} = y^n + \frac{3}{2}hk_1 + \frac{1}{2}hk_2 \quad (4.14)$$

where I is an $N \times N$ identity matrix; y^n and y^{n+1} are the solutions at time $t = t^n$ and t^{n+1} , respectively. Parameter γ is a constant and chosen as $\gamma = 1 + \frac{1}{\sqrt{2}}$ in order to achieve the L-stability and provide better nonlinear stability behavior under a large time step [Verwer et al., 1999].

For the ROS-3 method, the following order conditions for the three-stage calculation must be satisfied:

$$b_1 + b_2 + b_3 = 1 \quad (4.15)$$

$$b_2(a_{21} + \gamma_{21}) + b_3(a_{31} + \gamma_{31} + a_{32} + \gamma_{32}) = \frac{1}{2} - \gamma \quad (4.16)$$

$$b_2a_{21}^2 + b_3(a_{31} + a_{32})^2 = \frac{1}{3} \quad (4.17)$$

$$b_3(a_{21} + \gamma_{21})(a_{32} + \gamma_{32}) = \frac{1}{6} - \gamma + \gamma^2 \quad (4.18)$$

$$b_3(a_{32} + \gamma_{32})a_{21}^2 = \frac{1}{6} - \frac{2}{3}\gamma \quad (4.19)$$

$$\gamma^2 - \gamma + \frac{1}{6} = 0 \quad (4.20)$$

By solving Equation (4.20) we have $\gamma = \frac{3 \pm \sqrt{3}}{6}$. Choosing $\gamma = \frac{3 + \sqrt{3}}{6}$ further provides an A-stability [Table 6.3, Hairer and Wanner, 1996] and thus this value is used for the rest of this study. By choosing the parameters from Equation (4.15) to Equation (4.19) wisely, we can include only two evaluations of the right hand side function and avoid the matrix-vector multiplication during the three-stage calculation, which yields the following form [Lang and Verwer, 2001]:

$$\left(\frac{1}{h\gamma} - J\right) k_1 = F(y_n) \quad (4.21)$$

$$\left(\frac{1}{h\gamma} - J\right) k_2 = F(y_n + a_{21}k_1) + \frac{c_{21}}{h} k_1 \quad (4.22)$$

$$\left(\frac{1}{h\gamma} - J\right) k_3 = F(y_n + a_{31}k_1 + a_{32}k_2) + \frac{c_{31}}{h} k_1 + \frac{c_{32}}{h} k_2 \quad (4.23)$$

$$y_{n+1} = y_n + b_1k_1 + b_2k_2 + b_3k_3 \quad (4.24)$$

The values for the required parameters are listed in Table 4.1. It is evident that the ROS-3 solver can be implemented into the CAM4-Chem based on the framework for ROS-2 solver [Sun et al., 2017]. Therefore, it can also take the advantage of using most original subroutines that have been highly optimized.

Table 4.1. Coefficients for the third-order Rosenbrock method.

Parameter lists	
$\gamma = 0.7886751345948129$	$c_{31} = -3.464101615137755$
$a_{21} = 1.267949192431123$	$c_{32} = -1.732050807568877$
$a_{31} = a_{21}$	$b_1 = 2.0$
$a_{32} = 0$	$b_2 = 0.5773502691896258$
$c_{21} = -1.607695154586736$	$b_3 = 0.4226497308103742$

4.2.3 Port Validation modules in CAM

It is natural for any global atmospheric model that small solution differences may exist due to different machine architecture, compilation or runtime environments. In CAM, there is a port validation module, which designs a series of perturbation growth tests [Rosinski and Williamson, 1997] to obtain a reasonable confidence about solution

differences. The basic idea is to use the “random_number” function in Fortran to generate a group of random numbers from the uniform distribution of [0, 1) and then transform those number to range from (-1, 1] by using the Equation (4.25):

$$\text{pertval} = 2 * (0.5 - \text{pertval}) \quad (4.25)$$

where pertval is the random number sampled from the uniform distribution of [0, 1). The port validation module has been widely used to perturb the initial conditions for the ensemble simulations of CESM and study its variability [Evans et al., 2014; Baker et al., 2015; Kay et al., 2015; Srivver et al., 2015]. In this study, the port validation module from CAM is borrowed to perturb the initial conditions for the chemistry box model, examining the stability of the IMP, ROS-2 and ROS-3 solvers to the stochastic perturbation of initial conditions. Nine scenarios are designed ($\pm 10\%$, $\pm 20\%$, $\pm 30\%$, $\pm 40\%$, $\pm 50\%$, $\pm 60\%$, $\pm 70\%$, $\pm 80\%$, $\pm 90\%$) to perturb the initial conditions and for each scenario, 1000 samples are generated with enough input and reasonable computational cost.

4.3 Results and Discussion

4.3.1 Perturbation of initial concentration

Similar to the previous work done by Sun et al. [2017], the output from some arbitrarily chosen grid cells in CAM4-Chem are used as the initial conditions to drive the chemistry box model and they all yield the similar behavior. Therefore, one grid at Eastern U.S. (latitude = 34.4° , longitude = -90° , level = 26 (bottom layer)) is used to serve as an example for the more realistic state of atmospheric chemistry (e.g., ozone concentration is around 60 ppb). Using the port validation module described in Section 4.2.3, we keep other conditions like reaction rates the same but perturb the initial concentrations of species by $\pm 10\%$, $\pm 20\%$, $\pm 30\%$, $\pm 40\%$, $\pm 50\%$, $\pm 60\%$, $\pm 70\%$, $\pm 80\%$ and $\pm 90\%$, respectively. In this way, we could examine the convergence of three chemical solvers under the stochastic perturbation of initial species concentrations. The ozone concentrations after the integration of one climate time step (1800 seconds) under the

nine perturbation scenarios are first compared. It seems that for the IMP solver with different time step sizes, the results are generally consistent (Figure 4.1). The detailed self-comparison for the IMP solver shows that using the 180 seconds or smaller time step, the largest difference between the predicted ozone concentrations and the reference values (results using the 0.18 second time step) is almost negligible (-0.042 ppb under the $\pm 90\%$ perturbation scenario). For the 1800 seconds time step, as large as -0.33 ppb difference can be observed for the maximum value under the $\pm 90\%$ perturbation scenario. The largest differences for the mean and median values are both around -0.17 ppb ($\pm 90\%$ perturbation scenario) and the largest difference for the minimum value is around -0.14 ppb ($\pm 10\%$ perturbation scenario). For the ROS-2 solver, it is similar that using the time step sizes smaller than 180 seconds produces very close results to the reference values. However, the difference grows much faster with the time step size and when using the 1800 seconds time step, the largest difference can reach up to -1.72 ppb for the maximum value ($\pm 90\%$ perturbation scenario), 0.2 ppb for the minimum value ($\pm 30\%$ perturbation scenario), -0.26 ppb for the mean value ($\pm 90\%$ perturbation scenario) and 0.08 ppb for the median value ($\pm 10\%$ perturbation scenario). The ROS-3 solver behaves similarly as the ROS-2 solver but the largest difference is slightly smaller when using the 1800 seconds time step (-1.13 ppb for the maximum value ($\pm 90\%$ perturbation scenario), -0.04 ppb for the minimum value ($\pm 80\%$ perturbation scenario), -0.22 ppb for the mean value ($\pm 90\%$ perturbation scenario) and -0.03 ppb for the median value ($\pm 90\%$ perturbation scenario)). The comparison of differences between the ROS-2 solver and IMP solver further shows that when the time step size is smaller than the 18 seconds, these two solvers agree well with each other. The largest difference is bounded within ± 0.01 ppb for one climate time step integration. Nevertheless, the largest differences grow rapidly to -0.11 ppb and -1.4 ppb (maximum value under $\pm 90\%$ perturbation scenario) for the 180 and 1800 seconds time step, respectively. The comparison of differences between the ROS-3 solver and IMP solver again yields similar behavior but the largest differences may only reach -0.02 ppb and -0.8 ppb (maximum value under $\pm 90\%$ perturbation scenario) for the 180 and 1800 seconds time step, respectively. Clearly for either the comparison of solver itself or

the comparison between solvers, the largest difference of maximum value is much higher than that of minimum, mean or median value. This is consistent with the previous findings that the errors for both global and regional models respond disproportionately at the high end of the ozone concentration [Tong et al., 2009; Hollaway et al., 2012]. In order to ensure that all the three solvers have solved the system correctly, we check the so-called backward error expressed as:

$$\text{Backward error} = \frac{\|M\hat{x} - b\|}{\|M\| \cdot \|\hat{x}\|} \quad (4.26)$$

where M is the system matrix; b is the right hand side vector; $\hat{x}$ is the solved solution for the problem $Mx=b$; $\|\cdot\|$ defines the vector and matrix norms. The $M\hat{x} - b$ is also recognized as the residual and should be minimized, meaning that the numerical solution is close to the true solution. The backward errors for all the three solvers are smaller than 1.E-16 (machine epsilon), suggesting that all the three solvers have solved the system correctly. Therefore, the big difference of maximum values is related to the nature of the chemical problems rather than the failure to properly solve the system by the solvers. It is well-known that the system of atmospheric chemistry is very stiff [Dabdub and Seinfeld, 1995; Verwer and Simpson, 1995; Sandu and Miehe, 2010], due to the large number of species and the wide range of evolving time scales for different species, varying from millisecond to year. Sun et al [2017] also found that the system matrix for the atmospheric chemistry was generally ill-conditioned, where a tiny change to the system could lead to very different solutions. It is evident that the time step plays a role in the formula of system matrix (Equation (4.3), (4.12-4.13) and (4.21-4.23)) for all the three solvers. In addition, the Rosenbrock-type solver is intrinsically solving a slightly different system matrix (depending on the choice of γ) and right hand side vector than the IMP solver. Considering the stiffness of real atmospheric chemistry, this could explain why the three solvers respond differently to the same initial condition. Since the system matrix of Rosenbrock-type solver will differ more with a larger time step size, it is reasonable to observe the largest differences for all the nine perturbation scenarios at 1800 seconds time step. Note that although the three solvers yield consistent mean and median values under different

perturbation scenarios, the ROS-3 solver can converge to the reference solution more quickly than the ROS-2 solver under large stochastic perturbation of initial concentrations.

4.3.2 Perturbation of reaction rates

In the previous section, the impact of initial species concentrations is examined on the behavior of chemistry box model. It's worth noting that the reaction rates in the realistic atmosphere can vary largely due to the meteorological conditions like temperature and solar radiation. Therefore, it is also necessary to investigate the convergence of three chemical solvers under the stochastic perturbation of reaction rates. The initial species concentrations from the input in Section 4.3.1 are kept the same but the reaction rates are perturbed by $\pm 10\%$, $\pm 20\%$, $\pm 30\%$, $\pm 40\%$, $\pm 50\%$, $\pm 60\%$, $\pm 70\%$, $\pm 80\%$ and $\pm 90\%$, respectively. The ozone concentrations predicted by the IMP solver after the integration of one climate time step agree very well with each other under different perturbation scenarios when the time step size is smaller or equal to 180 seconds (Figure 4.2). The result of self-comparison with the reference values further confirms that even under the $\pm 90\%$ perturbation scenario, the largest difference is still smaller than -0.05 ppb. Using the 1800 seconds time step, the largest difference increases to -0.45 ppb for the maximum value under the $\pm 90\%$ perturbation scenario but is only around -0.18 ppb and -0.17 ppb for the mean and median values, respectively. The largest difference for the minimum value occurs at the $\pm 10\%$ perturbation scenario (-0.13 ppb). On the other hand, the ROS-2 solver yields a slight overestimation of ozone concentration under nine perturbation scenarios for the larger time step size. The self-comparison of ROS-2 solver shows that using the 1800 seconds time step, the largest difference is around 0.12 ppb for the maximum value and about 0.09 ppb for the mean and median values under the $\pm 90\%$ perturbation scenario. The largest difference for the minimum value is about 0.08 ppb under the $\pm 10\%$ perturbation scenario. For the ROS-3 solver, it is also likely to overestimate the maximum value using the larger time step size, but the largest difference is only around 0.02 ppb under the $\pm 90\%$ perturbation scenario. In contrast, the largest differences are negative for the minimum (-0.05 ppb under the $\pm 80\%$

Figure 4.1. Boxplot of ozone concentrations after one climate time step integration (1800 seconds) from the first-order implicit (IMP) solver (black), the second-order Rosenbrock (ROS-2) solver (blue) and the third-order Rosenbrock (ROS-3) solver (red) with different time step sizes. X-axis shows the time step size (unit: second) and Y-axis shows the ozone concentration (unit: ppb). The dots refer to the mean values. (a) to (i) correspond to the scenarios with $\pm 10\%$, $\pm 20\%$, $\pm 30\%$, $\pm 40\%$, $\pm 50\%$, $\pm 60\%$, $\pm 70\%$, $\pm 80\%$ and $\pm 90\%$ perturbation of initial species concentrations, respectively.

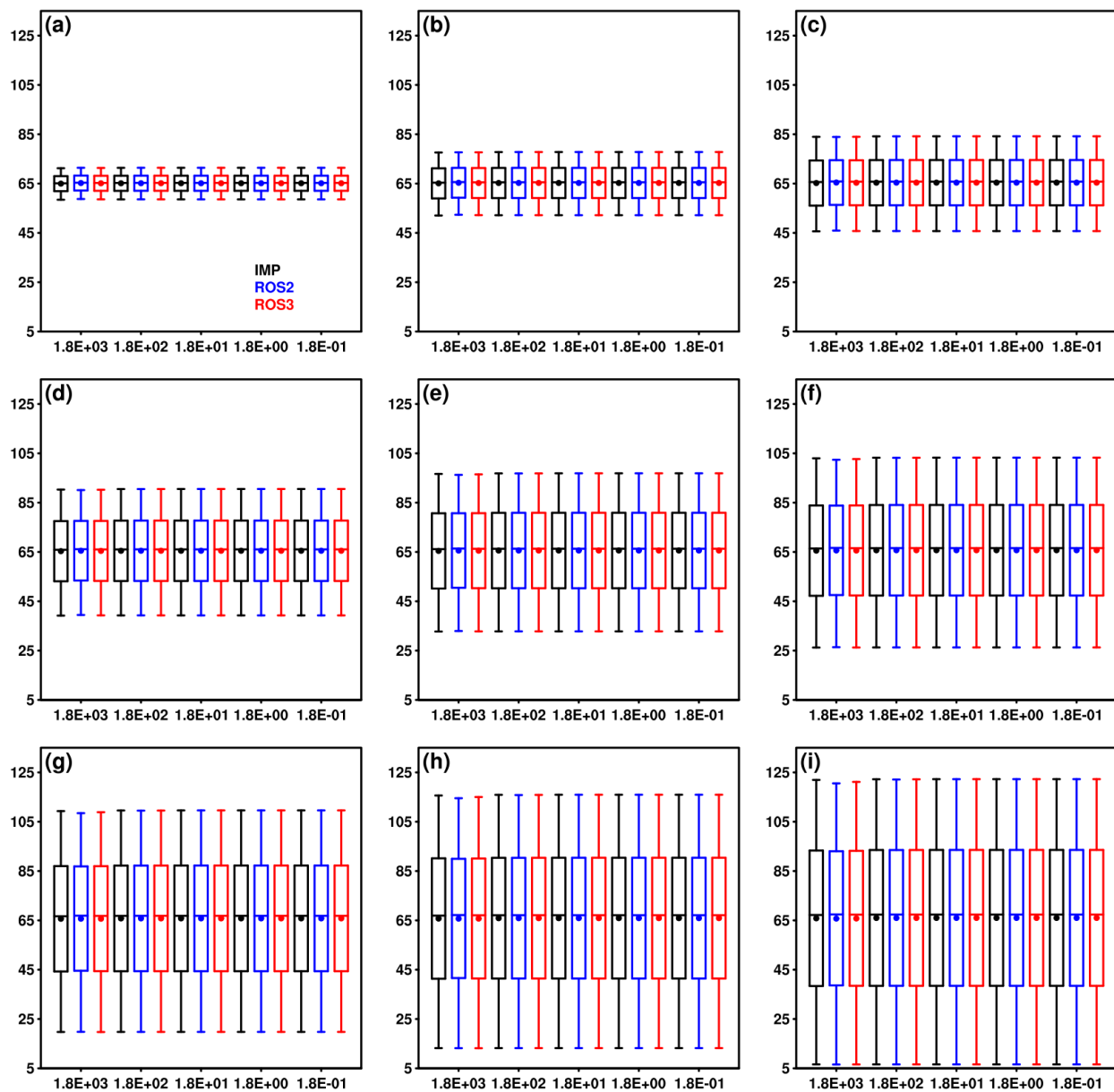


Figure 4.1. Continued.

perturbation scenario), mean (-0.02 ppb under the $\pm 10\%$ perturbation scenario) and median (-0.02 ppb under the $\pm 10\%$ perturbation scenario) values. Comparing the ROS-2 solver to the IMP solver, the largest difference is observed at 1800 seconds time step and varies from 0.21 ppb (minimum value under the $\pm 10\%$ perturbation scenario) to 0.57 ppb (maximum value under the $\pm 90\%$ perturbation scenario). The ROS-3 solver also provides a higher estimation than the IMP solver but the largest difference is smaller for the maximum (0.48 ppb under the $\pm 90\%$ perturbation scenario), minimum (0.11 ppb under the $\pm 90\%$ perturbation scenario), mean (0.17 ppb under the $\pm 90\%$ perturbation scenario) and median (0.15 ppb under the $\pm 90\%$ perturbation scenario) values. This is different from what we have observed in Section 4.3.1, where the Rosenbrock-type solvers generally yield negative largest difference and the magnitude is much larger as well. We again apply the Equation (4.26) to check the backward error and it is still at the magnitude of machine precision. Recalling the chemistry mechanism, the nature of non-linearity comes from the multiplication of volume mixing ratios of species, not the reaction rates. Therefore, the system Jacobian matrix and right hand side forcing terms (e.g., production and loss) respond less non-linearly to the perturbation of reaction rates than the initial concentrations. This leads to the smaller difference between the Rosenbrock-type solvers and the IMP solver even under the $\pm 90\%$ perturbation scenario. However, it is worth noting that the ROS-3 solver converges faster than both ROS-2 solver and IMP solver under different perturbation scenarios, even considering the mean and median values. This suggests that the ROS-3 solver responds better to the stochastic perturbation of reaction rates.

4.3.3 Model evaluation

In the previous study, the ROS-2 solver is found to reduce the positive bias of surface ozone concentration from the IMP solver in CAM4-Chem [Sun et al., 2017]. However, the vertical profile of ozone is not examined in that study, neither the ROS-3 solver. Therefore, it is necessary to further investigate the impact of different chemical solvers on the vertical distribution of ozone concentration. The ozone climatology data used for

model evaluation in this study is provided by Tilmes et al. [2012], which integrates the long-term high quality ozone soundings from 42 stations around the globe and further identifies 12 regions using the Hellinger distance. More details about the ozonesonde launch locations, screening criteria and data evaluation can be found in Tilmes et al. [2012] and references therein. Due to the computational limit, a 10-year simulation (1995-2004) is done for each solver and horizontal resolution, respectively. We restrict our model evaluation up to the height of tropopause (~200 hPa), since the TROP_MOZART chemistry mechanism used in this study is mainly designed for the troposphere.

For the annual median surface ozone concentration averaged between 1995 and 2004 (solid line in Figure 4.3), it is clearly overestimated by the IMP solver (>10 ppb) at the North Polar. The overestimation reduces slightly at the North Mid-Latitude region, ranging from 3.44 ppb (Japan) to 10.38 ppb (Eastern US). The largest variation appears at the Tropics/Subtropics, varying from 0.59 ppb (West Pacific/East Indian Ocean) to 19.96 ppb (North Hemisphere Subtropics). For the South Mid-Latitudes and South Polar, the overestimation of surface ozone concentration is relatively small (~4 ppb). The overestimation of vertical ozone concentration at the North Polar, North Subtropics, Tropics and South Polar yields similar trend, which decreases from 1000 hPa to 800 hPa and then increases along with the altitude. For the North Mid-Latitudes, the overestimation of vertical ozone concentration generally increases from the surface level. The overestimation of vertical ozone concentration is higher than that of surface ozone concentration from the 450 hPa and above for all the regions. However, the overestimation is reduced when the altitude is close to the height of tropopause (200 hPa) and even underestimation of ozone concentration is observed at the North Polar, Canada, South Mid-Latitudes and South Polar. The ROS-2 solver reduces the overestimation of surface ozone concentration from the IMP solver at all the regions, with the largest reduction occurs at Eastern US (-2.2 ppb). Considering the vertical ozone concentration, the ROS-2 solver also reduces the overestimation by the IMP solver but the magnitude of reduction is smaller than that at the surface level (<1 ppb

Figure 4.2. The same as Figure 1 but for the perturbation of initial reaction rates.

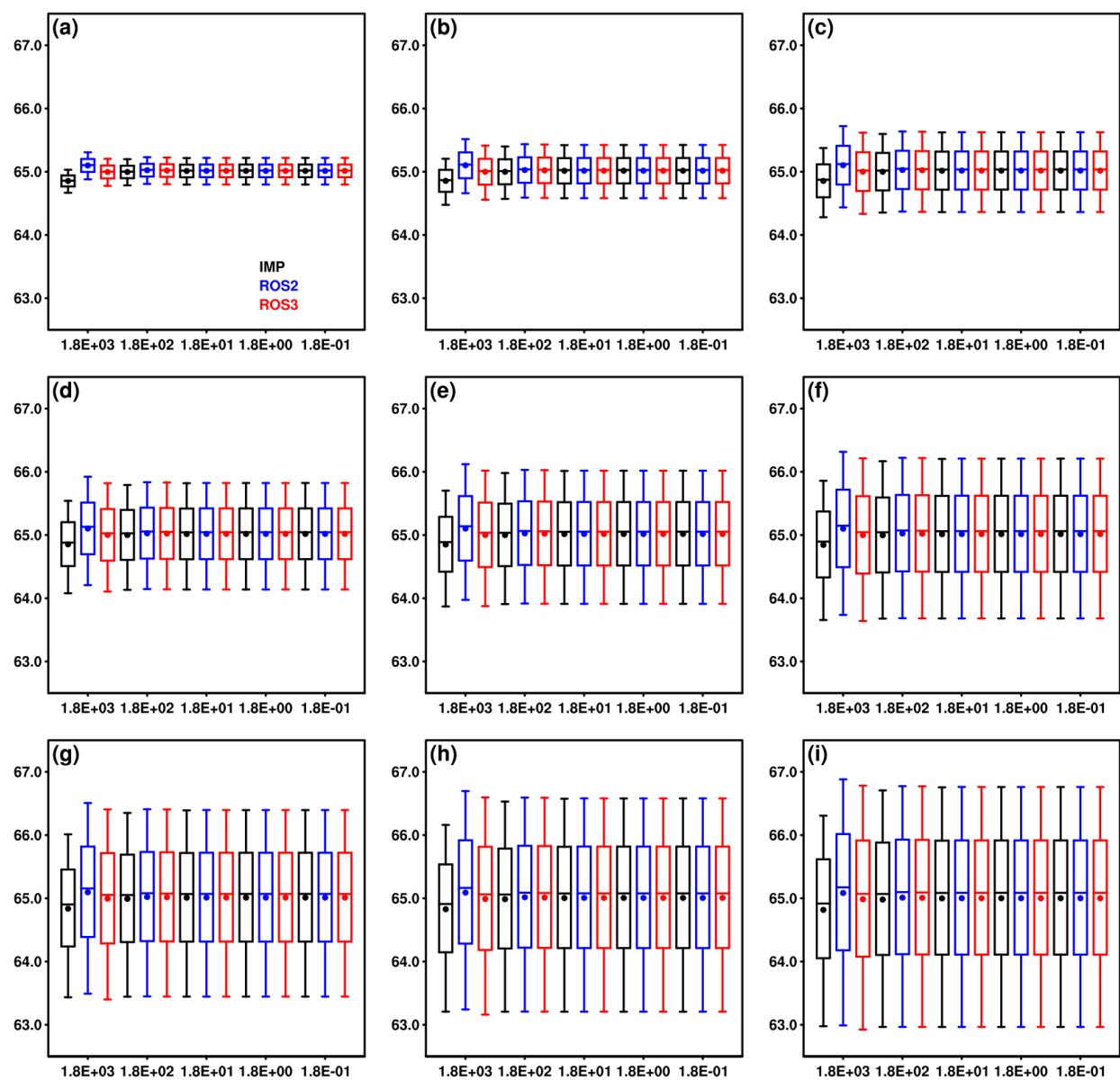


Figure 4.2. Continued.

from the 700 hPa and above). In contrast, the ROS-3 solver generally amplifies the overestimation of surface ozone concentration from the IMP solver at most regions except the North Mid-Latitudes. The reduced overestimation can be as much as -1.29 ppb (Eastern US) while the increased overestimation can be as high as 1.35 ppb (South Polar). The ROS-3 solver overestimates the vertical ozone concentration than the IMP solver from the 900 hPa and above, and the overestimation can reach up to 1.89 ppb at the 600 hPa of Eastern US.

The median surface ozone concentration of JJA averaged from 1995 to 2004 is also overestimated by the IMP solver at most regions except the North Polar and Japan (dash line in Figure 4.3). The overestimation during JJA is even higher than that for the annual median value at Eastern US (13.79 ppb), West Pacific/East Indian Ocean (5.64 ppb), Atlantic/Africa (10.77 ppb), South Mid-Latitudes (6.8 ppb) and South Polar (5.34 ppb). On the other hand, the average underestimation of surface ozone concentration is about -7.99 ppb at the North Polar and as much as -12.99 ppb at Japan. The vertical ozone concentration predicted by the IMP solver generally increases along with the altitude, which provides a better estimation at the North Polar and Japan since the ozone there is underestimated. However, the ozone concentration is eventually overestimated everywhere from the 600 hPa and above, and the overestimation could be larger than 40 ppb at 350 hPa. When the altitude is close to the height of tropopause, underestimation of ozone concentration is again observed at the North Polar, Canada, Western Europe, South Mid-Latitudes and South Polar. In contrast, the ROS-2 solver still predicts lower surface ozone concentration than the IMP solver over all the regions, with the large reduction at Japan (-2.74 ppb) and Eastern US (-2.69 ppb). Since the surface ozone concentration is underestimated at North Polar and Japan by the IMP solver, the lower estimation from the ROS-2 solver actually worsens the prediction there. However, the positive bias over the other regions is still reduced. For the vertical ozone concentration, the ROS-2 solver also yields a lower estimation and generally improves the prediction from the 600 hPa and above, compared to the IMP solver. In addition, unlike the annual median ozone concentration, the reduction of

overestimation of vertical ozone concentration at certain region (e.g., Eastern US) could still be larger than 1 ppb even when the height is close to the tropopause. The ROS-3 solver again amplifies the overestimation of surface ozone concentration by the IMP solver at most regions except the North Mid-Latitudes. The reduced overestimation can be as much as -1.78 ppb at Eastern US while the increased overestimation can be as high as 1.4 ppb at the South Polar. Since the IMP solver underestimates the surface ozone concentration at the North Polar, the ROS-3 solver slightly improves the prediction by 0.23 ppb. The vertical ozone concentration predicted by the ROS-3 solver is higher than that by the IMP solver from 900 hPa and above over all the regions. This could improve the model prediction slightly below 900 hPa at the North Polar and from surface to 700 hPa at Japan, but generally worsen the model prediction from 700 hPa and above over all the regions. This overestimation can reach up to 1.89 ppb at the 600 hPa of Eastern US, compared to the IMP solver. It turns out that ROS-2 solver yields a lower estimation of ozone concentration than the IMP solver at different altitudes, reducing the overestimation at most regions. On the other hand, the ROS-3 solver only reduces the overestimation of surface ozone concentration by the IMP solver at the North Mid-Latitudes and the underestimation of ozone concentration below 900 hPa at the North Polar and below 700 hPa at Japan. The bias is generally amplified elsewhere, especially for the vertical ozone concentration. The main reason is the frequent time step refinements involved by the ROS-3 solver in the CAM4-Chem simulation, which could lead to a higher estimation of ozone concentration based on the results from Section 3.1 and 3.2. This also explains the slow computational performance of ROS-3 solver as discussed in the Section 4.3.5. Therefore, the evaluation here suggests that although ROS-3 solver shows overall good convergence with respect to the stochastic perturbation of initial conditions (Section 4.3.1 and 4.3.2), the ROS-2 solver yields better prediction in the CAM4-Chem simulation, since the ozone concentration is widely overestimated and the ROS-2 solver tends to provide a lower estimation of ozone concentration.

Figure 4.3. Vertical distributions (median) of ozone concentration (unit: ppb) averaged for the whole year (solid line) and June-July-August (JJA, dash line) from 1995 to 2004 for the troposphere. Different colors mean the ozone soundings (black), IMP solver (green), ROS-2 solver (blue) and ROS-3 solver (red).

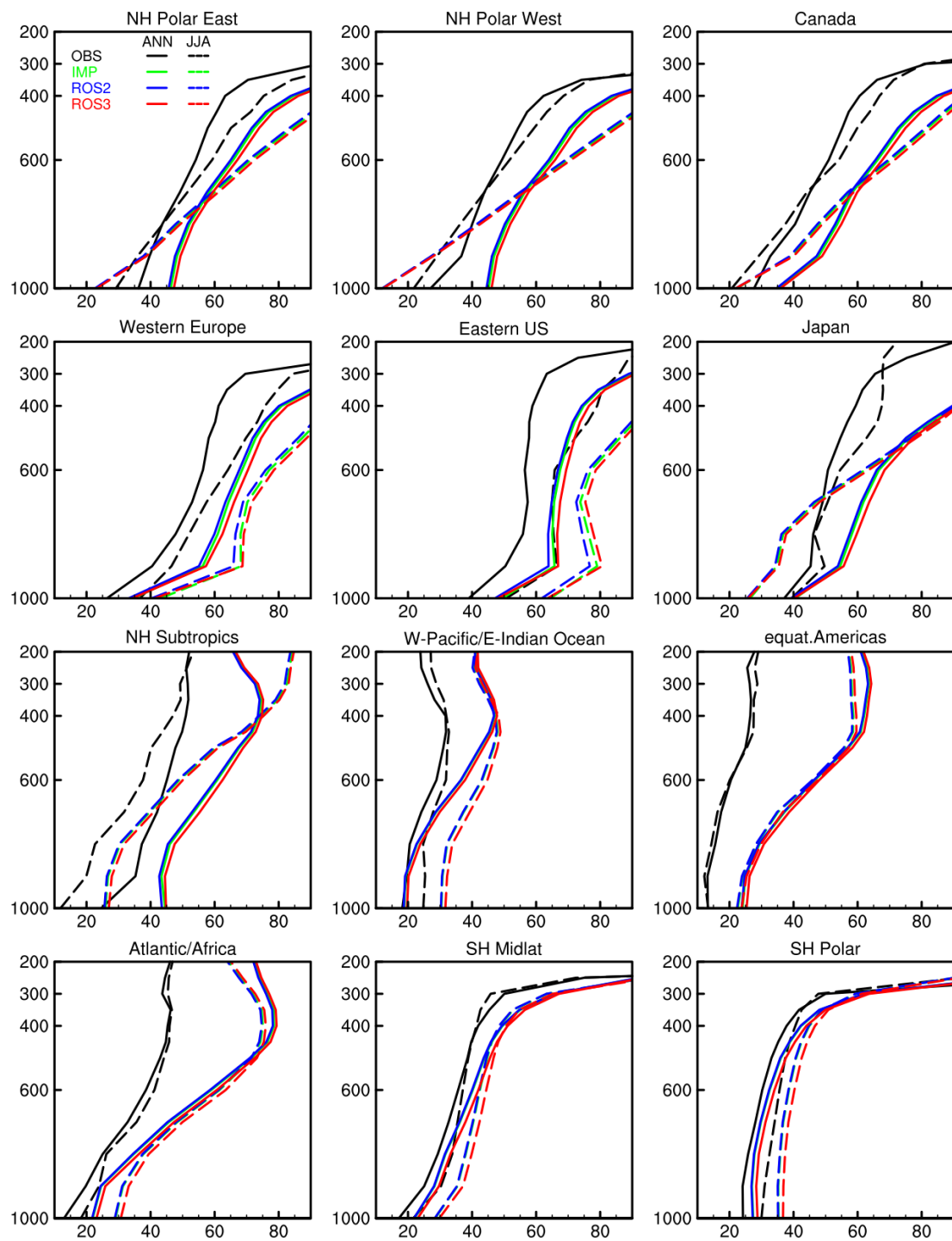


Figure 4.3. Continued.

4.3.4 Effect of grid resolution

As mentioned above, the fine resolution is now more affordable for the global climate simulation. The 10-year results for the three chemical solvers in Section 4.3.3 are generated at the $0.94^{\circ} \times 1.25^{\circ}$ (1°) horizontal resolution. In this section, the 10-year results for the three chemical solvers generated at the $0.47^{\circ} \times 0.625^{\circ}$ (0.5°), $0.94^{\circ} \times 1.25^{\circ}$ and $1.9^{\circ} \times 2.5^{\circ}$ (2°) horizontal resolution are cross-compared to evaluate the impact of grid resolution on the prediction of ozone concentration. The Taylor diagram [Taylor, 2001; Gleckler et al., 2008] is used to examine the correlation and ratio of standard deviation of seasonal median ozone concentrations between the simulated and ozonesonde climatology data from 1995 to 2004 (Figure 4.4). At the high latitudes (North Polar, Canada and South Polar), the 1° and 2° resolutions show a better correlation with ozonesonde climatology data for each chemical solver at 1000 hPa. The 2° resolution yields the smallest standard deviation while the 0.5° resolution yields the largest one, which suggests that the differences between seasons are larger for the finer resolution. Given a fixed resolution, ROS-2 solver performs best with respect to both correlation and seasonal variation, while the ROS-3 solver behaves relatively worse. At 500 hPa, the 1° resolution shows slightly higher correlation and smaller seasonal variation, but the 0.5° resolution still yields larger seasonal variation. The ROS-2 solver again outperforms over the other two solvers, especially using the 2.0° resolution. At 250 hPa, the 0.5° resolution shows highest correlation with the ozonesonde climatology data and the 0.5° and 1° resolutions yield relative small seasonal variation. All the three chemical solvers agree well with each other at different resolutions. At the middle latitudes (Eastern US, Western Europe, Japan and South Mid-Latitudes), the 0.5° resolution shows higher correlation than the coarser resolutions, as well as the seasonal variation, at 1000 hPa and 500 hPa, respectively. For a given resolution, the ROS-2 solver provides highest correlation and smallest seasonal variation while the IMP solver generally performs worst. At 250 hPa, the 0.5° resolution shows highest correlation but smallest seasonal variation, while the 2° resolution shows the largest seasonal variation. A good agreement among the three chemical solvers is again observed at different resolutions. At the Tropics (North Subtropics, West

Pacific/East Indian Ocean, equatorial Americas and Atlantic/Africa), the 0.5° and 1° resolutions yield a higher correlation than the 2° resolution while the seasonal variations are generally the same for different resolutions at 1000 hPa. The ROS-2 solver still behaves better than the other two solvers under all three horizontal resolutions and the ROS-3 solver performs slightly better than the IMP solver at the 0.5° resolution. At 500 hPa, different resolutions yield similar correlation with the ozonesonde climatology data, but the 2° resolution shows smallest seasonal variation while the 0.5° resolution shows the largest one. At 200 hPa, the 0.5° resolution yields relatively higher correlation while the 2° resolution shows smaller seasonal variation than the finer resolution. At both 500 hPa and 250 hPa, all the three chemical solvers do not differ clearly from each other for a given horizontal resolution. The results here indicate that high-resolution simulation (i.e., 0.5°) is likely to improve the pattern correlation with the ozonesonde climatology data, especially at the middle latitudes and Tropics. However, it may also increase the simulated seasonal variation, especially under the 500 hPa. We further investigate the regions in the North Mid-Latitudes, where the population is dense and the air pollution issue is severe. It confirms that using the finer resolution can provide a better pattern correlation with the ozonesonde climatology data at most regions and altitudes (Figure 4.5). However, the finer resolution also leads to a higher estimation of ozone concentration, exacerbating the positive bias from the coarse resolution at most regions. The difference of 10-year averaged seasonal mean temperature between different resolutions (Figure 4.6 and 4.7) is bounded within ± 1 K for the Eastern US, Western Europe and Japan, which implies that the meteorology is less likely to be the driver for the difference of ozone concentration there. In contrast, the monthly mean global flash frequencies of the 0.5° and 1.0° resolutions are about nine and three times as that of the 2.0° resolution, which causes a much higher lightning NO_x emission for the finer resolution. We turn off the lightning NO_x emission in CAM4-Chem and re-run one month for the three resolutions (Figure 4.8). The ozone concentration generated from the finer resolution (i.e., 0.5° and 1.0°) is now much lower than that from the coarse resolution (i.e., 2.0°), which should significantly reduce the positive bias shown in Figure 4.5. Due to the limitation of computational resource, we cannot run the new simulations for

different resolutions with scaled lightning NO_x emission. However, the lightning NO_x emission is clearly an important factor that needs to be evaluated carefully when we go to the finer resolution.

4.3.5 Computational cost

When the global chemistry-climate model goes into the high-resolution simulation, its computational burden become heavier correspondingly. Previous study suggested that ROS-2 solver was about twice fast as the IMP solver [Sun et al., 2017] and the speed-up was stable for long-term simulation. In this section, we will investigate whether the computational performance of the three chemical solvers respond differently to the various horizontal resolutions. One-month simulation is done for comparison and a uniform number of processors (i.e., 2,048) is used for all the test. The result shows that using the IMP solver, the summed computational time for the 0.5° and 1.0° resolutions is 16.33 and 4.1 times as that for the 2.0° resolution, respectively (Table 4.2). This factor increases slightly to 17.51 and 4.18 for the ROS-2 solver, but 17.33 and 3.97 for the ROS-3 solver. For the 2.0° resolution, the ROS-2 solver achieves ~1.81× speed-up compared to the IMP solver, and this factor reduces slightly to ~1.67× and ~1.78× for the 0.5° and 1.0° resolutions, respectively. In contrast, the ROS-3 solver costs 4.42 times computational hours compared to the IMP solver for the 2.0° case. This factor varies from 4.29 to 4.7 when the resolution goes from 1.0° to 0.5°. These results suggest that refining the horizontal grid from 2.0° to 1.0° leads to an increase of computational time by a factor of four for all the three solvers. However, further refining from 1.0° to 0.5° causes 3.98 times increase of computational time for the IMP solver, but 4.22 and 4.36 times for the ROS-2 and ROS-3 solvers, respectively. The slight increase in the computational time for the Rosenbrock-type solvers in the simulation with finer resolution is mainly contributed by the time step refinement during the chemistry update in each time integration. In order to achieve the stability for a non-linear problem and avoid the negative solution for the species concentrations, the Rosenbrock-type solvers request more frequent time step refinements than the IMP

Figure 4.4. Taylor diagram for regions at the High Latitudes (left panel), Mid-Latitudes (middle panel), and Tropics (right panel) with three different altitudes (top row: 1000 hPa, middle row: 500 hPa, bottom row: 250 hPa) in the troposphere. The 10-year results for the three chemical solvers generated at the $0.47^\circ \times 0.625^\circ$ (0.5 degree), $0.94^\circ \times 1.25^\circ$ (1.0 degree) and $1.9^\circ \times 2.5^\circ$ (2.0 degree) horizontal resolution are compared to the ozonesonde climatology data. The radial distance from the origin is the ratio of the standard deviation of simulated ozone concentrations over the standard deviation of ozonesonde climatology data. The azimuthal position is the correlation coefficient of seasonal median ozone concentrations between the simulated ozone concentrations and the ozonesonde climatology data. Different colors mean different chemical solvers (blue: ROS-2 solver, green: IMP solver, red: ROS-3 solver).

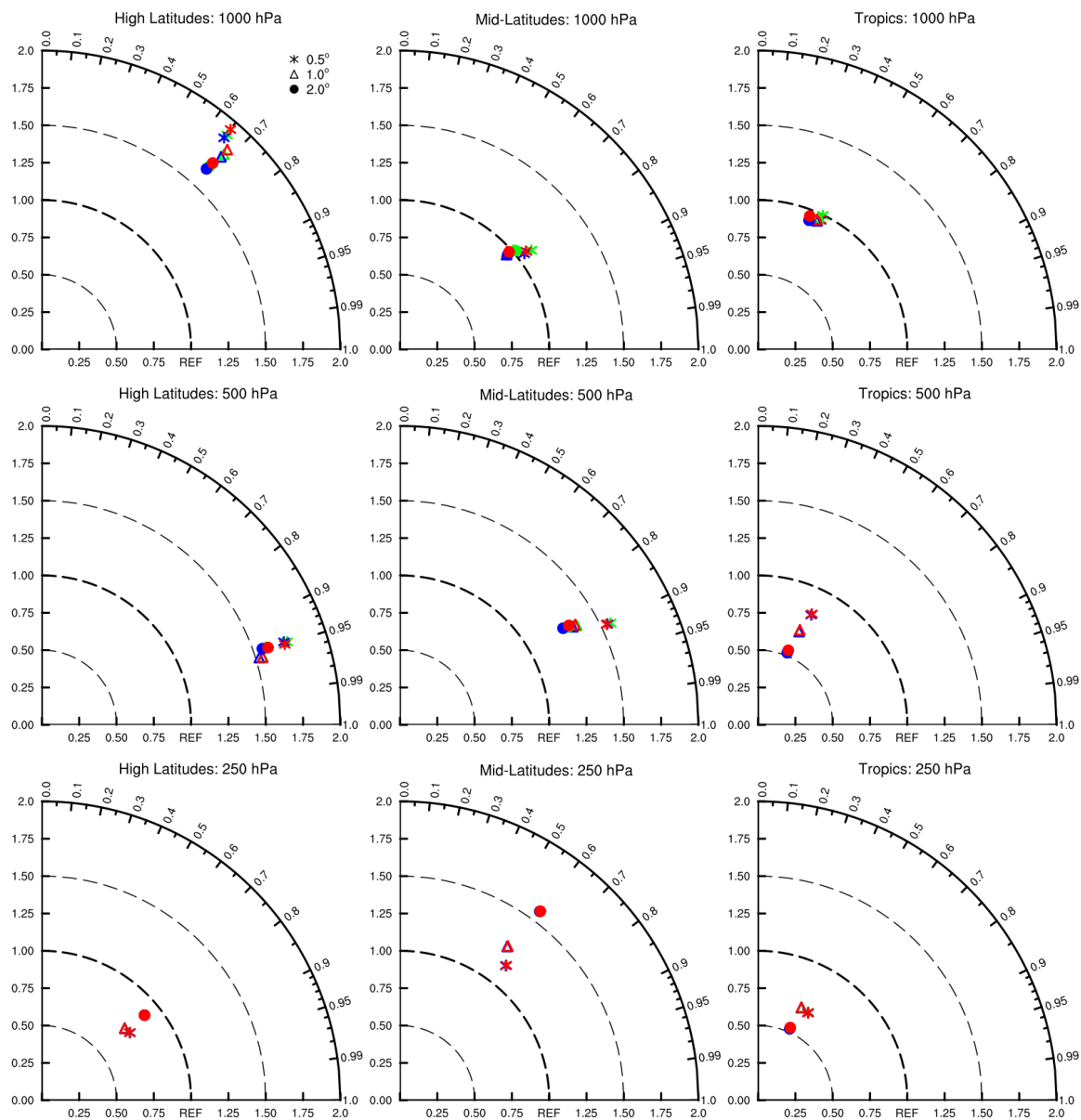


Figure 4.4. Continued.

Figure 4.5. 10-year averaged seasonal median ozone concentration (y-axis, unit: ppb) for regions in the North Mid-Latitudes at 1000 hPa (top row), 500 hPa (middle row), and 250 hPa (bottom row). The x-axis is the season (winter: December-January-February (DJF), spring: March-April-May (MAM), summer: June-July-August (JJA), fall: September-October-November (SON)). Different line patterns mean different resolutions (solid: 0.5°, long dash: 1.0°, short dash: 2.0°) and different colors mean different chemical solvers (blue: ROS-2 solver, green: IMP solver, red: ROS-3 solver). The vertical error bars refer to the one standard deviation.

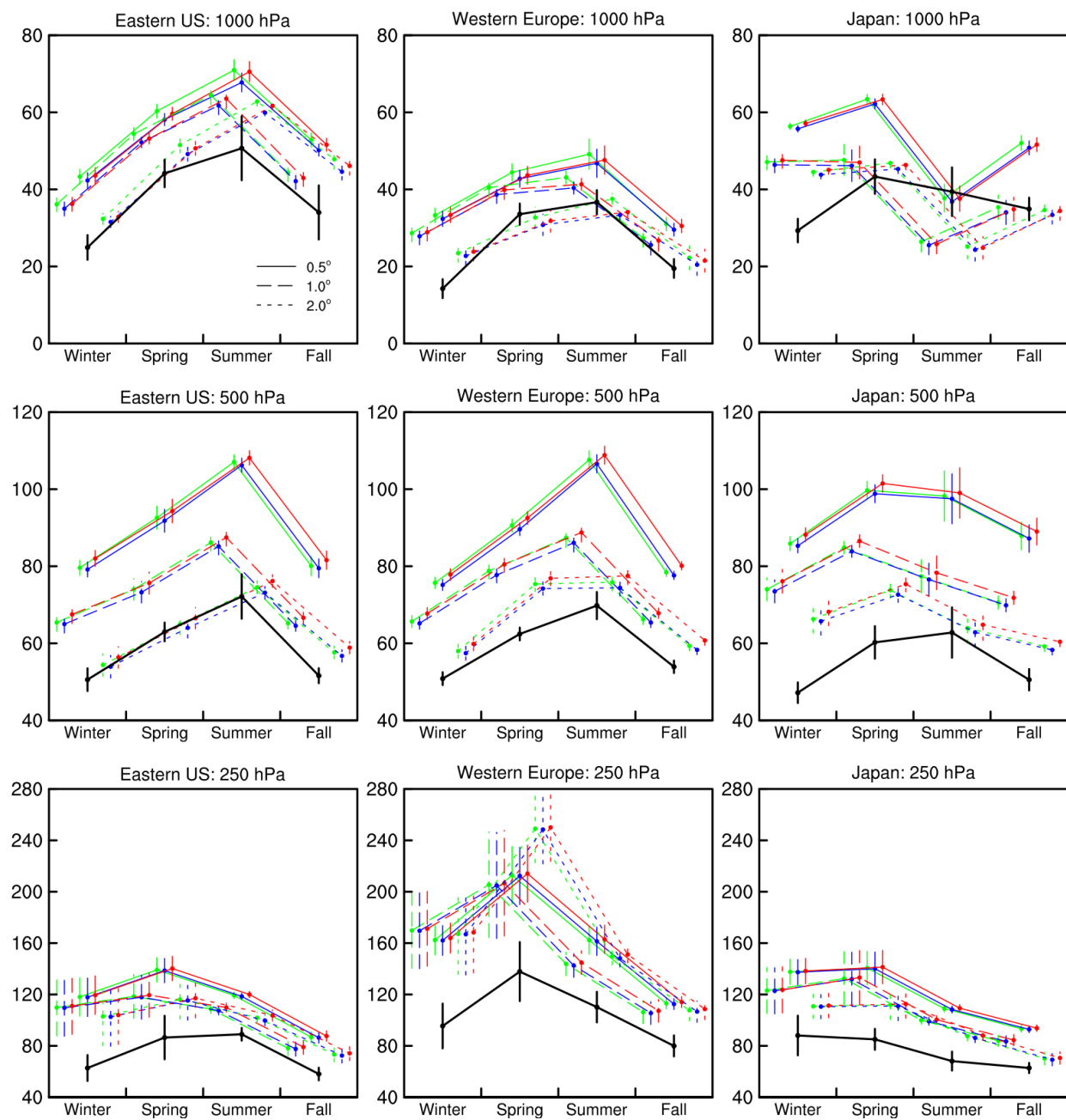


Figure 4.5. Continued.

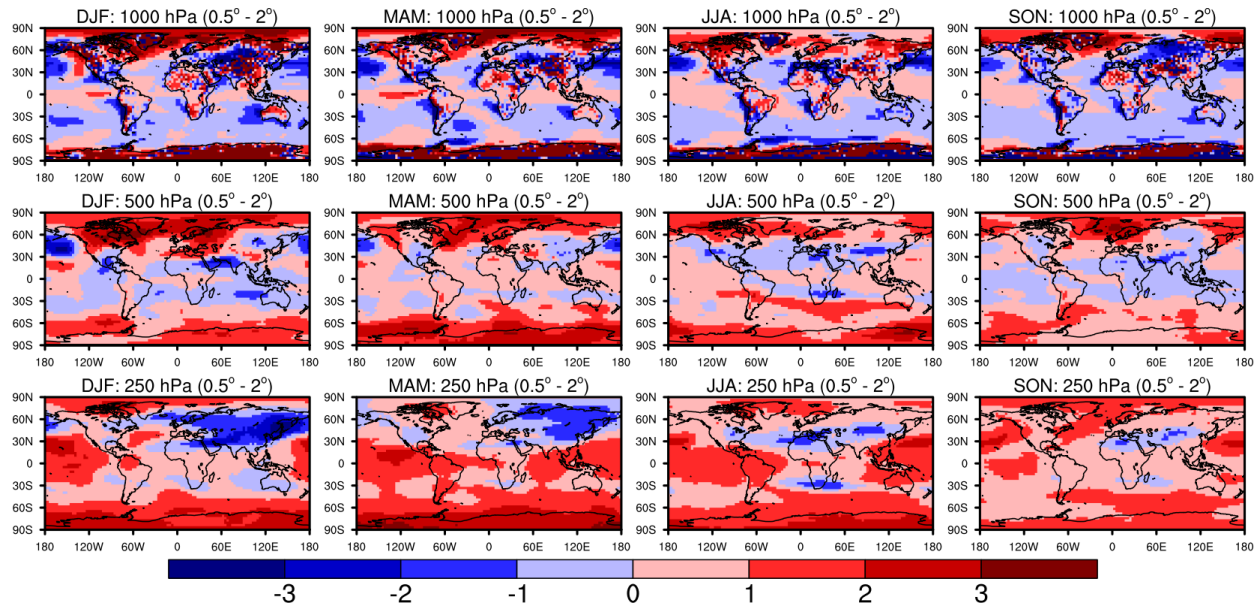


Figure 4.6. Difference of 10-year averaged seasonal mean temperature (unit: K) between the 0.5° (regridded to 2.0°) and 2.0° at 1000 hPa (top row), 500 hPa (middle row), and 250 hPa (bottom row).

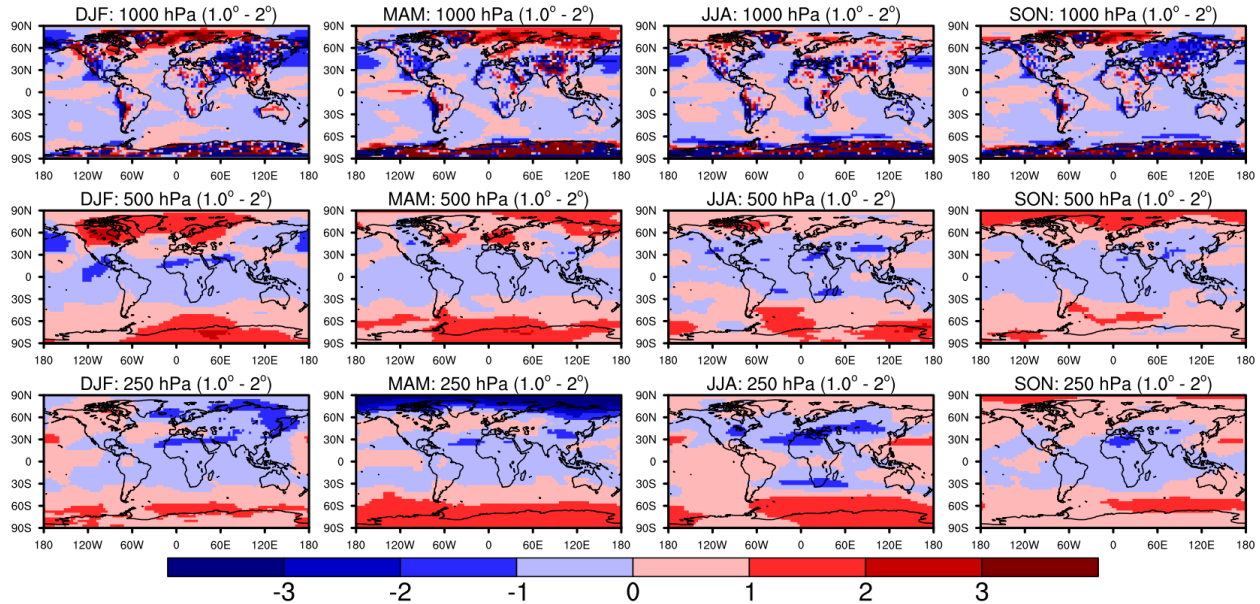


Figure 4.7. The same as Figure 6 but for the difference between the 1.0 degree (regridded to 2.0°) and 2.0°.

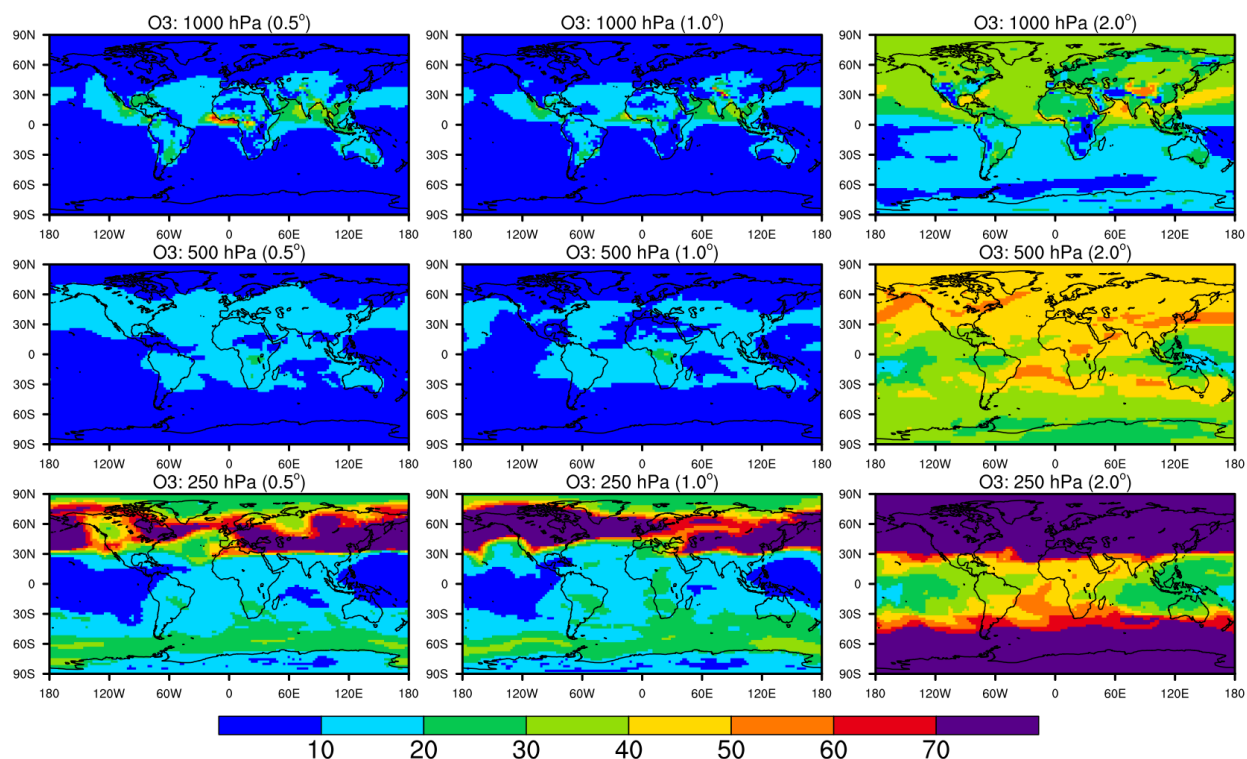


Figure 4.8. Monthly mean ozone concentrations at different altitudes for different resolutions without lightning NO_x emission.

Table 4.2. Summary of summed computational time (units: hours) for each solver (IMP: first-order implicit solver, ROS-2: second-order Rosenbrock solver, ROS-3: third-order Rosenbrock solver) with different horizontal resolutions.

Simulation period	Number of Processors	Solver	Computational time		
			$0.47^\circ \times 0.625^\circ$ (0.5°)	$0.94^\circ \times 1.25^\circ$ (1.0°)	$1.9^\circ \times 2.5^\circ$ (2.0°)
1 month	2048	IMP	192.26	48.31	11.77
		ROS-2	114.82	27.20	6.51
		ROS-3	903.81	207.11	52.15

solver for the simulation with finer resolution. In particular, the ROS-3 solver requires more time step refinements than the ROS-2 solver as a higher order method is generally stricter to the time step size. Despite of the time step refinement, the ROS-2 solver still achieves considerable speed-up compared to the IMP solver using different horizontal resolutions, benefiting from its avoidance of re-evaluation of Jacobian matrix and LU decomposition in the second stage of computation. For the ROS-3 solver, it does not require the re-evaluation of Jacobian matrix and LU decomposition in the second and third stage of computation, either. However, it does request one additional stage to update the right hand side source term and LU solve, which intrinsically makes it slower than the ROS-2 solver. Since the ROS-3 solver also triggers more frequent time step refinements than the ROS-2 solver, it is possible that the ROS-3 solver is slower than the IMP solver as shown here.

4.4 Conclusion

In this study, motivated by our previous success of implementing ROS-2 solver into CAM4-Chem, we further investigate the potential benefit from the higher order Rosenbrock-type solver (i.e., ROS-3). The perturbation study of the chemistry box model first confirms that all the three solvers (IMP, ROS-2 and ROS-3) have resolved their system matrices correctly and agreed very well with each other using 180 seconds or less time step. However, using 1800 seconds time step, the ROS-2 and ROS-3 solvers yield a lower maximum ozone concentration than the IMP solver under the $\pm 90\%$ perturbation scenario of initial species concentrations. On the other hand, using 1800 seconds time step leads to higher maximum, mean and median ozone concentrations for the ROS-2 and ROS-3 solvers under all the nine perturbation scenarios of reaction rates. Perturbing the initial species concentrations causes larger difference of ozone concentration among the three solvers than perturbing the reaction rates, since the system Jacobian matrix and right hand side forcing terms respond more non-linearly to the perturbation of initial species concentrations. Note that under the

perturbation of either initial species concentrations or reaction rates, the ROS-3 solver converges faster to the reference solution than the other two solvers. This suggests that the ROS-3 solver responds better to the stochastic perturbation of initial input for the chemistry box model.

A 10-year free run of climate is then done for each chemical solver. Compared to the ozonesonde climatology data, the ROS-2 solver not only reduces the positive bias of surface ozone concentration compared to the IMP solver, but also the vertical ozone concentration at most regions. In contrast, the ROS-3 solver can only reduce the positive bias of surface ozone concentration at the North Mid-Latitudes and the negative bias of surface ozone concentration at the North Polar and Japan, compared to the IMP solver. The ROS-3 solver generally provides a higher estimation of vertical ozone concentration than the IMP solver, thus leading to a further overestimation at most regions. The effect of grid resolution is also evaluated by examining the 0.5°, 1.0° and 2.0° horizontal resolutions, respectively. It turns out that all the three solvers yield similar seasonal variation and pattern correlation with the ozonesonde climatology data for a given grid resolution, while the 0.5° resolution is able to achieve relatively better pattern correlation but larger seasonal variation than the 1.0° and 2.0° resolutions for a given chemical solver at most regions and altitudes. Furthermore, compared to the IMP solver, the ROS-2 solver achieves $\sim 1.81\times$ speed-up for the 2° resolution but the speed-up decreases slightly for the finer resolution. Nevertheless, the ROS-3 solver is much slower than the IMP solver, especially for the 0.5° resolution. The main reason is the more frequent time step refinements involved by the ROS-3 solver, which also explains its exacerbation of the positive bias of ozone concentration.

Although ROS-3 solver yields better convergence than the ROS-2 and IMP solvers with respect to the stochastic perturbation of initial input in the chemistry box model, the simulation of CAM4-Chem may not benefit too much from the ROS-3 solver for the model prediction and computational efficiency. The ROS-2 solver is again shown as a computationally efficient solver for CAM4-Chem with improved prediction of vertical

ozone concentration. The simulation with finer resolution provides a better pattern correlation, as well as higher seasonal variation. However, the lightning NO_x emission needs to be scaled carefully before comparing the bias between the model and observation data.

References

- Aghedo, A. M., K. W. Bowman, H. M. Worden, S. S. Kulawik, D. T. Shindell, J. F. Lamarque, G. Faluvegi, M. Parrington, D. B. A. Jones and S. Rast (2011), The vertical distribution of ozone instantaneous radiative forcing from satellite and chemistry climate models, *J. Geophys. Res.*, 116, D01305, doi:10.1029/2010JD014243.
- Bacmeister, J. T., M. F. Wehner, R. B. Neale, A. Gettelman, C. Hannay, P. H. Lauritzen, J. M. Caron and J. E. Truesdale (2014), Exploratory High-Resolution Climate Simulations using the Community Atmosphere Model (CAM), *J. Climate*, 27, 3073-3099, doi:10.1175/JCLI-D-13-00387.1.
- Baker, A. H., D. M. Hammerling, M. N. Levy, H. Xu, J. M. Dennis, B. E. Eaton, J. Edwards, C. Hannay, S. A. Mickelson, R. B. Neale, D. Nychka, J. Shollenberger, J. Tribbia, M. Vertenstein and D. Williamson (2015), A new ensemble-based consistency test for the Community Earth System Model (pyCECT v1.0), *Geosci. Model Dev.*, 8(9), 2829-2840, doi:10.5194/gmd-8-2829-2015.
- Bell, M. L., R. Goldberg, C. Hogrefe, P. L. Kinney, K. Knowlton, B. Lynn, J. Rosenthal, C. Rosenzweig, and J. A. Patz (2007), Climate change, ambient ozone, and health in 50 US cities, *Climatic Change*, 82, 61-76, doi:10.1007/s10584-006-9166-7.
- Collins, W. J., R. G. Derwent, C. E. Johnson and D. S. Stevenson (2002), The Oxidation of Organic Compounds in the Troposphere and their Global Warming Potentials, *Climatic Change*, 52(4), 453-479, doi:10.1023/A:1014221225434.
- Dabdub, D. and J. H. Seinfeld (1995), Extrapolation techniques used in the solution of stiff odes associated with chemical kinetics of air quality models, *Atmospheric Environment*, 29(3), 403-410, doi:10.1016/1352-2310(94)00282-P.
- Donner, L. J., B. L. Wyman, R. S. Hemler, L. W. Horowitz, Y. Ming, M. Zhao, J. Golaz, P. Ginoux, S. J. Lin, M. Schwarzkopf, J. Austin, G. Alaka, W. F. Cooke, T. L. Delworth, S. M. Freidenreich, C. T. Gordon, S. M. Griffies, I. M. Held, W. J. Hurlin, S. A. Klein, T. R. Knutson, A. R. Langenhorst, H. Lee, Y. Lin, B. I. Magi, S. L. Malyshev, P. D. Milly, V. Naik, M. J. Nath, R. Pincus, J. J. Ploshay, V. V. Ramaswamy, C. J. Seman, E.

Shevliakova, J. J. Sirutis, W. F. Stern, R. J. Stouffer, R. Wilson, M. Winton, A. T. Wittenberg and F. Zeng (2011), The Dynamical Core, Physical Parameterizations, and Basic Simulation Characteristics of the Atmospheric Component AM3 of the GFDL Global Coupled Model CM3, *J. Climate*, 24, 3484-3519, doi: 10.1175/2011JCLI3955.1.

Eller, P., K. Singh, A. Sandu, K. Bowman, D. K. Henze and M. Lee (2009), Implementation and evaluation of an array of chemical solvers in the Global Chemical Transport Model GEOS-Chem, *Geosci. Model Dev.*, 2(2), 89-96, doi:10.5194/gmd-2-89-2009.

Emmons, L. K., S. Walters, P. G. Hess, J.-F. Lamarque, G. G. Pfister, D. Fillmore, C. Granier, A. Guenther, D. Kinnison, T. Laepple, J. Orlando, X. Tie, G. Tyndall, C. Wiedinmyer, S. L. Baughcum and S. Kloster (2010), Description and evaluation of the Model for Ozone and Related chemical Tracers, version 4 (MOZART-4), *Geosci. Model Dev.*, 3, 43-67, doi:10.5194/gmd-3-43-2010.

Evans, K. J., S. Mahajan, M. Branstetter, J. L. McClean, J. Caron, M. E. Maltrud, J. J. Hack, D. C. Bader, R. Neale and J. K. Leifeld (2014), A spectral transform dynamical core option within the Community Atmosphere Model (CAM4), *J. Adv. Model. Earth Syst.*, 6(3), 902–922, doi:10.1002/2014MS000329.

Eyring, V., J. M. Arblaster, I. Cionni, J. Sedláček, J. Perlwitz, P. J. Young, S. Bekki, D. Bergmann, P. Cameron-Smith, W. J. Collins, G. Faluvegi, K.-D. Gottschaldt, L. W. Horowitz, D. E. Kinnison, J.-F. Lamarque, D. R. Marsh, D. Saint-Martin, D. T. Shindell, K. Sudo, S. Szopa and S. Watanabe (2013), Long-term ozone changes and associated climate impacts in CMIP5 simulations, *J. Geophys. Res. Atmos.*, 118(10), 5029-5060, doi:10.1002/jgrd.50316.

Finlayson-Pitts, B. J. and J. N. Pitts (1993), Atmospheric Chemistry of Tropospheric Ozone Formation: Scientific and Regulatory Implications, *Air & Waste*, 43(8), 1091-1100, doi:10.1080/1073161X.1993.10467187.

Fiore, A. M., D. J. Jacob, I. Bey, R. M. Yantosca, B. D. Field, A. C. Fusco and J. G. Wilkinson (2002), Background ozone over the United States in summer: Origin, trend, and contribution to pollution episodes, *J. Geophys. Res.*, 107, D15, ACH11-1-ACH11-25, doi: 10.1029/2001JD000982.

Fiore, A. M., D. J. Jacob, R. Mathur and R. V. Martin (2003), Application of empirical orthogonal functions to evaluate ozone simulations with regional and global models, *J. Geophys. Res.*, 108(D14), 4431-4445, doi:10.1029/2002JD003151.

Fiore, A. M., J. J. West, L. W. Horowitz, V. Naik and M. D. Schwarzkopf (2008), Characterizing the tropospheric ozone response to methane emission controls and the benefits to climate and air quality, *J. Geophys. Res.*, 113, D08307, doi:10.1029/2007JD009162.

Fiore, A. M., F. J. Dentener, O. Wild, C. Cuvelier, M. G. Schultz, P. Hess, C. Textor, M. Schulz, R. M. Doherty, L. W. Horowitz, I. A. MacKenzie, M. G. Sanderson, D. T. Shindell, D. S. Stevenson, S. Szopa, R. Van Dingenen, G. Zeng, C. Atherton, D. Bergmann, I. Bey, G. Carmichael, W. J. Collins, B. N. Duncan, G. Faluvegi, G. Folberth, M. Gauss, S. Gong, D. Hauglustaine, T. Holloway, I. S. A. Isaksen, D. J. Jacob, J. E. Jonson, J. W. Kaminski, T. J. Keating, A. Lupu, E. Marmer, V. Montanaro, R. J., Park, G. Pitari, K. J. Pringle, J. A. Pyle, S. Schroeder, M. G. Vivanco, P. Wind, G. Wojcik, S. Wu and A. Zuber (2009), Multimodel estimates of intercontinental source-receptor relationships for ozone pollution, *Journal of Geophysical Research*, **114**, D04301, doi: 10.1029/2008JD010816.

Gent, P. R., S. G. Yeager, R. B. Neale, S. Levis and D. A. Bailey (2010), Improvements in a half degree atmosphere/land version of the CCSM, *Clim. Dyn.*, 34, 819-833, doi: 10.1007/s00382-009-0614-8.

Gleckler, P. J., K. E. Taylor and C. Doutriaux (2008), Performance metrics for climate models, *J. Geophys. Res.*, **113**, D06104, doi:10.1029/2007JD008972.

Hairer, E. and G. Wanner (1996), *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, 2nd ed., Berlin, Springer-Verlag, ISBN: 978-3-642-05220-0.

Henze, D. K., A. Hakami and J. H. Seinfeld (2007), Development of the adjoint of GEOS-Chem, *Atmos. Chem. Phys.*, 7(9), 2413-2433, doi:10.5194/acp-7-2413-2007.

Hollaway, M. J., S. R. Arnold, A. J. Challinor and L. D. Emberson (2012), Intercontinental trans-boundary contributions to ozone-induced crop yield losses in the Northern Hemisphere, *Biogeosciences*, **9**(1), 271-292, doi: 10.5194/bg-9-271-2012.

Jonson, J. E., A. Stohl, A. M. Fiore, P. Hess, S. Szopa, O. Wild, G. Zeng, F. J. Dentener, A. Lupu, M. G. Schultz, B. N. Duncan, K. Sudo, P. Wind, M. Schulz, E. Marmer, C. Cuvelier, T. Keating, A. Zuber, A. Valdebenito, V. Dorokhov, H. De Backer, J. Davies, G. H. Chen, B. Johnson, D. W. Tarasick, R. Stübi, M. J. Newchurch, P. von der Gathen, W. Steinbrecht and H. Claude (2010), A multi-model analysis of vertical ozone profiles, *Atmos. Chem. Phys.*, 10(12), 5759-5783, doi:10.5194/acp-10-5759-2010.

Kay, J. E., C. C. Deser, A. A. Phillips, A. A. Mai, C. C. Hannay, G. G. Strand, J. M. Arblaster, S. C. Bates, G. G. Danabasoglu, J. J. Edwards, M. M. Holland, P. P. Kushner, J.-F. Lamarque, D. D. Lawrence, K. K. Lindsay, A. A. Middleton, E. E. Munoz, R. R. Neale, K. K. Oleson, L. L. Polvani and M. M. Vertenstein (2015), The Community Earth System Model (CESM) Large Ensemble Project: A Community Resource for Studying Climate Change in the Presence of Internal Climate Variability. *Bull. Amer. Meteor. Soc.*, 96, 1333-1349, doi: 10.1175/BAMS-D-13-00255.1.

Kim, H. C., P. Lee, F. Ngan, Y. Tang, H. L. Yoo and L. Pan (2015), Evaluation of modeled surface ozone biases as a function of cloud cover fraction, *Geosci. Model Dev.*, 8(9), 2959-2965, doi: 10.5194/gmd-8-2959-2015.

Kinnison, D. E., G. P. Brasseur, S. Walters, R. R. Garcia, D. R. Marsh, F. Sassi, V. L. Harvey, C. E. Randall, L. Emmons, J.-F. Lamarque, P. Hess, J. J. Orlando, X. X. Tie, W. Randel, L. L. Pan, A. Gettelman, C. Granier, T. Diehl, U. Niemeier and A. J. Simmons (2007), Sensitivity of chemical tracers to meteorological parameters in the MOZART-3 chemical transport model, *J. Geophys. Res.*, **112**, D20302, doi: 10.1029/2006JD007879.

Kumar, N., M. T. Odman and A. G. Russell (1994), Multiscale air quality modeling: Application to southern California, *J. Geophys. Res.*, 99(D3), 5385-5397, doi: 10.1029/93JD03197.

Lamarque, J.-F., P. Hess, L. Emmons, L. Buja, W. Washington and C. Granier (2005), Tropospheric ozone evolution between 1890 and 1990, *J. Geophys. Res.*, 110, D08304, doi:10.1029/2004JD005537.

Lamarque, J.-F., G. P. Kyle, M. Meinshausen, K. Riahi, S. J. Smith, D. P. van Vuuren, A. Conley and F. Vitt (2011), Global and regional evolution of short-lived radiatively-active gases and aerosols in the Representative Concentration Pathways, *Climatic Change*, 109(1), 191-212, doi:10.1007/s10584-011-0155-0.

Lamarque, J.-F., L. K. Emmons, P. G. Hess, D. E. Kinnison, S. Tilmes, F. Vitt, C. L. Heald, E. A. Holland, P. H. Lauritzen, J. Neu, J. J. Orlando, P. J. Rasch and G. K. Tyndall (2012), CAM-chem: description and evaluation of interactive atmospheric chemistry in the Community Earth System Model, *Geosci. Model Dev.*, 5(2), 369-411, doi: 10.5194/gmd-5-369-2012.

Lang, J. and J. Verwer (2001), ROS3P—An Accurate Third-Order Rosenbrock Solver Designed for Parabolic Problems, *BIT Numerical Mathematics*, 41(4), 731-738, doi:10.1023/A:1021900219772.

Liang, J. and M. Z. Jacobson (2000), Effects of subgrid segregation on ozone production efficiency in a chemical model, *Atmos. Environ.*, 34(18), 2975-2982, doi: 10.1016/S1352-2310(99)00520-8.

Linford, J. C., J. Michalakes, M. Vachharajani and A. Sandu (2009), Multi-core acceleration of chemical kinetics for simulation and prediction, *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, SC '09*, Association for Computing Machinery, New York, NY, 7, 1-11, doi: 10.1145/1654059.1654067.

Luecken, D. J., S. Phillips, G. Sarwar and C. Jang (2008), Effects of using the CB05 vs. SAPRC99 vs. CB4 chemical mechanism on model predictions: Ozone and gas-phase photochemical precursor concentrations, *Atmospheric Environment*, 42(23), 5805-5820, doi: 10.1016/j.atmosenv.2007.08.056.

Martin, R. V., D. J. Jacob, R. M. Yantosca, M. Chin and P. Ginoux (2003), Global and regional decreases in tropospheric oxidants from photochemical effects of aerosols, *J. Geophys. Res.*, 108, D34097, doi: 10.1029/2002JD002622.

Monks, P. S. (2005), Gas-phase radical chemistry in the troposphere, *Chem. Soc. Rev.*, 34(5), 376-395, doi: 10.1039/B307982C.

Monks, P. S., A. T. Archibald, A. Colette, O. Cooper, M. Coyle, R. Derwent, D. Fowler, C. Granier, K. S. Law, G. E. Mills, D. S. Stevenson, O. Tarasova, V. Thouret, E. von Schneidemesser, R. Sommariva, O. Wild and M. L. Williams (2015), Tropospheric ozone and its precursors from the urban to the global scale from air quality to short-lived climate forcer, *Atmos. Chem. Phys.*, 15(15), 8889-8973, doi:10.5194/acp-15-8889-2015.

Murazaki, K. and P. Hess (2006), How does climate change contribute to surface ozone change over the United States? *J. Geophys. Res.*, 111, D05301, doi: 10.1029/2005JD005873.

Neale, R. B., J. Richter, S. Park, P. H. Lauritzen, S. J. Vavrus, P. J. Rasch and M. H. Zhang (2013), The Mean Climate of the Community Atmosphere Model (CAM4) in Forced SST and Fully Coupled Experiments, *Journal of Climate*, **26**, 5150-5168, doi: 10.1175/JCLI-D-12-00236.1.

Nørsett, S. P. and A. Wolfbrandt (1979), Order conditions for Rosenbrock type methods, *Numer. Math.*, 32(1), 1-15, doi:10.1007/BF01397646.

Pan, L., D. Tong, P. Lee, H.-C. Kim and T. F. Chai (2014), Assessment of NO_x and O₃ forecasting performances in the U.S. National Air Quality Forecasting Capability before and after the 2012 major emissions updates, *Atmospheric Environment*, 95, 610-619, doi: 10.1016/j.atmosenv.2014.06.020.

Park, R. J., K. E. Pickering, D. J. Allen, G. L. Stenchikov and M. S. Fox-Rabinovitz (2004), Global simulation of tropospheric ozone using the University of Maryland Chemical Transport Model (UMD-CTM): 2. Regional transport and chemistry over the central United States using a stretched grid, *J. Geophys. Res.*, 109, D09303, doi:10.1029/2003JD004269.

Rang, J. and L. Angermann (2005), New Rosenbrock W-Methods of Order 3 for Partial Differential Algebraic Equations of Index 1, *BIT Numerical Mathematics*, 45(4), 761-787, doi:10.1007/s10543-005-0035-y.

Rosinski, J. M. and D. L. Williamson (1997), The accumulation of rounding errors and port validation for global atmospheric models, *SIAM J. SCI. COMPUT.*, 18(2), 552-564. doi: 10.1137/S1064827594275534.

Sandu A. and P. Miehe (2010), Forward, tangent linear, and adjoint Runge–Kutta methods for stiff chemical kinetic simulations, *International Journal of Computer Mathematics*, 87(11), 2458-2479, doi:10.1080/00207160802676562.

Sarwar, G., D. Luecken, G. Yarwood, G. Z. Whitten, and W. L. Carter (2008), Impact of an Updated Carbon Bond Mechanism on Predictions from the CMAQ Modeling System: Preliminary Assessment, *J. Appl. Meteor. Climatol.*, 47, 3-14, doi:10.1175/2007JAMC1393.1.

Sarwar, G., J. Godowitch, B. H. Henderson, K. Fahey, G. Pouliot, W. T. Hutzell, R. Mathur, D. Kang, W. S. Goliff and W. R. Stockwell (2013), A comparison of atmospheric composition using the Carbon Bond and Regional Atmospheric Chemistry Mechanisms, *Atmos. Chem. Phys.*, **13**, 9695-9712, doi:10.5194/acp-13-9695-2013.

Shindell, D. T., O. Pechony, A. Voulgarakis, G. Faluvegi, L. Nazarenko, J.-F. Lamarque, K. Bowman, G. Milly, B. Kovari, R. Ruedy and G. A. Schmidt (2013), Interactive ozone and methane chemistry in GISS-E2 historical and future climate simulations, *Atmos. Chem. Phys.*, 13(5), 2653-2689, doi:10.5194/acp-13-2653-2013.

Shintani, H. (1982), Modified Rosenbrock methods with approximate Jacobian matrices, *Hiroshima Math. J.*, 12(3), 559-568, <http://projecteuclid.org/euclid.hmj/1206133647>.

Sillman, S., J. A. Logan and S. C. Wofsy (1990), A regional scale model for ozone in the United States with subgrid representation of urban and power plant plumes, *J. Geophys. Res.*, 95(D5), 5731-5748, doi: 10.1029/JD095iD05p05731.

Sillman, S. (1995), The use of NO_y , H_2O_2 , and HNO_3 as indicators for ozone- NO_x -hydrocarbon sensitivity in urban locations, *J. Geophys. Res.*, 100(D7), 14175-14188, doi:10.1029/94JD02953.

Sitch, S., P. M. Cox, W. J. Collins and C. Huntingford (2007), Indirect radiative forcing of climate change through ozone effects on the land-carbon sink, *Nature*, 448(7), 791-794, doi:10.1038/nature06059.

Small, R. J., J. Bacmeister, D. Bailey, A. Baker, S. Bishop, F. Bryan, J. Caron, J. Dennis, P. Gent, H.-M. Hsu, M. Jochum, D. Lawrence, E. Muñoz, P. diNezio, T. Scheitlin, R. Tomas, J. Tribbia, Y.-H. Tseng and M. Vertenstein (2014), A new synoptic

scale resolving global climate simulation using the Community Earth System Model, J. Adv. Model. Earth Syst., 6(4), 1065-1094, doi:10.1002/2014MS000363.

Sriver, R. L., C. E. Forest and K. Keller (2015), Effects of initial conditions uncertainty on regional climate variability: An analysis using a low-resolution CESM ensemble, Geophys. Res. Lett., 42(13), 5468-5476, doi:10.1002/2015GL064546.

Steihaug, T. and A. Wolfbrandt (1979), An Attempt to Avoid Exact Jacobian and Nonlinear Equations in the Numerical Solution of Stiff Differential Equations, Mathematics of Computation, 33(146), 521-534, doi:10.2307/2006293.

Sun, J., J. S. Fu, K. Huang and Y. Gao (2015), Estimation of future PM_{2.5}- and ozone-related mortality over the continental United States in a changing climate: An application of high-resolution dynamical downscaling technique, *Journal of the Air & Waste Management Association*, **65**(5), 611-623, doi: 10.1080/10962247.2015.1033068.

Sun, J., J. S. Fu, J. Drake, J.-F. Lamarque, S. Tilmes, and F. Vitt (2017), Improvement of the prediction of surface ozone concentration over conterminous U.S. by a computationally efficient second-order Rosenbrock solver in CAM4-Chem, J. Adv. Model. Earth Syst., 9, doi:10.1002/2016MS000863.

Stevenson, D. S., F. J. Dentener, M. G. Schultz, K. Ellingsen, T. P. C. van Noije, O. Wild, G. Zeng, M. Amann, C. S. Atherton, N. Bell, D. J. Bergmann, I. Bey, T. Butler, J. Cofala, W. J. Collins, R. G. Derwent, R. M. Doherty, J. Drevet, H. J. Eskes, A. M. Fiore, M. Gauss, D. A. Hauglustaine, L. W. Horowitz, I. S. A. Isaksen, M. C. Krol, J.-F. Lamarque, M. G. Lawrence, V. Montanaro, J.-F. Müller, G. Pitari, M. J. Prather, J. A. Pyle, S. Rast, J. M. Rodriguez, M. G. Sanderson, N. H. Savage, D. T. Shindell, S. E. Strahan, K. Sudo and S. Szopa (2006), Multimodel ensemble simulations of present-day and near-future tropospheric ozone, *J. Geophys. Res.*, **111**, D08301, doi: 10.1029/2005JD006338.

Stevenson, D. S., P. J. Young, V. Naik, J.-F. Lamarque, D. T. Shindell, A. Voulgarakis, R. B. Skeie, S. B. Dalsoren, G. Myhre, T. K. Berntsen, G. A. Folberth, S. T. Rumbold, W. J. Collins, I. A. MacKenzie, R. M. Doherty, G. Zeng, T. P. C. van Noije, A. Strunk, D. Bergmann, P. Cameron-Smith, D. A. Plummer, S. A. Strode, L. Horowitz, Y. H. Lee, S. Szopa, K. Sudo, T. Nagashima, B. Josse, I. Cionni, M. Righi, V. Eyring, A. Conley, K.

W. Bowman, O. Wild and A. Archibald (2013), Tropospheric ozone changes, radiative forcing and attribution to emissions in the Atmospheric Chemistry and Climate Model Intercomparison Project (ACCMIP), *Atmos. Chem. Phys.*, 13(6), 3063-3085, doi:10.5194/acp-13-3063-2013.

Taylor, K. E (2001), Summarizing multiple aspects of model performance in a single diagram, *J. Geophys. Res.*, **106**(D7), 7183-7192, doi:10.1029/2000JD900719.

Tong, D. Q., R. Mathur, D. Kang, S. Yu, K. L. Schere and G. Pouliot (2009), Vegetation exposure to ozone over the continental United States: Assessment of exposure indices by the Eta-CMAQ air quality forecast model, *Atmospheric Environment*, 43(3), 724-733, doi: 10.1016/j.atmosenv.2008.09.084.

Travis, K. R., D. J. Jacob, J. A. Fisher, P. S. Kim, E. A. Marais, L. Zhu, K. Yu, C. C. Miller, R. M. Yantosca, M. P. Sulprizio, A. M. Thompson, P. O. Wennberg, J. D. Crounse, J. M. St. Clair, R. C. Cohen, J. L. Laughner, J. E. Dibb, S. R. Hall, K. Ullmann, G. M. Wolfe, I. B. Pollack, J. Peischl, J. A. Neuman and X. Zhou (2016), Why do models overestimate surface ozone in the Southeast United States?, *Atmos. Chem. Phys.*, 16(21), 13561-13577, doi:10.5194/acp-16-13561-2016.

Tie, X., G. Brasseur and Z. Ying (2010), Impact of model resolution on chemical ozone formation in Mexico City: application of the WRF-Chem model, *Atmos. Chem. Phys.*, 10(18), 8983-8995, doi:10.5194/acp-10-8983-2010.

Tilmes, S., J.-F. Lamarque, L. K. Emmons, A. Conley, M. G. Schultz, M. Saunois, V. Thouret, A. M. Thompson, S. J. Oltmans, B. Johnson and D. Tarasick (2012), Technical Note: Ozone sonde climatology between 1995 and 2011: description, evaluation and applications, *Atmos. Chem. Phys.*, 12(16), 7475-7497, doi:10.5194/acp-12-7475-2012.

Val Martin, M., C. L. Heald and S. R. Arnold (2014), Coupling dry deposition to vegetation phenology in the Community Earth System Model: Implications for the simulation of surface O₃, *Geophys. Res. Lett.*, **41**(8), 2988-2996, doi: 10.1002/2014GL059651.

Val Martin, M., C. L. Heald, J.-F. Lamarque, S. Tilmes, L. K. Emmons and B. A. Schichtel (2015), How emissions, climate, and land use change will impact mid-century

air quality over the United States: a focus on effects at national parks, *Atmos. Chem. Phys.*, **15**, 2805-2823, doi:10.5194/acp-15-2805-2015.

Verwer, J. G. and D. Simpson (1995), Explicit methods for stiff ODEs from atmospheric chemistry, *Applied Numerical Mathematics*, 18(1-3), 413-430, doi: 10.1016/0168-9274(95)00068-6.

Verwer, J. G., E. J. Spee, J. G. Blom and W. Hundsdorfer (1999), A second-order rosenbrock method applied to photochemical dispersion problems, *SIAM J. Sci. Comput.*, **20**(4), 1456–1480, doi:10.1137/S1064827597326651.

Wild, O. and M. J. Prather (2006), Global tropospheric ozone modeling: Quantifying errors due to grid resolution, *J. Geophys. Res.*, 111, D11305, doi:10.1029/2005JD006605.

Wilkinson, S., G. Mills, R. Illidge and W. J. Davies (2012), How is ozone pollution reducing our food supply? *Journal of Experimental Botany*, 63(2), 527-536, doi: 10.1093/jxb/err317.

Yan, Y., J. Lin, J. Chen and L. Hu (2016), Improved simulation of tropospheric ozone by a global-multi-regional two-way coupling model system, *Atmos. Chem. Phys.*, 16(4), 2381-2400, doi:10.5194/acp-16-2381-2016.

Young, P. J., A. T. Archibald, K. W. Bowman, J.-F. Lamarque, V. Naik, D. S. Stevenson, S. Tilmes, A. Voulgarakis, O. Wild, D. Bergmann, P. Cameron-Smith, I. Cionni, W. J. Collins, S. B. Dalsøren, R. M. Doherty, V. Eyring, G. Faluvegi, L. W. Horowitz, B. Josse, Y. H. Lee, I. A. MacKenzie, T. Nagashima, D. A. Plummer, M. Righi, S. T. Rumbold, R. B. Skeie, D. T. Shindell, S. A. Strode, K. Sudo, S. Szopa and G. Zeng (2013), Pre-industrial to end 21st century projections of tropospheric ozone from the Atmospheric Chemistry and Climate Model Intercomparison Project (ACCMIP), *Atmos. Chem. Phys.*, **13**, 2063-2090, doi:10.5194/acp-13-2063-2013.

Zedan, H. (1990), Avoiding the exactness of the Jacobian matrix in Rosenbrock formulae, *Computers & Mathematics with Applications*, 19(2), 83-89, doi: 10.1016/0898-1221(90)90011-8.

Zhang, Y. Q., O. R. Cooper, A. Gaudel, A. M. Thompson, P. Nédélec, S.-Y. Ogino and J. J. West (2016), Tropospheric ozone change from 1980 to 2010 dominated by

equatorward redistribution of emissions, *Nature Geoscience*, 9, 875-879,
doi:10.1038/ngeo2827.

CHAPTER V CONCLUSION

5.1 Summary of this study

This study aims at investigating the overestimation of ozone concentration from the global chemistry-climate model (CAM4-Chem). Instead of tuning the emission inventory or meteorological field, we mainly focus on the numerical property of the chemical solver itself. To do so, the high-order Rosenbrock-type solver is examined within a chemistry box model for the analysis of convergence and implemented into CAM4-Chem for the evaluation of model prediction. The chemistry box model is isolated from the CAM4-Chem and thus we could investigate the impact of different chemical solvers on the chemistry module alone. The analysis convergence, as well as the response to the stochastic perturbation of initial input, provides a detailed insight into the numerical behavior of different chemical solvers. This helps to explain the difference in the real simulation when different chemical solvers are implemented into CAM4-Chem. The simulated data are compared to the observation of surface ozone concentration and the ozone climatology data for vertical profile. This gives a robust evaluation of the model capability to simulate the ozone concentration. Limited by the computation, a 10-year simulation is done for the present year. The impact of different chemical solvers on chemistry under the projection of future scenarios remains unexplored. It may also be necessary to investigate the feedback of chemistry to the climate through the direct and indirect radiative effects.

In addition, due to the growing complexity of chemistry-climate model, the computational burden increases rapidly and further development is prohibited by the limitation of computational resources. Therefore, finding a computationally efficient solver is important to advance the model development and conduct more sensitivity analysis. This is also a motivation of choosing Rosenbrock-type solvers in this study, since it does not require iteration like the current IMP solver during each time

integration. Moreover, by choosing the parameters wisely, the Rosenbrock-type solver can achieve L-stability for a linear system and avoid the re-evaluation of Jacobian matrix and LU decomposition during different stage of computation. On the other hand, the next-generation supercomputer is equipped with more GPUs, which is much powerful than CPU for computation. Embracing the strong power of GPU is critical to further speed up the computational performance of global chemistry-climate model. We focus on the chemistry box model and optimize it with several strategies on GPU. Evident speed-up is gained compared to the CPU and we believe it will contribute significantly to the development of next-generation model.

5.2 Summary from CHAPTER II

By implementing the ROS-2 solver in CAM4-Chem and comparing to the IMP solver, the ROS-2 solver provides widely lower annual mean surface ozone concentration using the 1800 seconds time step. During the JJA, the ROS-2 solver similarly shows a lower estimation of surface ozone concentration over the CONUS and the absolute difference is much larger. This implies that the ROS-2 solver can help improve the performance of surface ozone concentration over the CONUS, especially during the summer season when the photochemical reactions are the most active. In addition, refining the time step size from 1800 seconds to 180 seconds, the IMP solver benefits slightly with less visible improvement of surface ozone concentration. This highlights the importance to implement other solvers like ROS-2 solver in CAM4-Chem to improve the model prediction. Nevertheless, the outperformance of the ROS-2 solver may not come from its second order accuracy with respect to time scale, but from its suitability for solving stiff problems.

The analysis of computational cost for the IMP solver and ROS-2 solver reveals that the ROS-2 solver takes about 47% less computational time than the IMP solver when using the same time step size. This $\sim 2.0\times$ speedup is due to the fact that the ROS-2 solver

uses the same structure of Jacobian matrix and LU factorization during the two-stage calculations, which could consume lots of computational time for the chemistry update. Refining the time step size by a factor of 10 won't lead to ten times slower, mainly attributed by the implementation of adaptive time step method.

5.3 Summary from CHAPTER III

The strong computational power and fast memory access of the GPU in modern supercomputer architectures provides a good opportunity to accelerate the computation of chemistry from the global chemistry-climate model. To explore the benefit from GPU, we port the ROS-2 solver in the chemistry module of CAM4-Chem to the GPU and compare its computational performance to the CPU. The analysis shows that different compilers and memory layouts play a critical role in the final computational performance on the CPU. The Intel compiler with BIML achieves the lowest computational time. The GNU and PGI compilers can benefit more from SML. Formation of the Jacobian matrix and LU (both factorization and solve) are the most time-consuming parts of the chemistry update for most configurations.

In contrast, using the FIML yields better performance on the GPU. The most optimized configuration on the GPU is achieved by using CUDA streams, with declaring the mapping array on the constant memory and all the other arrays by a contiguous allocation. The GPU gains as much as $\sim 11.7\times$ speed-up of the computational time compared to the CPU. Even when the data movement between CPU and GPU is considered, the speed-up can be as high as $3.82\times$. This significant speed-up can provide useful information to the software engineering and model developers for the optimization of the next-generation global chemistry-climate model.

5.4 Summary from CHAPTER IV

Motivated by the previous success of implementing ROS-2 solver into CAM4-Chem, we further implement the ROS-3 solver and investigate its potential benefit. By perturbing the initial input of the chemistry box model, it turns out that the IMP solver, ROS-2 solver and ROS-3 solver have all resolved their system matrices correctly and agreed very well with each other using 180 seconds or less time step. Nevertheless, using 1800 seconds time step, the ROS-2 and ROS-3 solvers provide a lower estimation of maximum ozone concentration than the IMP solver under the $\pm 90\%$ perturbation scenario of initial species concentrations. On the other hand, using 1800 seconds time step causes higher estimation of maximum, mean and median ozone concentrations for the ROS-2 and ROS-3 solvers under all the nine perturbation scenarios of reaction rates. We find that the ROS-3 solver can respond better than the other two chemical solvers to the stochastic perturbation of initial input for the chemistry box model.

By doing a 10-year free-run of CAM4-Chem and comparing to the ozonesonde climatology data, the ROS-2 solver is shown to reduce the positive bias of ozone concentration predicted by the IMP solver at most regions. However, the ROS-3 solver can only reduce the positive bias of surface ozone concentration at the North Mid-Latitudes and the negative bias of surface ozone concentration at the North Polar and Japan, compared to the IMP solver. Considering the effect of horizontal grid resolutions, all the three solvers yield similar seasonal variation and pattern correlation with the ozonesonde climatology data for a given grid resolution. Nevertheless, the 0.5° resolution subjects to relatively better pattern correlation but larger seasonal variation than the 1.0° and 2.0° resolutions for a given chemical solver at most regions and altitudes.

Furthermore, the ROS-2 solver gains $\sim 1.81\times$ speed-up for the 2° resolution compared to the IMP solver. This speed-up is reduced slightly for the finer resolution. On the other

hand, the ROS-3 solver costs much more computational time than the IMP solver for different grid resolutions. This is mainly due to the more frequent time step refinements triggered by the ROS-3 solver, which could explain its further overestimation of the ozone concentration compared to the IMP solver as well.

CHAPTER VI FUTURE WORK

In this study, we mainly focus on the TROP_MOZART mechanism. However, there are other chemistry mechanisms in CAM-Chem that are designed for stratosphere, aerosols or background tropospheric ozone chemistry. It is worth investigating the potential benefit of Rosenbrock-type solvers for those mechanisms. Evaluating the model prediction and computational efficiency, this could give a more comprehensive analysis of different chemical solvers with respect to the different mechanisms.

Secondly, though the GPU has achieved significant speed-up compared to the CPU for the chemistry box model, careful design is required to balance the workload between CPU and GPU when implementing the GPU version of chemical solver in CAM4-Chem. The ideal strategy is to assign larger loop of iterations to one single CPU core and let it call the GPU to do the computation, while the other CPU cores in the same node are assigned with smaller loop of iterations. However, modification of the main data structure of CAM4-Chem is necessary to reschedule the workload and overlap the computation.

Last, the global flash frequency of the finer grid resolution is much higher than that of the coarse grid resolution, which leads to a much higher lightning NO_x emission for the finer resolution. This further causes a strong overestimation of ozone concentration for the finer resolution, which could be misleading. A scaled lightning NO_x emission is required for the finer resolution before we can conclude whether the atmospheric chemistry can benefit from the high-resolution simulation.

VITA

Jian Sun was born in Hangzhou, Zhejiang Province, China. In 2012, Jian graduated as a Bachelor of Engineering from Zhejiang University, Zhejiang, China. The same year, he entered the University of Tennessee at Knoxville to pursue the Ph.D degree at the Department of Civil Engineering. In May 2018, he finished the Ph.D program with an interdisciplinary graduate minor in computational science.