https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

Enhancing the advection module performance in the EPICC-Model V1.0 via GPU-HADVPPM4HIP V1.0 coupling and GPU-optimized strategies

Kai Cao[1], Qizhong Wu[2], Xiao Tang[1,3], Jinxi Li[1], Xueshun Chen[1,3], Huansheng Chen[1], Wending Wang[1], Huangjian Wu[1], Lei Kong[1], Jie Li[1,3], Jiang Zhu[1,3], and Zifa Wang[1,3]

[1]State Key Laboratory of Atmospheric Environment and Extreme Meteorology, Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100029, China

[2]College of Global Change and Earth System Science, Faculty of Geographical Science, Beijing Normal University, Beijing 100875, China

[3]College of Earth and Planetary Sciences, University of Chinese Academy of Sciences, Beijing, 100049, China

Correspondence to: Qizhong Wu (wqizhong@bnu.edu.cn); Xiao Tang (tangxiao@mail.iap.ac.cn)

Abstract

The rapid development of Graphics Processing Units (GPUs) has established new computational paradigms for enhancing air quality modeling efficiency. In this study, the heterogeneous-compute interface for portability (HIP) was implemented to parallel computing of the piecewise parabolic method (PPM) advection solver (HADVPPM) on China's domestic GPU-like accelerators (GPU-HADVPPM4HIP V1.0). Computational performance was enhanced through three strategic optimizations: reducing the central processing unit (CPU) and GPU (CPU-GPU) data transfer frequency, thread-block coordinated indexing, and the Message Passing Interface and HIP ("MPI+HIP") hybrid parallelization across heterogeneous computing clusters. Following validation of the GPU-HADVPPM4HIP V1.0 program's offline computational consistency and the pollutant simulation performance of the Emission and atmospheric Processes Integrated and Coupled Community version 1.0 (EPICC-Model V1.0) on the Earth System Numerical Simulation Facility (EarthLab), comprehensive performance testing was conducted. Offline benchmark results

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

demonstrated that GPU-HADVPPM4HIP V1.0 achieved a maximum speedup of 556.5x on a domestic GPU-like accelerator with compiler optimization. Integration of GPU-HADVPPM4HIP V1.0 into EPICC-Model V1.0, combined with optimized CPU-GPU communication frequency and thread-block coordinated indexing strategies, yielded model-level computational efficiency improvements of 17.0x and 1.5x, respectively. At the module level, GPU-HADVPPM4HIP V1.0 exhibited a 39.3% computational efficiency gain when accounting for CPU-GPU data transfer overhead, which escalated to 20.5x acceleration when excluding communication costs. This coupling establishes a foundational framework for adapting air quality models to China's domestic GPU-like architectures and identifies critical optimization pathways. Moreover, the methodology provides essential technical support for achieving full-model GPU implementation of the EPICC-Model, addressing both current computational constraints and future demands for high-resolution air quality simulations.

1. Introduction

Air pollution, a source of fine particulate matter in both urban and rural areas, is associated with an elevated risk of strokes, heart diseases, lung cancer, and acute and chronic respiratory diseases (Atkinson et al., 2010; Kim et al., 2015; Liu et al., 2016; Milton and White, 2020). The air quality forecasting system centered on the air quality model plays a critical role in the timely dissemination of forecasting alerts and early warning information to the public. The accuracy of air quality forecasting is jointly constrained by the spatial resolution of input datasets, including emission inventories, terrain, and meteorological parameters (Gupta et al., 2015; Georgiou et al., 2022). High-resolution model configurations have demonstrated improvements in the accuracy of air quality forecasting (Georgiou et al., 2018; Podrascanin et al., 2019; Adani et al., 2022; Gao and Zhou, 2024). However, current operational forecasting systems predominantly employ horizontal resolutions ranging from several to tens of kilometers (Wu et al., 2014; Guevara et al., 2021; Tang et al., 2022; Gao and Zhou, 2024), which inadequately address the requirements for urban-scale high-resolution forecasting and precision management.

Computational demands emerge as a critical limiting factor for high-resolution air quality modeling. On the one hand, doubling the horizontal resolution quadruples the number of

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

computational grids. On the other hand, maintaining numerical integration stability necessitates proportional reduction in temporal integration steps (Georgiou et al., 2022). These combined effects result in exponential growth of computational workload with increasing resolution. It is estimated that when the horizontal resolution of the air quality model is increased by 18 times, the computational load of the model increases by 300 times (Thompson and Selin 2012).

Enhancement of computational efficiency in air quality modeling has been predominantly achieved through hardware-based acceleration strategies. Wang et al. (2017) ported the Global Nested Air Quality Prediction Modeling System (GNAQPMS) to the second-generation Intel Xeon Phi processor (KNL), achieving a 3.5x computational acceleration via MPI and OpenMP hybrid parallelization, vectorization optimization, memory access pattern refinement, thread-local storage reduction, and global communication optimization. The gas-phase chemistry module is widely recognized as the dominant computational bottleneck in air quality models, typically accounting for over 40% of total simulation time (Elbern, 1997; Linford et al., 2011; Wang et al., 2017; Cao et al., 2023). To address this limitation, Wang et al. (2019) developed the MP CBM-Z mechanism by implementing vectorized computation techniques within the CBM-Z framework. Leveraging Single Instruction Multiple Data (SIMD) architecture, their approach enabled multi-point parallel computation for gas-phase chemistry, achieving a 4.9x acceleration in the chemistry module and a 2.22x overall speedup for the entire NAQPMS model when deployed on Intel Xeon Gold 6132 CPUs.

In recent years, GPUs have emerged as transformative accelerators in artificial intelligence and high-performance computing, driven by their massive parallel computing capabilities. In December 2024, the 64th TOP500 list of supercomputers revealed that the El Capitan system has achieved the top spot, becoming the third exascale computing system following Frontier and Aurora, with an HPL score of 1.742 EFLOP/s (Top500, 2024). This computational supremacy primarily originates from its AMD Instinct MI300A GPU accelerators, each containing 14,592 stream processors and delivering a double-precision floating-point performance of 61.3 TFLOP/s. Remarkably, the computational efficiency of a single MI300A GPU exceeds 1.8 times the peak performance of the Earth-Simulator supercomputer (CPU-based architecture) in Japan, which is Top1 supercomputer in 2003.

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

The formidable computational capacity of GPUs has opened new directions for enhancing the computational efficiency of air quality models. Alvanos and Christoudias (2017) developed a software package for the global atmospheric chemistry model ECHAM/MESSy Atmospheric Chemistry (EMAC), enabling automated generates CUDA kernels to numerically integrate atmospheric chemical kinetics by the Kinetic PreProcessor (KPP, Damian et al., 2002). Subsequent memory optimization and thread management strategies achieved a 20.4x acceleration for the chemistry module on NVIDIA P100 GPUs. In parallel efforts, Sun et al. (2018) implemented CUDA-based optimization for the second-order Rosenbrock chemical solver (Sandu et al., 1997) within the CAM4-Chem global chemistry-climate model. Through strategic enhancements in fully interleaved memory layout, CUDA streams, and constant memory, they achieved an 11.7x speedup for computation alone and a 3.8x speedup when the data transfer between the CPU and GPU is considered on the NVIDIA Tesla K20X GPU. Notably, Quevedo et al. (2025) adapted the third-order Rosenbrock solver in the CMAQ model by converting Fortran code to CUDA Fortran, evaluating its performance across three chemical mechanisms: RACM2, CB6R5, and SAPRC07. Comparative analysis revealed 51%, 50%, and 35% computational efficiency gains on NVIDIA RTX 2080 Ti GPUs, respectively, while maintaining numerical consistency with CPU-based benchmarks. Through code refactoring from Fortran to standard C and the HIP programming technology, Cao et al. (2025) successfully parallelized the fourth-order Rosenbrock solver on China's domestic GPU-like architecture. Concurrently, the total model elapsed time was reduced by 46.9%. Regarding another hotspot module in air quality models—the advection module, Cao et al. (2023, 2024) implemented GPU-accelerated adaptations of the CAMx model's advection module using CUDA and HIP heterogeneous technologies, respectively, and the optimized advection module achieved maximum speedups of 80.2x on NVIDIA Tesla V100 GPUs and 28.9x on China's domestic GPU-like accelerator.

Following Cao et al.'s (2025) successful implementation of parallel computing for the gas-phase chemistry module in the EPICC-Model on China's domestic GPU-like accelerators, the computational time proportion of this module was significantly reduced. Consequently, the advection module has emerged as a computational hotspot with comparable time consumption to the optimized chemistry module. To address this shift, this study focuses on enhancing the advection

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

module performance in EPICC-Model V1.0 via GPU-HADVPPM4HIP V1.0 coupling and GPU-optimized strategies. Sect. 2 details the EPICC-Model's computational framework, baseline performance tests, and the heterogeneous computing platform employed in this research. Sect. 3 elaborates on the optimization framework specifically designed for the computational characteristics of the EPICC-Model advection module. Sect. 4 presents experimental results, including offline benchmarking of the standalone advection module and coupled-system performance evaluation within the full EPICC-Model environment.

2. The EPICC-Model and experiments

2.1. The framework of the EPICC-Model

The emission and atmospheric processes integrated and coupled community model version V1.0 (EPICC-Model V1.0; EPICC-Model Working Group, 2025; Wang et al., 2025) is a new-generation air quality modeling system specifically designed for air pollution complex in China (Zhu et al., 2023), and developed by the Institute of Atmospheric Physics, Chinese Academy of Sciences based on the Earth System Numerical Simulation Facility (EarthLab, Chai et al., 2021). The model framework is fundamentally based on the species continuity equation and is used to simulate the complex physical and chemical processes of pollutants in the atmosphere. These processes include emissions, advection, diffusion, aerosol processes, gas-phase chemistry, and deposition. The EPICC-Model V1.0 adopts a modular architecture developed using Fortran programming language, a high-performance computing language specifically designed for scientific applications. The model code is open-source and shared (EPICC-Model Working Group, 2024a, 2024b). This open-source code repository enables rapid integration of novel mechanisms and modules proposed by diverse research groups, thereby enhancing collaborative development efficiency.

The computational framework and workflow of the EPICC-Model V1.0 are illustrated in Figure 1. The system primarily comprises three components: model inputs, physical-chemical processes, and outputs analysis. Model input data include emissions, meteorological data, and other datasets. Emissions inventories such as the Multi-resolution Emission Inventory for China (MEIC, Li et al., 2017), the Emissions Database for Global Atmospheric Research (EDGAR, Crippa et al.,

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

2024), the HTAP (Crippa et al., 2023), and the Inversed Emission Inventory for Chinese Air Quality (CAQIEI, Kong et al., 2024) can be utilized. Meteorological data are predominantly derived from simulations generated by the mesoscale Weather Research and Forecasting (WRF) model. Other datasets encompass configuration files, terrain data, TUV photolysis data, as well as initial conditions (BC) and boundary conditions (BC). Physical-chemical processes primarily include horizontal advection, vertical diffusion, dry deposition, wet scavenging, gas-phase chemistry, aqueous-phase chemistry, heterogeneous reactions, inorganic aerosol thermodynamics, etc. For the vertical diffusion module, either the scheme of Byun and Dennis (1995) or the YSU scheme (Hong et al., 2006) can be selected to calculate the turbulent vertical diffusion coefficient. The dry deposition module can employ either the scheme of Wesely (1989) or Zhang et al. (2003) to compute deposition velocities. The gas-phase chemistry module offers the option to utilize either the CBM-Z (Zaveri and Peters, 1999) or CB6r5 (Yarwood et al., 2020) chemical mechanisms. For heterogeneous reactions, the model defaults to the scheme of Li et al. (2012). Additionally, it integrates mechanisms for HONO heterogeneous chemical reactions (Zhang et al., 2022), sulfate heterogeneous chemical reactions (Li et al., 2018), and N_2O_5 heterogeneous hydrolysis (Yang et al., 2024). Inorganic aerosol is simulated using the ISORROPIA aerosol thermodynamic model (Nenes et al., 1998). The aqueous-phase chemistry module originates from the Regional Acid Deposition Model (RADM, Chang et al., 1987). Regarding model output analysis, the EPICC-Model can generate pollutant concentration fields, pollutant deposition fluxes, process analysis data, and source apportionment data. For a comprehensive technical description of the model architecture and implementation details, refer to the EPICC-Model Working Group (2025).

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

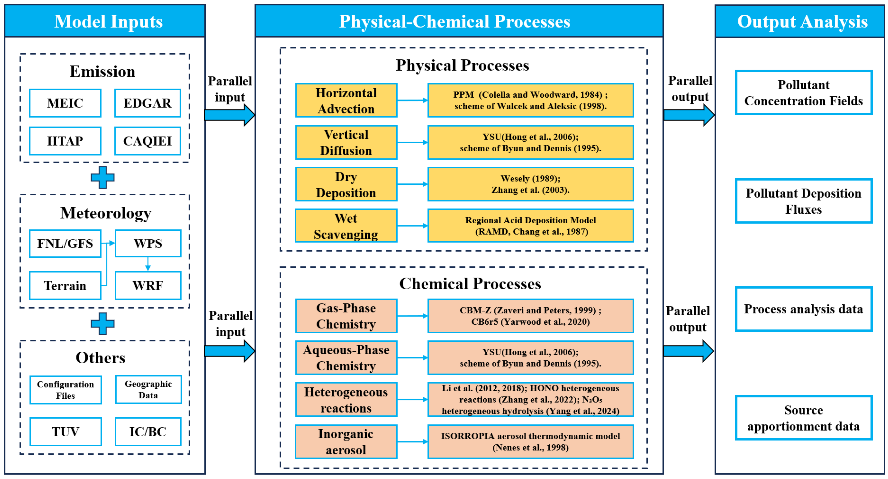


Figure 1. The computational framework and workflow of the EPICC-Model V1.0. In the section of physical-chemical processes, yellow represents the physical module, and orange indicates the chemical module.

For the horizontal advection module that is the focus in this study, two high-precision numerical schemes are available, the positive-definite mass-conservative differencing scheme (Walcek and Aleksic, 1998) and the piecewise parabolic method (PPM, Colella and Woodward, 1984). The PPM scheme, an extension of high-order Godunov's method, operates by partitioning the integration domain into subregions and approximating solutions using parabolic functions. Renowned for its numerical precision and robustness in complex fluid dynamics, this classic algorithm has been widely adopted in atmospheric chemistry models including the latest CMAQ and CAMx (Appel, et al., 2021; Emery, et al., 2024). Within the EPICC-Model V1.0 framework, the advection module sequentially executes transport processes in the *x*-direction and *y*-direction. It employs species-specific PPM solvers (HADVPPM subroutine) for gaseous species, inorganic aerosols, organic aerosols, dust, and sea salt. Our previous studies have demonstrated a significant acceleration performance of PPM solver in the CAMx model through HIP heterogeneous programming technologies for China's domestic GPU-like accelerator (Cao et al., 2024).

2.2. Benchmark performance of testing

As mentioned above, Cao et al. (2025) implemented the fourth-order Rosenbrock solver for the gas-phase chemistry module in the EPICC-Model V1.0, employing the CBM-Z chemical

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

mechanism. Through code restructure from Fortran to standard C programming language and implementation of the HIP heterogeneous programming framework, the computational efficiency of the gas-phase chemistry module improved by 2.88 times when accounting for data transfer between CPUs and the GPU-like accelerator.

In this study, the aforementioned version of EPICC-Model V1.0 was adopted as the baseline to quantify the computational time distribution across individual modules. During testing, the elapsed time of the advection module was measured using built-in timing functions in the Fortran programming language, with the PPM scheme selected for the advection solver. The experiments were launched with 10 CPU processes and configured with 10 China's domestic GPU-like accelerators. As shown in Figure 2, the implementation of parallel computing for gas-phase chemistry modules on China's domestic GPU-like accelerator achieved significant efficiency improvements, reducing its computational time proportion from 45.7% (Cao et al., 2025) to 29.9%. Notably, the MPI_Barrier synchronization function consumed 13.4% of the total runtime. A critical observation emerged regarding the advection module, whose computational time proportion increased from 13.3% (Cao et al., 2025) to 25.0%, establishing it as a new performance bottleneck comparable to the optimized chemistry module. This performance shift necessitates subsequent optimization efforts focusing on heterogeneous porting and parallel acceleration of the PPM scheme for China's domestic GPU-like architectures within the EPICC-Model V1.0 framework, aiming to enhance the computational efficiency of the advection module.

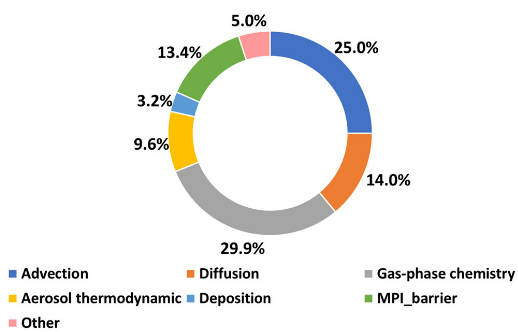


Figure 2. Proportions of computing time across main modules in the EPICC-Model V1.0.

2.3. Hardware platform and software environment of experiments

All performance testing of the EPICC-Model V1.0 and heterogeneous adaptation and optimization studies of the advection module on domestic GPU-like accelerators were conducted at

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

the Earth System Numerical Simulation Facility (EarthLab, Chai et al., 2021). Jointly developed by the Institute of Atmospheric Physics, Chinese Academy of Sciences and collaborating institutions, this platform, specifically designed for earth system modeling and high-resolution regional environmental simulation, employs a CPU and GPU heterogeneous architecture. Detailed hardware components and software environment are presented in Table 1. The Chinese domestic CPUs and GPU-like accelerators used in this studying are the first-generation versions. Each GPU-like node contains two China's domestic CPU processors and two GPU-like accelerators (Cao et al., 2024) interconnected via PCIe 4.0 buses. The software stack employs Intel OneAPI 2021.3.0 toolkit for CPU code compilation and dtk-23.04.1 toolkit for domestic GPU-like accelerator code compilation, ensuring full compatibility with heterogeneous computing paradigms.

Table 1 The hardware components and software environment for the dedicated accelerator node on the EarthLab.

	CPU	**GPU**
Hardware components	two of China’s domestic CPU processors，2.0 GHz, 32 cores	two of China’s domestic GPU-like accelerators，3840 computing units，16 GB memory
Software environment	Intel OneAPI 2021.3.0 toolkit	dtk-23.04.1

Compared to CPU processors, domestic GPU-like accelerators demonstrate superior capability in launching massive thread-level parallelism. Similar to the NVIDIA GPU architectures (NVIDIA, 2020), these domestic GPU-like accelerators employ a three-level parallelism hierarchy comprising grids, blocks, and threads, which collaboratively execute parallel computations through coordinated indexing. Specifically, a computational grid is partitioned into multiple thread blocks with three-dimensional coordinates, each thread block containing an array of three-dimensional indexed threads. As the fundamental execution unit, individual threads perform concrete computational tasks, each possessing a unique index ID that precisely determines its spatial position within the thread block hierarchy. Consequently, the design of hierarchical indexing schemes coordinating blocks and threads constitutes a critical challenge in achieving efficient parallel computation for three-dimensional numerical modeling grids.

Analogous to the AMD's ROCm software stack (AMD, 2023), the dtk-23.04.1 toolkit (Cao et

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

al., 2024) includes programming models, tools, compilers, libraries, and runtimes for artificial intelligence and high-performance computing applications on domestic GPU-like accelerators. Mirroring ROCm's design paradigm, dtk-23.04.1 adopts the HIP programming language as its application programming interface (API). This implementation leverages the Single-Instruction Multiple-Thread (SIMT) execution model to effectively manage and coordinate massive thread parallelism on China's domestic GPU-like accelerators.

3. Implementation details

3.1. Description of the heterogeneous porting and optimization scheme

The heterogeneous porting and parallel optimization schemes of this study are illustrated in Figure 3. Similar to the heterogeneous porting approach for the advection module of the air quality model CAMx on China's domestic GPU-like accelerators (Cao et al. 2024), the first step involved the porting and adaptation of the HADVPPM advection solver from the EPICC-Model V1.0 to domestic GPU-like accelerators. Firstly, the Fortran code of the HADVPPM subroutine was reconstructed using standard C programming language, followed by implementing parallel computing on domestic GPU-like accelerators through the HIP API. Similar to CUDA program execution on NVIDIA GPUs, the implementation of GPU-HADVPPM4HIP V1.0 on domestic GPU-like accelerators follows four key steps: (1) Device memory allocation via the hipMalloc interface, (2) Data transfer from CPU to domestic GPU-like accelerator through hipMemcpy operations, (3) Parallel computation using kernel launching (hipLaunchKernelGGL) with thread-index-based parallel processing after successful data transmission, and (4) Final data retrieval from GPU back to CPU through hipMemcpy operations.

Following the implementation of GPU-HADVPPM4HIP V1.0 parallel computing on China's domestic GPU-like accelerators, the second phase involves architecture-specific parallel optimizations tailored for domestic GPU-like accelerator characteristics. Three optimization strategies were sequentially implemented to fully exploit the SIMT vectorization parallelism of domestic GPU-like accelerators, thereby enhancing the computational performance of the EPICC-Model advection module. These strategies include: (1) reducing the frequency of communication between the CPU and GPU, (2) collaborative indexing between threads and blocks, and (3) hybrid

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

parallelization of "MPI+HIP". For systematic reference, three progressively optimized configurations were designated, namely HIP-Ori, HIP-Opt1, and HIP-Opt2. The HIP-Ori is baseline implementation after GPU-HADVPPM4HIP V1.0 integration into EPICC-Model without optimizations. The HIP-Opt1 is the version implementing reduced CPU-GPU communication frequency. The HIP-Opt2 is the enhanced version incorporating collaborative thread-block indexing. The hybrid "MPI+HIP" parallelization strategy was implemented across all three heterogeneous versions to enhance parallel scalability of the EPICC-Model V1.0 on the EarthLab.

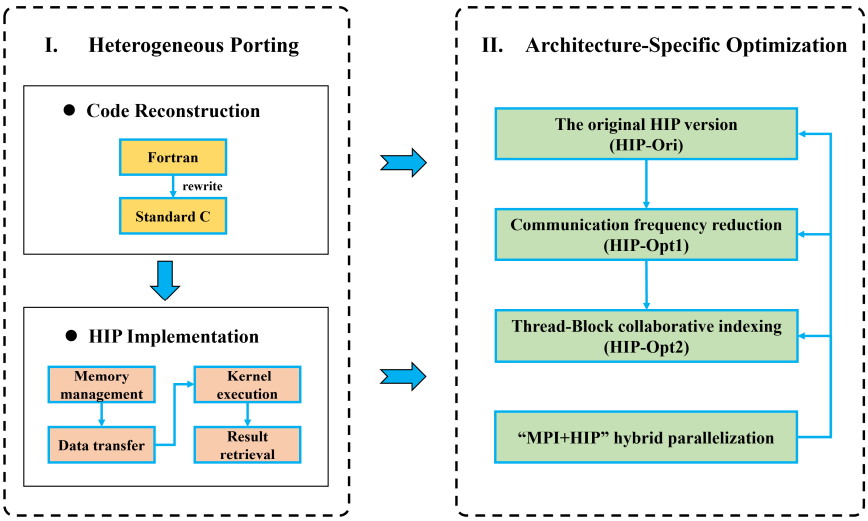


Figure 3. Heterogeneous porting and parallel optimization scheme of advection module in the EPICC-Model V1.0.

3.2. HIP-Opt1: Communication frequency reduction

Influenced by the evolutionary trajectory of high-performance computing, most geoscientific numerical models, including the EPICC-Model, are predominantly coded in Fortran and designed for general-purpose CPU architectures. These models typically execute computations through grid-wise loop iterations. Taking the HADVPPM subroutine in the EPICC-Model as an example, its computational kernel is structured with triple nested loops progressing from innermost to outermost: species loop (loop_species), latitudinal grid loop (loop_j), and vertical grid loop (loop_k). The EPICC-Model innovatively categorizes atmospheric pollutants within the species loop into five

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

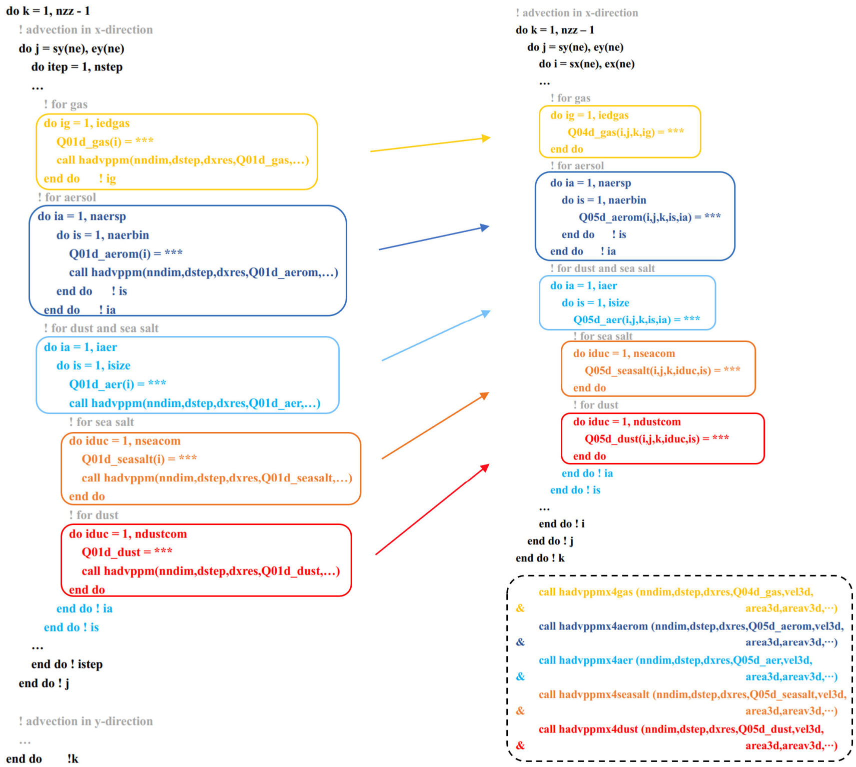
distinct classes: gaseous species (74 chemical constituents), inorganic aerosols (14 species subdivided into coarse- and fine-mode particle sizes), organic aerosols, dust aerosols (8 species across 4 size bins), and sea salt aerosols (11 species across 4 size bins).

The EPICC-Model calculates the advection process through looping of chemical-specified variables, which has relatively low computational efficiency. A benchmark test using a two nested grid configuration (horizontal grids: d01=228×165, d02=465×300) indicated that each timestep requires approximately 9.8 million calls of the HADVPPM subroutine for advection processes in both *x*-direction and *y*-direction. Consequently, the HIP-Ori version, generated by integrating GPU-HADVPPM4HIP V1.0 into the EPICC-Model, incurs 9.8 million CPU-GPU data transfers per timestep. Benchmark tests demonstrated that the computational time for 1-hour integration increased from 1,015.0 seconds in the original Fortran version to 20,400.3 seconds under HIP-Ori version, with frequent CPU-GPU communication identified as a primary performance bottleneck. To address this critical bottleneck, our optimization framework prioritizes architectural redesign of the advection module's loop hierarchy, strategically reducing communication frequency while increasing data transfer sizes to better exploit GPU computational capacity.

Figure 4 illustrates the code-level implementation of CPU-GPU communication optimization, with panel (a) depicting the HIP-Ori baseline and panel (b) presenting the optimized HIP-Opt1 version. In the two nested-domain case, the HIP-Ori configuration required approximately 4.9 million GPU calls for *x*-direction advection alone. To mitigate this computational overhead, we restructured the advection module's loop architecture and expanded array dimensionality. The HIP-Opt1 optimization framework implements multidimensional array restructuring, beginning with the dimensional expansion of concentration variables from their original 1D representations (Q01d) in HIP-Ori to 4D/5D configurations (Q04d/Q05d), while auxiliary parameters such as grid area adjustment vector (area) and interfacial area adjustment vector (areav) are similarly upgraded from 1D to 3D structures. Prior to GPU execution, these variables undergo systematic multidimensional reorganization, as demonstrated by the transformation of gaseous concentration variables from Q01d_gas(*i*) to Q04d_gas (*i, j, k, species*) in Figures 4(a)-(b). This architectural redesign enables complete *x*-direction advection computation through a single GPU call per pollutant category. Consequently, the total GPU calls for both *x* and *y* direction advection decrease from approximately

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

9.8 million in HIP-Ori to 10 in HIP-Opt1, achieved through one GPU call per spatial dimension across five pollutant categories, thereby optimizing computational efficiency through batched multidimensional data processing.



(a) Baseline code of the HIP-ori. **(b) Optimized code of the HIP-Opt1.**

Figure 4. The code-level implementation of CPU-GPU communication optimization. Panel (a) is the baseline Fortran code of the HIP-ori Panel (b) is the optimized Fortran code of the HIP-Opt1.

3.3. HIP-Opt2: Thread and block coordinated indexing

The architectural advantage of China's domestic GPU-like accelerators manifests in their capacity to support massive thread concurrency for parallel computing. To leverage this capability, the coordinated thread-block indexing methodology which proposed by Cao et al. (2023) was implemented, whereby in which each grid cell in the two-dimensional horizontal plane is assigned a dedicated thread. Specifically, blocks were configured based on the meridional grid dimension, with each block allocated threads corresponding to the zonal grid count. This hierarchical parallelization strategy achieves comprehensive full parallel processing of across the two-

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

dimensional planar grid structure through coordinated thread-block resource allocation.

3.4. "MPI+HIP" hybrid parallelization

The recent advancements in GPU technology have solidified the dominance of "CPU+GPU" heterogeneous architectures in global high-performance computing, with 9 out of the top 10 supercomputers in 2024 utilizing heterogeneous configurations (Top 100, 2024). These large-scale heterogeneous clusters typically deploy one or multiple GPU accelerators per compute node. Aligned with this trend, EarthLab employs heterogeneous architecture in its dedicated computing nodes, integrating China's domestic CPUs with GPU-like accelerators. To maximize GPU utilization and enhance the parallel scalability of the EPICC-Model on heterogeneous clusters, an "MPI+HIP" hybrid parallelization scheme was designed tailored for EarthLab, inspired by the "MPI+CUDA" approach proposed by Cao et al. (2023) for the CAMx model. As illustrated in Figure 5 using an 8 CPU cores and 8 GPUs configuration, the framework assigns one domestic GPU-like accelerator to each CPU process via MPI and HIP hybrid parallelization. The EPICC-Model divides the simulation domain into 8 subregions using the Message Passing Interface (MPI) software standard, with each CPU process handling its allocated subdomain. During advection module execution, computational tasks originally processed by CPUs are offloaded to corresponding GPUs, with results subsequently returned to CPUs. Given EarthLab's node configuration of 2 domestic GPU-like accelerators per accelerator node, only 2 CPU processes are launched per node to ensure optimal resource pairing.

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

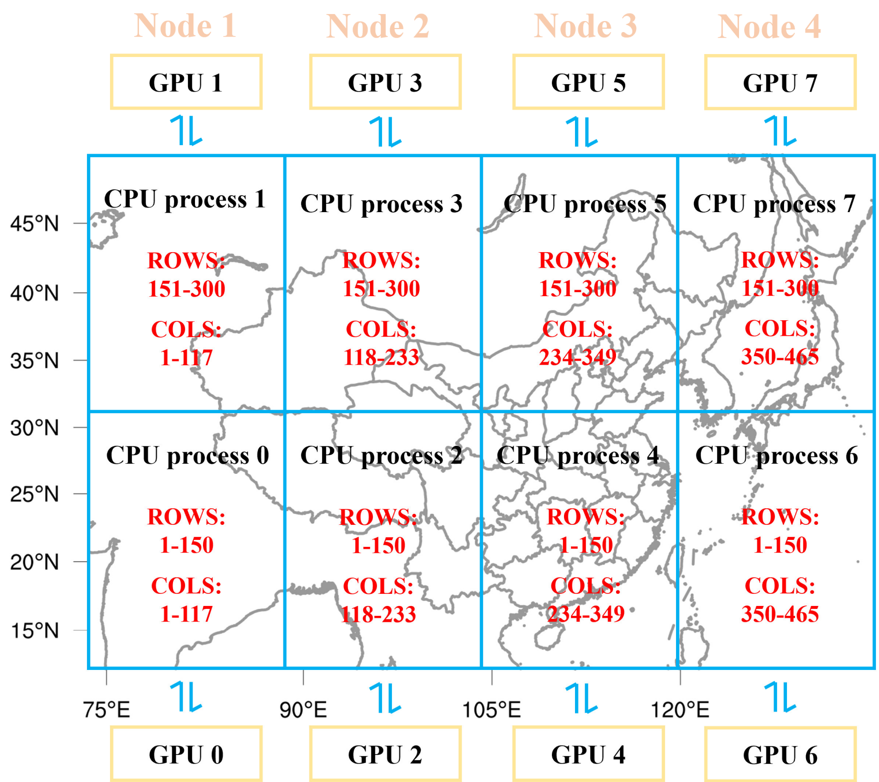


Figure 5. An example schematic of domain subdivision and mapping to CPU processes, where each CPU process is equipped with a domestic GPU-like accelerator.

4. Experimental results

4.1. Experimental setup

The centre of the simulation domain is located at (35 °N, 105 °E) and its two true latitude lines are 25 °N and 45 °N, respectively. The EPICC-Model V1.0 employs a two-nested configuration, the first domain (d01) covers East Asia with a 45 *km* × 45 *km* horizontal resolution on 228×165 grid cells. The lower left corner of the second domain has its starting positions in the grid of the first domain as 36 and 39, respectively, and the second domain focuses on China with a 15 *km* × 15 *km* horizontal resolution on 465×300 grid cells. In vertical, 20 terrain-following layers are configured with the height of the top layer set to 20 *km* and six layers below 1 *km*. The numerical simulation was conducted from 00:00 UTC July 1 to 23:00 UTC on 31 July, 2021, spanning a total duration of

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

744 hours. The initial 168-hour period was allocated for model spin-up. The MEIC emission inventory (Li et al., 2017) was adopted as the emission input, and baseline year is 2019. The numerical schemes selected during the EPICC-Model V1.0 simulation are listed in Table 2. Furthermore, the $PM_{2.5}$ and O_3 observations are from the China National Environmental Monitoring Centre, which provides hourly $PM_{2.5}$ and O_3 observations for eight cities in China. The station information, including station name and its latitude and longitude, is listed in Table 3. Meteorological inputs are generated using the Weather Research and Forecasting (WRF, Skamarock et al., 2008) model, a state-of-the-art mesoscale numerical weather prediction system, and is widely adopted in both theoretical research and operational forecasting. This study employed the WRF version 3.9.1, and the model domain configurations maintains identical nesting architecture and spatial coverage as the EPICC-Model.

Table 2. The physical and chemical numerical schemes selected during EPICC-Model V1.0 simulation.

Process	Numerical schemes
Horizontal advection	PPM (Colella and Woodward, 1984)
Vertical diffusion	YSU scheme (Hong et al., 2006)
Gas-phase chemistry	CBM-Z (Zaveri and Peters, 1999)
Aqueous-phase chemistry and wet deposition	RADM (Chang et al., 1987)
Dry deposition	Scheme of Wesely (1989)
Inorganic aerosol thermodynamic partitioning	ISORROPIA v1.7 (Nenes et al., 1998)

Table 3. The names and latitude-longitude information of the $PM_{2.5}$ and O_3 observation stations.

Name	Latitude (°N)	Longitude (°E)
Beijing	40.2865	116.1700
Taiyuan	37.7394	112.5583
Hangzhou	30.3058	120.3480
Hefei	31.7386	117.2780
Fuzhou	25.9664	119.5189
Qingdao	36.1032	120.3664
Guangzhou	23.5538	113.5890
Kunming	24.9786	102.7997

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

4.2. Simulation performance analysis

4.2.1. Offline error analysis of the GPU-HADVPPM4HIP

As elaborated in Sect. 3.1, the implementation of the HADVPPM Fortran code on China's domestic GPU-like accelerators comprises two principal phases. Initially, the Fortran code undergoes reconstruction using standard C programming language through a Fortran-to-C conversion process. Subsequently, the standard C-version HADVPPM program is adapted to domestic GPU-like accelerators through C-to-HIP expansion employing the HIP heterogeneous programming technology. Following successful GPU adaptation, the offline precision verification was conducted to compare computational accuracy among three implementations, the original Fortran source code, restructured standard C code, and HIP-accelerated code. To ensure input consistency, a dedicated Fortran program was developed to generate identical input datasets, including 100 double-precision floating-point numbers, for all three implementations. Each implementation executed a complete advection integration computation, with subsequent output recording and analysis. Notably, both Fortran and C implementations were executed on China's domestic CPUs and compiled using the Intel OneAPI 2021.3.0 tookit, while GPU-HADVPPM4HIP was compiled with the dtk-23.04.1 toolkit for GPU execution. For enhanced analytical rigor, we further implemented compilation with -O0 and -O3 optimization flags. The -O0 flag maintains default compilation settings without code optimization, whereas -O3 flag enables more aggressive loop and memory-access optimizations, such as scalar replacement, loop unrolling (Intel Software, 2018).

Table 4 presents the mean absolute error (AE) and relative errors (RE) in computational precision between the Fortran and standard C implementations (F-to-C), standard C and HIP implementations (C-to-HIP), and Fortran and HIP implementations (F-to-HIP) of the HADVPPM program under two compilation configurations. Notably, the -O3 optimization flag prioritizes computational performance through code optimization at the expense of precision degradation. Consequently, the AE and RE values for all three porting processes (F-to-C, C-to-HIP, and F-to-HIP) under the -O0 configuration flag are systematically lower than those under -O3 flag. For instance, between the Fortran and HIP implementations, the AE and RE under -O0 flag are measured at 1.4×10^{-7} and $4.3 \times 10^{-7}\%$, respectively. However, when compiled with -O3 flag, these errors increase significantly to 3.1×10^{-7} and $2.5 \times 10^{-6}\%$, respectively. Similar error escalation

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

patterns are observed in the F-to-C and C-to-HIP comparisons under -O3 optimization flag. This phenomenon aligns with expectations, as aggressive compiler optimizations (e.g., loop unrolling and memory access reorganization) may introduce numerical instability through altered operation sequences and reduced intermediate precision preservation.

Furthermore, it is noteworthy that across both -O0 and -O3 compilation configurations, the AE and RE values of the F-to-C process consistently exceed those of the C-to-HIP process. This indicates that the computational errors introduced during the heterogeneous porting of the HADVPPM Fortran code from domestic CPUs to GPU-like accelerators predominantly originate from the Fortran-to-C transcoding phase. For instance, under the -O0 configuration, the AE and RE for F-to-C are measured at 1.5×10^{-7} and 5.1×10^{-7}%, respectively, whereas those for C-to-HIP are significantly smaller, at -9.5×10^{-9} and -8.0×10^{-8}%, respectively. This discrepancy arises from inherent differences between Fortran and C in programming paradigms and data precision management. The Fortran-to-C code restructuring involves two fundamentally distinct programming languages, differing in object-oriented design philosophies and numerical representation conventions, thereby introducing computational inaccuracies. In contrast, the HIP programming model, analogous to NVIDIA CUDA, is inherently an extension of standard C. As detailed in Sect. 3.1, HIP achieves GPU compatibility by augmenting standard C programming language with critical GPU-specific functionalities, such as memory allocation and data transfer operations. Since HIP code essentially constitutes an enhanced standard C framework, the C-to-HIP adaptation introduces minimal computational bias, resulting in markedly lower error compared to the F-to-C transformation.

Table 4. Comparative of mean AE and RE across compilation configurations for F-to-C, C-to-HIP, and F-to-HIP Processes.

	AE			RE (%)		
	F-to-C	C-to-HIP	F-to-HIP	F-to-C	C-to-HIP	F-to-HIP
-O0	1.5×10^{-7}	-9.5×10^{-9}	1.4×10^{-7}	5.1×10^{-7}	-8.0×10^{-8}	4.3×10^{-7}
-O3	5.4×10^{-7}	-2.4×10^{-7}	3.1×10^{-7}	2.8×10^{-6}	-2.9×10^{-7}	2.5×10^{-6}

4.2.2. Simulation performance verification of the EPICC-Model

After parallelizing the HADVPPM program in the air quality model CAMx on China's domestic GPU-like accelerators, Cao et al. (2024) conducted comparative analyses of simulation

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

results between NVIDIA GPUs and domestic GPUs through offline and coupled testing approaches. Although both experimental results indicated smaller computational errors introduced by China's domestic GPUs, the study did not validate the discrepancies between CAMx simulation results and actual observational data, particularly regarding model performance in real-case scenarios. To address this gap, the current study integrates GPU-HADVPPM4HIP V1.0 into the EPICC-Model V1.0 and performs one-month real-case simulations following the experimental configuration described in Sect. 4.1. This serves dual purposes: firstly, to verify the computational stability of EPICC-Model V1.0 in cross-architecture heterogeneous cluster environments using China's domestic hardware, and secondly, to evaluate the model's pollutant simulation performance through observational validation. For observational data, we collected $PM_{2.5}$ and O_3 observations from major Chinese cities including Beijing, Taiyuan, Hangzhou, Hefei, Fuzhou, Qingdao, Guangzhou, and Kunming, implementing quality control procedures following the methodology established by Wu et al. (2018). Regarding simulation data, we extracted model outputs from grid cells in the d02 domain corresponding to the geographical coordinates of monitoring stations.

Figure 6 and Figure 7 along with Table 5, present the time series comparisons between daily simulated and observed $PM_{2.5}$ and O_3 concentrations, as well as relevant statistical metrics, following the one-month simulation after coupling GPU-HADVPPM4HIP V1.0 to the EPICC-Model V1.0. The formulas for calculating statistical parameters are detailed in the Supplementary Materials. For daily $PM_{2.5}$ concentrations, the EPICC-Model V1.0 demonstrated robust simulation performance across most cities, with slight overestimations observed in Beijing and Hefei. Notably, the model effectively captured the temporal variations of $PM_{2.5}$ in Fuzhou, achieving a root mean square error (RMSE) of 3.9 $\mu g \cdot m^{-3}$ and a correlation coefficient (R) of 0.89. Across the eight cities, the mean RMSE and R between simulated and observed $PM_{2.5}$ were 9.4 $\mu g \cdot m^{-3}$ and 0.70, respectively. Regarding daily maximum 8-hour average (MDA8) O_3 concentrations, the model exhibited minor underestimations at certain times in Taiyuan, and Hefei but performed well in Beijing, Fuzhou, Qingdao, and Guangzhou. In Qingdao and Guangzhou, the correlation coefficient between simulated and observed O_3 reached 0.91 and 0.87, with RMSE values of 12.2 ppbV and 8.7 ppbV, respectively. The mean RMSE and R for O_3 across eight cities were 12.1 ppbV and 0.76. These statistical results indicate that the EPICC-Model V1.0, integrated with GPU-

HADVPPM4HIP V1.0, reasonably reproduces the spatiotemporal characteristics of $PM_{2.5}$ and O_3 concentrations on China's domestic heterogeneous computing clusters.

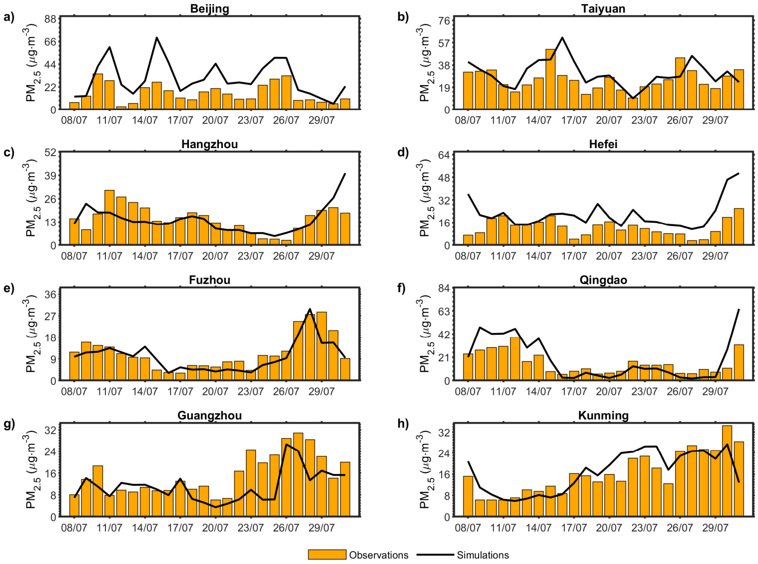


Figure 6. Time series of daily observed and simulated $PM_{2.5}$ concentrations in major cities of China on 8-31 July, 2021. Panels **(a) - (h)** represent Beijing, Taiyuan, Hangzhou, Hefei, Fuzhou, Qingdao, Guangzhou, and Kunming.

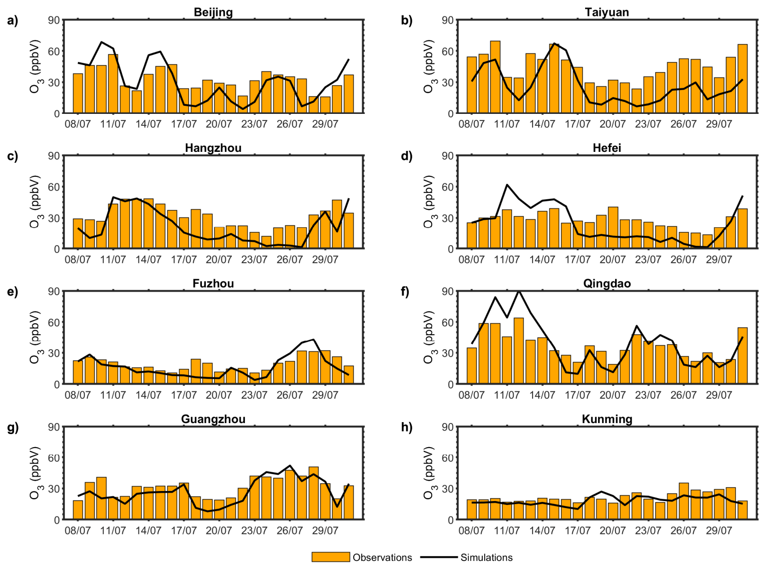


Figure 7. Time series of observed and simulated MAD8 O_3 concentrations in major cities of China on 8-31 July, 2021. Panels **(a) - (h)** represent Beijing, Taiyuan, Hangzhou, Hefei, Fuzhou, Qingdao,

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

Guangzhou, and Kunming.

Table 5. The statistics of the $PM_{2.5}$ and O_3 simulations of EPICC-Model over different eight cities.

	$PM_{2.5}$		**O_3**	
	RMSE ($\mu g \cdot m^{-3}$)	R	RMSE (ppbV)	R
Beijing	17.3	0.84	13.2	0.77
Taiyuan	11.0	0.54	21.6	0.78
Hangzhou	7.4	0.50	14.8	0.80
Hefei	12.4	0.54	14.2	0.70
Fuzhou	3.9	0.87	7.5	0.77
Qingdao	11.0	0.90	12.2	0.91
Guangzhou	7.2	0.65	7.8	0.87
Kunming	5.3	0.76	6.0	0.46
Average	9.4	0.70	12.1	0.76

The underestimations or overestimations of $PM_{2.5}$ and O_3 simulations observed in specific cities and periods are primarily attributable to two factors. First, the coarse model resolution—the horizontal resolution of the d02 domain in this experiment is 15 *km*—hindered the accurate representation of terrain features, meteorological variables, and spatial variations in emission sources. Second, discrepancies exist between the baseline year of the emission inventory and the simulation year. Specifically, the MEIC emission inventory used in this study is based on 2019 data, whereas the simulation year is 2021. For cities with stringent pollution control policies, such as Beijing, the 2019 MEIC inventory may inadequately reflect actual emission reductions achieved by 2021, particularly for pollutants under strict abatement measures. This discrepancy could lead to overestimated simulated concentrations.

4.3. Computational performance analysis

4.3.1. Offline performance comparison

A comparison of offline computational results indicates that the computational errors introduced during the Fortran-to-HIP heterogeneous porting process under both compilation settings are minimal. Specifically, the HADVPPM program exhibits small discrepancies on the order of 10^{-7} when ported from CPU to domestic GPU-like accelerator architectures. Based on the verified consistency of offline results, the computational performance of the HADVPPM program was further evaluated on domestic CPU and GPU-like accelerator under different compilation configurations. To achieve this, Fortran-based test programs were implemented to generate randomized input arrays with varying scales, ranging from 10^2 to 10^8, for both Fortran and HIP versions of the HADVPPM program. Figure 8 illustrates the computational time and speedup ratios

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

of the HADVPPM program across different compilation options and data scales on domestic CPU and GPU-like accelerator. It is explicitly stated that the execution time measurements for the HADVPPM program on the domestic GPU-like accelerator exclusively account for the computational time of the kernel functions on the device, while overheads such as GPU memory allocation and host-device data transfer are excluded.

As illustrated in Figure 8(a), under both -O0 and -O3 compilation flags, the SIMT vectorized parallel computing advantages of the domestic GPU-like accelerator become prominent for large data scales exceeding 10^4, demonstrating significantly higher computational efficiency compared to domestic CPU. At a data scale of 10^8 with the -O0 flag, the Fortran-based HADVPPM program required approximately 11.97 seconds to complete computation on the CPU, while the HIP version on the domestic GPU-like accelerator achieved the same task in 0.55 seconds, yielding a speedup ratio of 21.87x. The -O3 compilation flag further enhances computational efficiency through automated code optimization, albeit at a slight cost to numerical precision. At the same 10^8 data scale, the Fortran version on the CPU required 2.75 seconds, whereas the HIP version on the GPU completed computations in 0.02 seconds, achieving a remarkable speedup of 128.03×. Notably, the HIP version compiled with -O3 exhibited 556.5x higher efficiency than the Fortran version compiled with -O0 flag. However, for smaller data scales such as 10^2 or 10^3, the architectural advantages of the domestic GPU-like accelerator diminish, with computational efficiency comparable to or even lower than that of a CPU. For instance, at a 10^2 scale with the -O3 option, the GPU's performance matched the CPU (speedup ratio is approximately 1). Under the -O0 option, the GPU's speedup dropped to 0.16×, indicating inferior efficiency relative to the CPU. These results underscore that the domestic GPU-like accelerator is well-suited for large-scale matrix parallel computing tasks.

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

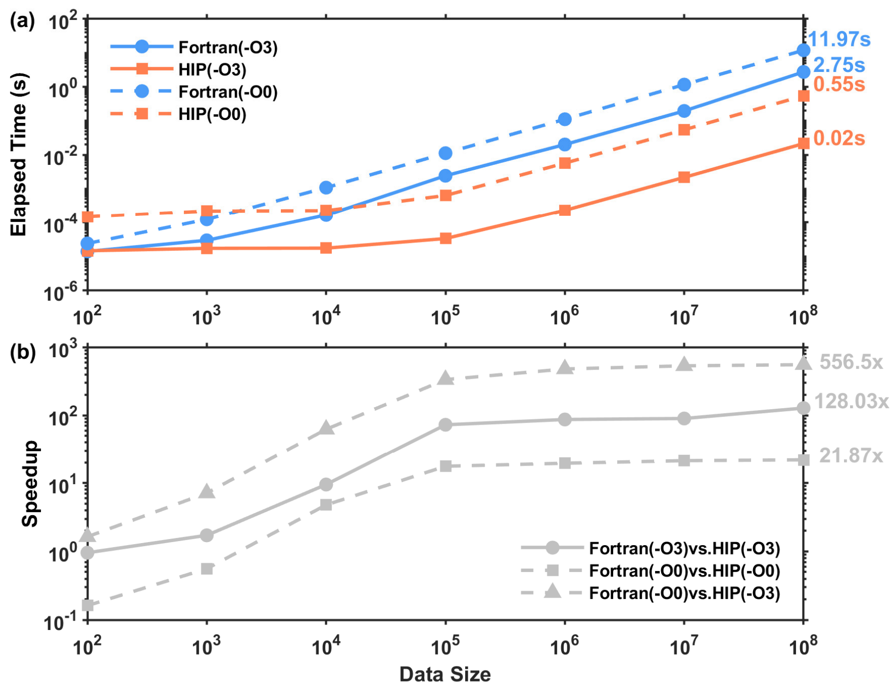


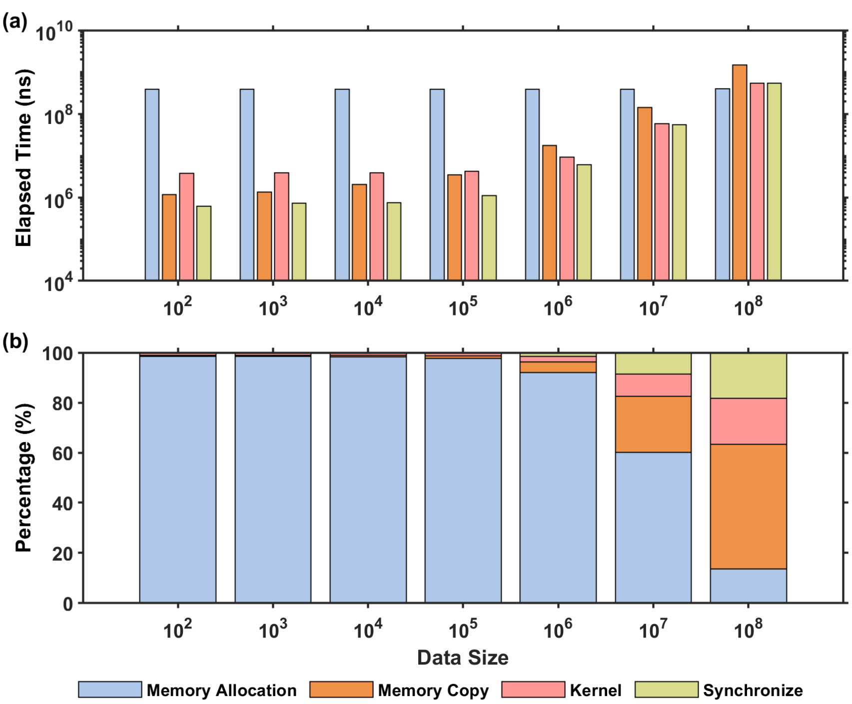
Figure 8. The offline computational time **(a)** and speedup ratios **(b)** of the HADVPPM program across different compilation options and data scales on domestic CPU and GPU-like accelerator.

As described in Sect. 3.1, a complete heterogeneous computation of GPU-HADVPPM4HIP V1.0 on the domestic GPU-like accelerator involves critical processes such as GPU memory allocation, data transfer between CPU and GPU, and kernel launching for parallel computing. To quantify the overheads of these processes, the time consumption and proportional contributions was analyzed under the -O0 compilation flag and varying data scales. In Figure 9, the label of "Memory Allocation", "Memory Copy", "Kernel", and "Synchronize" represent the processes of memory allocation on the GPU, bidirectional data transfer between CPU and GPU, kernel launch and parallel computation, as well as thread synchronization within the kernel, respectively.

As shown in Figure 9, for data scales smaller than 10^6, memory allocation dominates the time consumption, exceeding 90% of the total execution time and significantly surpassing the durations of the other three processes. Notably, the time required for memory allocation remains approximately 0.4 seconds regardless of increases in data scale. The dominance of memory allocation at small data scales highlights its fixed overhead nature. This suggests that memory

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

allocation is largely independent of data volume. While negligible in large-scale computations, this fixed cost becomes a critical bottleneck for small-scale tasks. When the data scale surpasses 10^5, the overhead of memory copy rises rapidly, with a growth rate higher than those of kernel execution and synchronization processes. At a data scale of 10^8, memory copy accounts for approximately 50% of the total time and exhibits a tendency for further increase. Under this condition, the time contributions of memory allocation, kernel execution, and synchronization are approximately 14%, 18%, and 18%, respectively. The rapid escalation of memory copy overhead underscores the limitations of host-device data transfer bandwidth. The growth rate of memory copy time implies that the data transfer between the CPU and GPU becomes one of the primary factors influencing the computational performance of programs in heterogeneous cluster environments, and the I/O-bound workloads often underutilize GPU compute capabilities. Future efforts could focus on reducing communication overhead through strategies such as unified memory architecture and asynchronous communication. Additionally, integrating mixed-precision methods (Váňa et al., 2017), converting variables with minimal impact on simulation results from double-precision to single-precision, could further enhance data transfer efficiency between CPUs and GPUs.



https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

Figure 9. Time consumption **(a)** and proportional contributions **(b)** for memory allocation, memory copy, kernel, and synchronize process under the -O0 compilation flag and varying data scales.

4.3.2. Coupling performance comparison

Following the integration of GPU-HADVPPM4HIP V1.0 into the EPICC-Model V1.0, computational efficiency on the EarthLab was improved through communication optimization described in Sect. 3.2 and enhanced thread and block collaborative indexing detailed in Sect. 3.3. Furthermore, the hybrid parallelization scheme outlined in Sect. 3.4 was employed to extend the parallel computing scalability of the EPICC-Model V1.0 on the EarthLab. As introduced in Sect. 3.1, three model versions were established: the baseline unoptimized version HIP-Ori, the communication-optimized version HIP-Opt1, and the collaboratively indexed version HIP-Opt2 Figure 10(a) displays the average elapsed time required for a 28-hour simulation across these three versions. To ensure comparability, all tests adopted identical hardware configurations, including the MPI+HIP hybrid parallelization scheme with 10 CPU processes and 10 domestic GPU-like accelerators, and were compiled using the -O3 optimization flag.

The GPU-HADVPPM4HIP V1.0 module features triple nested loops, and each invocation by the EPICC-Model V1.0 necessitates data transfer between the CPU and domestic GPU-like accelerator. Frequent CPU-GPU data transfer severely compromised the computational efficiency of the EPICC-Model V1.0 on the EarthLab. In the HIP-Ori unoptimized version, completing a 28-hour simulation required 149.7 hours, with an average hourly elapsed time of 19,241.0 seconds, equivalent to approximately 5.3 hours, reflecting critically low efficiency. After implementing communication optimizations in the HIP-Opt1 version, the communication frequency between the CPU and domestic GPU-like accelerator was reduced from roughly 9.8 million to 10. This optimization drastically decreased the total elapsed time from 149.7 hours for HIP-Ori to 8.8 hours for HIP-Opt1, while the average hourly elapsed time dropped from 19,241.0 seconds to 1,133.7 seconds, achieving a 17.0x improvement in computational efficiency. Subsequently, collaborative indexing of threads and blocks was applied to parallelize the two-dimensional grid computations in the EPICC-Model V1.0. Compared to HIP-Opt1, the HIP-Opt2 version further reduced the total elapsed time from 8.8 hours to 6.0 hours, with the average hourly elapsed time decreasing from 1,133.7 seconds to 768.6 seconds. This enhancement delivered an additional 1.5x efficiency gain. Cumulatively, these optimizations elevated the computational efficiency of the EPICC-Model V1.0

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

on the EarthLab by approximately 25.0x.

Figure 10(b) further compares the computational performance between the original Fortran-based version and the HIP-Opt2 version. In the figure, the label Fortran refers to the legacy CPU cluster implementation of the EPICC-Model V1.0, while HIP represents the HIP-Opt2 version. Following the parallelization of the advection module on China's domestic GPU-like accelerators, the EPICC-Model V1.0 demonstrated superior computational efficiency on heterogeneous clusters compared to its Fortran counterpart on conventional CPU clusters. The total elapsed time for a 28-hour simulation decreased from 2,287.4 seconds for the Fortran version to 2,152.0 seconds for the HIP version, with the average hourly elapsed time reduced from 817.0 seconds to 768.6 seconds.

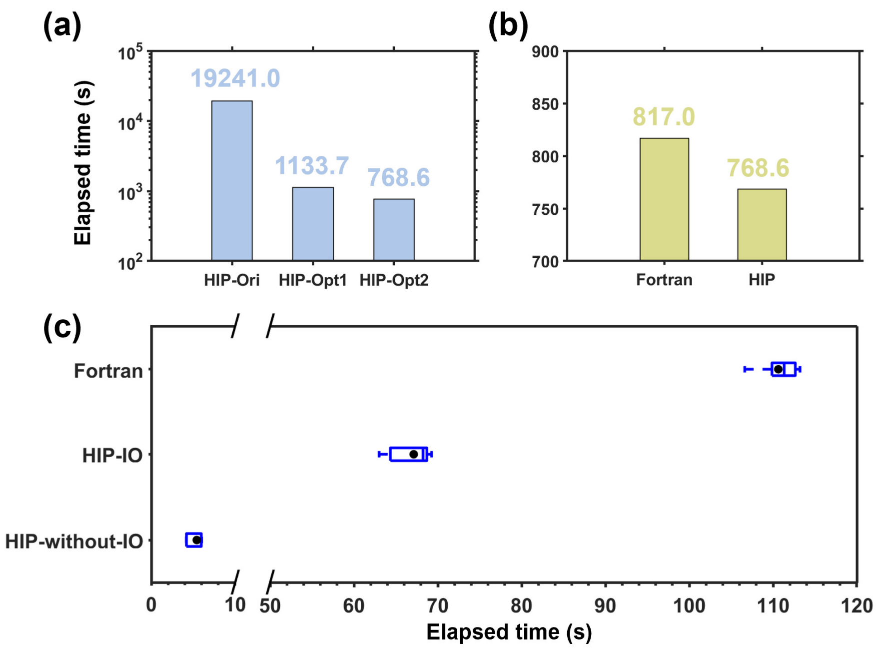


Figure 10. (a) The average hourly elapsed time required for a 28-hour simulation across HIP-Ori, HIP-Opt1, Opt2 versions; **(b)** the average hourly elapsed time required for a 28-hour simulation between the Fortran and HIP-Opt2 versions; **(c)** the hourly elapsed time required for a 28-hour simulation across the original Fortran-based HADVPPM program on CPUs, the GPU-HADVPPM4HIP V1.0 with CPU-GPU data transfer, and the GPU-HADVPPM4HIP V1.0 without data transfer. The black dots represent the average values, and the unit is seconds (s).

As demonstrated by the timing analysis of key processes in Sect. 4.3.1 for the offline

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

heterogeneous computation of GPU-HADVPPM4HIP V1.0 on domestic GPU-like accelerators, memory copying, specifically data transfer between the CPU and GPU, emerges as the dominant factor influencing parallel computational efficiency on heterogeneous clusters when handling large-scale datasets. This overhead surpasses the time spent on kernel function parallelization. To quantify this effect, the computational time of GPU-HADVPPM4HIP V1.0 was separately evaluated on domestic GPU-like accelerators under two scenarios: (1) including CPU-GPU data transfer and (2) excluding data transfer (kernel-only computation). In Figure 10 (c), the y-axis labels Fortran, HIP-IO, and HIP-without-IO correspond to the computational time of the original Fortran-based HADVPPM module on CPUs, the GPU-HADVPPM4HIP V1.0 with CPU-GPU data transfer on domestic GPU-like accelerators, and the GPU-HADVPPM4HIP V1.0 without data transfer on domestic GPU-like accelerators, respectively. All tests were compiled with the -O3 optimization flag, and timing metrics were averaged over a 28-hour simulation, focusing on the hourly computational cost of the advection module.

The original Fortran-based advection module required an average of 110.6 seconds per simulated hour on CPUs. After heterogeneous porting and parallel optimization to domestic GPU-like accelerators, the GPU-HADVPPM4HIP V1.0 with data transfer achieved an average time of 67.1 seconds per simulation hour, representing a 39.3% improvement in computational efficiency. When excluding data transfer, the kernel-only GPU-HADVPPM4HIP V1.0 reduced the average time to 5.4 seconds per simulation hour, achieving a 20.5x acceleration compared to the Fortran version. These results underscore that while domestic GPU-like accelerators deliver substantial computational power, the efficiency of CPU-GPU data transfer critically constrains overall performance on heterogeneous clusters. To mitigate this bottleneck, future work will focus on retaining input/output (I/O) operations on CPUs while porting the entire physicochemical integration module (excluding I/O) to GPUs for parallel computation. This strategy is expected to reduce the impact of inter-device data transfer and further enhance scalability. Furthermore, leveraging the characteristics of air quality forecasting, computational efficiency can be enhanced by retaining only essential I/O for conventional pollutants (e.g., CO, PM_{10}, $PM_{2.5}$, SO_2, O_3, NO_2) and their associated variables, while eliminating non-critical variables such as sea salt aerosols. This selective I/O optimization would further streamline data transfer efficiency between CPUs and

GPUs.

5. Conclusions and discussion

In recent years, the rapid advancement in the computational performance of GPUs has provided novel approaches and hardware foundations for improving the computational efficiency of air quality models. Building upon the heterogeneous porting and parallel optimization technology system for air quality model, this study further implements parallel computing of the advection module in the EPICC-Model air quality modeling system on China's domestic GPU-like accelerators, validating the feasibility of the heterogeneous porting framework. The study involves three key technical improvements. The first is restructuring the original Fortran code of the advection module using standard C language programming. The second is porting the advection module to China's domestic GPU-like accelerators through HIP heterogeneous programming technology, in addition, computational efficiency was enhanced through optimizing CPU-GPU data transfer frequency reduction, coordinated indexing of threads and blocks, and hybrid parallelization strategies. These optimizations collectively improved both the computational performance of the advection module and the parallel computing scalability of the EPICC-Model V1.0 on the EarthLab.

This study systematically conducted comparative efficiency analyses by the validation of computational consistency in GPU-HADVPPM4HIP V1.0 through offline testing methodologies, as well as the verification of EPICC-Model's pollutant simulation performance on the EarthLab. Initial benchmarking compared the offline computational efficiency between the original Fortran-based HADVPPM program on domestic CPUs and the GPU-HADVPPM4HIP V1.0 implementation. The results demonstrated that the -O3 compiler optimization flag significantly enhanced GPU-HADVPPM4HIP's computational efficiency, with acceleration effects becoming more pronounced at larger data scales. Specifically, at 10^8 data size configuration, GPU-HADVPPM4HIP V1.0 achieved a maximum 556.5x speedup over the Fortran baseline using default -O0 compilation, while maintaining a 128.0x speedup advantage even when both implementations employed -O3 optimization. Further profiling of GPU-HADVPPM4HIP's heterogeneous computation on domestic GPU-like accelerators revealed critical characteristics: Memory copy operations (i.e., CPU-GPU data transfers) exhibited elapsed time increases rapidly with data size

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

increasing, accounting for approximately 50% of total computation time at 10^8 data size with a continuing upward trend. This observation underscores data transfer efficiency as a critical bottleneck for high-resolution air quality simulations in heterogeneous computing environments.

Coupling GPU-HADVPPM4HIP V1.0 into EPICC-Model V1.0 with data transfer optimizations and thread-block coordinated indexing strategies yielded system-level performance improvements of 17.0x and 1.5x respectively on the EarthLab. The detailed module-level analysis demonstrated that GPU-HADVPPM4HIP V1.0 achieved 39.3% efficiency enhancement over the original Fortran advection module when accounting for CPU-GPU data transfer overheads, escalating to a 20.5x acceleration when excluding data transfer costs. These findings quantitatively validate the substantial impact of CPU-GPU data transfer efficiency on the operational performance of air quality models in heterogeneous computing architectures.

For the parallel computing implementation of the air quality model EPICC-Model V1.0 on the EarthLab, several critical research directions warrant further investigation. First, priority should be given to optimizing CPU-GPU data transfer efficiency by reducing communication overhead through strategies such as unified memory architecture, asynchronous communication, mixed-precision methods, and minimizing non-essential variable I/O in air quality forecasting. Second, while GPU-accelerated modules including the gas-phase chemistry module (Cao et al., 2025) and advection module have been individually developed, their systematic integration into EPICC-Model requires architectural refinement to increase GPU code coverage. Concurrently, parallel computing implementations for other computationally intensive modules should be pursued. Third, while the current implementation employs two-dimensional thread-block indexing to achieve parallel computation for horizontal grid structures, future development will focus on adopting three-dimensional (3D) indexing strategies to enable full 3D grid parallelism.

Code and data availability. The source codes of EPICC-Model are available online via ZENODO (https://doi.org/10.5281/zenodo.17071574, EPICC-Model Working Group, 2024b). The EPICC-Model codes are only accessible for research and educational purposes; commercial utilization is strictly prohibited. To request access, eligible users must contact Working-Group@EPICC-Model.cn from an institutional email address, providing personal details, affiliation, and a statement of intended use. Access will be granted upon signing and returning the required user agreement. The datasets and codes related to this study are available online via ZENODO (https://doi.org/10.5281/zenodo.16916413, Cao and Wu, 2025).

Author contributions. KC and QW refactored the existing code, visualization, and prepared the materials. QW, XT, JZ, and ZW planned and organized the project. KC, QW, JinL, XC, HC, WW, and LK optimized the GPU-based codes. HC, WW, HW, and JieL prepared the data and conducted the simulation. KC, QW, TX, XC, WW, JieL, JZ, and ZW carried out formal analysis of the model results. KC, QW, TX, JinxL, XC, HC, WW, LK, and ZW took part in the discussion.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. The National Key Research and Development Program of China (grant no. 2023YFC3705705), the Strategic Priority Research Program of the Chinese Academy of Sciences (grant No. XDB0760401), the State Key Laboratory of Atmospheric Environment and Extreme Meteorology (grant no. 2024QN08), the National Natural Science Foundation of China (grant no. 42377105), and the Key Research and Development Program of Henan Province (grant no. 241111212300) funded this work. The authors would like to thank for the technical support of the National large Scientific and Technological Infrastructure “Earth System Numerical Simulation Facility” (https://cstr.cn/31134.02.EL).

Financial support. This research has been supported by the National Key Research and Development Program of China (grant no. 2023YFC3705705), the Strategic Priority Research Program of the Chinese Academy of Sciences (grant No. XDB0760401), the State Key Laboratory of Atmospheric Environment and Extreme Meteorology (grant no. 2024QN08), and the National

Natural Science Foundation of China (grant no. 42377105).

Reference

Adani, M., D'Isidoro, M., Mircea, M., Guarnieri, G., Vitali, L., D'Elia, I., Ciancarella, L., Gualtieri, M., Briganti, G., Cappelletti, A., Piersanti, A., Stracquadanio, M., Righini, G., Russo, F., Cremona, G., Villani, M. G., and Zanini, G.: Evaluation of air quality forecasting system FORAIR-IT over Europe and Italy at high resolution for year 2017, Atmos. Pollut. Res., 13, 10.1016/j.apr.2022.101456, 2022.

Alvanos, M. and Christoudias, T.: GPU-accelerated atmospheric chemical kinetics in the ECHAM/MESSy (EMAC) Earth system model (version 2.52), Geosci. Model Dev., 10, 3679-3693, 10.5194/gmd-10-3679-2017, 2017.

AMD: ROCm Documentation Release 5.7.1, Advanced Micro Devices Inc., https://rocm.docs.amd.com/en/docs-5.7.1/ (last access: 26 May 2025), 2023.

Appel, K. W., Bash, J. O., Fahey, K. M., Foley, K. M., Gilliam, R. C., Hogrefe, C., Hutzell, W. T., Kang, D., Mathur, R., Murphy, B. N., Napelenok, S. L., Nolte, C. G., Pleim, J. E., Pouliot, G. A., Pye, H. O. T., Ran, L., Roselle, S. J., Sarwar, G., Schwede, D. B., Sidi, F. I., Spero, T. L., and Wong, D. C.: The Community Multiscale Air Quality (CMAQ) model versions 5.3 and 5.3.1: system updates and evaluation, Geosci. Model Dev., 14, 2867-2897, 10.5194/gmd-14-2867-2021, 2021.

Atkinson, R. W., Fuller, G. W., Anderson, H. R., Harrison, R. M., and Armstrong, B.: Urban ambient particle metrics and health: a time-series analysis, Epidemiology (Cambridge, Mass.), 21, 501-511, 10.1097/EDE.0b013e3181debc88, 2010.

Byun, D. W. and Dennis, R.: Design artifacts in eulerian air quality models: Evaluation of the effects of layer thickness and vertical profile correction on surface ozone concentrations, Atmos. Environ., 29, 105-126, https://doi.org/10.1016/1352-2310(94)00225-A, 1995.

Cao, K. and Wu, Q.: The dataset of the manuscript "Enhancing the advection module performance in the EPICC-Model V1.0 via GPU-HADVPPM4HIP V1.0 coupling and GPU-optimized

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

strategies", Zenodo [data set], https://doi.org/10.5281/zenodo.16916413, 2025.

Cao, K., Wu, Q., Wang, L., Wang, N., Cheng, H., Tang, X., Li, D., and Wang, L.: GPU-HADVPPM V1.0: a high-efficiency parallel GPU design of the piecewise parabolic method (PPM) for horizontal advection in an air quality model (CAMx V6.10), Geosci. Model Dev., 16, 4367-4383, 10.5194/gmd-16-4367-2023, 2023.

Cao, K., Wu, Q., Wang, L., Guo, H., Wang, N., Cheng, H., Tang, X., Li, D., Liu, L., Li, D., Wu, H., and Wang, L.: GPU-HADVPPM4HIP V1.0: using the heterogeneous-compute interface for portability (HIP) to speed up the piecewise parabolic method in the CAMx (v6.10) air quality model on China's domestic GPU-like accelerator, Geosci. Model Dev., 17, 6887-6901, 10.5194/gmd-17-6887-2024, 2024.

Cao, K., Tang, X., Chen, H., Ma, J., Wu, Q., Wang, W., Chen, X., Li J., Wang, Z.: Porting and Parallel Optimization of the Gas-phase Chemistry Module of the Air Quality Model EPICC-Model on China's Domestic Accelerators, Frontiers of Data&Computing, https://link.cnki.net/urlid/10.1649.TP.20250417.1428.002, 2025(in Chinese).

Chai, Z., Zhang, H., Zhang, M., Tang, X., Zheng, W., Zhu, J., Zhou, G., Cao, J., and Zeng, Q.: China's EarthLab—Forefront of Earth System Simulation Research, Adv. Atmos. Sci., 38, 1611-1620, 10.1007/s00376-021-1175-y, 2021.

Chang, J. S., Brost, R. A., Isaksen, I. S. A., Madronich, S., Middleton, P., Stockwell, W. R., and Walcek, C. J.: A three-dimensional Eulerian acid deposition model: Physical concepts and formulation, J. Geophys. Res.-Atmos., 92, 14681-14700, https://doi.org/10.1029/JD092iD12p14681, 1987.

Colella, P. and Woodward, P. R.: The Piecewise Parabolic Method (PPM) for gas-dynamical simulations, J. Comput. Phys., 54, 174-201, https://doi.org/10.1016/0021-9991(84)90143-8, 1984.

Crippa, M., Guizzardi, D., Pagani, F., Schiavina, M., Melchiorri, M., Pisoni, E., Graziosi, F., Muntean, M., Maes, J., Dijkstra, L., Van Damme, M., Clarisse, L., and Coheur, P.: Insights into the spatial distribution of global, national, and subnational greenhouse gas emissions in the Emissions Database for Global Atmospheric Research (EDGAR v8.0), Earth Syst. Sci. Data, 16, 2811-2830, 10.5194/essd-16-2811-2024, 2024.

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

Crippa, M., Guizzardi, D., Butler, T., Keating, T., Wu, R., Kaminski, J., Kuenen, J., Kurokawa, J., Chatani, S., Morikawa, T., Pouliot, G., Racine, J., Moran, M. D., Klimont, Z., Manseau, P. M., Mashayekhi, R., Henderson, B. H., Smith, S. J., Suchyta, H., Muntean, M., Solazzo, E., Banja, M., Schaaf, E., Pagani, F., Woo, J. H., Kim, J., Monforti-Ferrario, F., Pisoni, E., Zhang, J., Niemi, D., Sassi, M., Ansari, T., and Foley, K.: The HTAP_v3 emission mosaic: merging regional and global monthly emissions (2000–2018) to support air quality modelling and policies, Earth Syst. Sci. Data, 15, 2667-2694, 10.5194/essd-15-2667-2023, 2023.

Damian, V., Sandu, A., Damian, M., Potra, F., and Carmichael, G. R.: The kinetic preprocessor KPP-a software environment for solving chemical kinetics, Comput. Chem. Eng., 26, 1567-1579, https://doi.org/10.1016/S0098-1354(02)00128-X, 2002.

Elbern, H.: Parallelization and load balancing of a comprehensive atmospheric chemistry transport model, Atmos. Environ., 31, 3561-3574, https://doi.org/10.1016/S1352-2310(97)00157-X,1997.

Emery, C., Baker, K., Wilson, G., and Yarwood, G.: Comprehensive Air Quality Model with Extensions: Formulation and Evaluation for Ozone and Particulate Matter over the US, Atmosphere, 15, 1158. https://doi.org/10.3390/atmos15101158, 2024.

EPICC-Model Working Group.: Description and evaluation of the Emission and atmospheric Processes Integrated and Coupled Community (EPICC) Model version 1.0. Adv. Atmos. Sci., http://www.iapjournals.ac.cn/aas/en/article/doi/10.1007/s00376-025-4384-y, 2025.

EPICC-Model Working Group.: Source code and input data of EPICC-Model [code], EPICC-Model Working Group, https://EarthLab.iap.ac.cn/resdown/info_388.html, (last access: 26 May 2025), 2024a.

EPICC-Model Working Group.: Emission and atmospheric Processes Integrated and Coupled Community Model (EPICC-Model). Zenodo. https://doi.org/10.5281/zenodo.17071574, 2024b.

Gao, Z. and Zhou, X.: A review of the CAMx, CMAQ, WRF-Chem and NAQPMS models: Application, evaluation and uncertainty factors, Environ. Pollut., 343, 10.1016/j.envpol.2023.123183, 2024.

Georgiou, G. K., Christoudias, T., Proestos, Y., Kushta, J., Hadjinicolaou, P., and Lelieveld, J.: Air

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

quality modelling in the summer over the eastern Mediterranean using WRF-Chem: chemistry and aerosol mechanism intercomparison, Atmos. Chem. Phys., 18, 1555-1571, 10.5194/acp-18-1555-2018, 2018.

Georgiou, G. K., Christoudias, T., Proestos, Y., Kushta, J., Pikridas, M., Sciare, J., Savvides, C., and Lelieveld, J.: Evaluation of WRF-Chem model (v3.9.1.1) real-time air quality forecasts over the Eastern Mediterranean, Geosci. Model Dev., 15, 4129-4146, 10.5194/gmd-15-4129-2022, 2022.

Guevara, M., Jorba, O., Tena, C., Denier van der Gon, H., Kuenen, J., Elguindi, N., Darras, S., Granier, C., and Pérez García-Pando, C.: Copernicus Atmosphere Monitoring Service TEMPOral profiles (CAMS-TEMPO): global and European emission temporal profile maps for atmospheric chemistry modelling, Earth Syst. Sci. Data, 13, 367-404, 10.5194/essd-13-367-2021, 2021.

Gupta, M. and Mohan, M.: Validation of WRF/Chem model and sensitivity of chemical mechanisms to ozone simulation over megacity Delhi, Atmos. Environ., 122, 220-229, 10.1016/j.atmosenv.2015.09.039, 2015.

Hong, S.-Y., Noh, Y., and Dudhia, J.: A New Vertical Diffusion Package with an Explicit Treatment of Entrainment Processes, Mon. Weather Rev., 134, 2318-2341, https://doi.org/10.1175/MWR3199.1, 2006.

Intel Software: Quick Reference Guide to Optimization with Intel C++ and Fortran Compilers v19, Intel, https://www.intel.cn/content/www/cn/zh/developer/articles/technical/intel-fortran-and-c-compiler-documentation.html. (last access: 26 May 2025), 2020.

Kim, K.-H., Kabir, E., and Kabir, S.: A review on the human health impact of airborne particulate matter, Environ. Int., 74, 136-143, https://doi.org/10.1016/j.envint.2014.10.005, 2015.

Kong, L., Tang, X., Wang, Z., Zhu, J., Li, J., Wu, H., Wu, Q., Chen, H., Zhu, L., Wang, W., Liu, B., Wang, Q., Chen, D., Pan, Y., Li, J., Wu, L., and Carmichael, G. R.: Changes in air pollutant emissions in China during two clean-air action periods derived from the newly developed Inversed Emission Inventory for Chinese Air Quality (CAQIEI), Earth Syst. Sci. Data, 16, 4351-4387, 10.5194/essd-16-4351-2024, 2024.

Li, J., Wang, Z., Zhuang, G., Luo, G., Sun, Y., and Wang, Q.: Mixing of Asian mineral dust with anthropogenic pollutants over East Asia: a model case study of a super-duststorm in March 2010, Atmos. Chem. Phys., 12, 7591-7607, 10.5194/acp-12-7591-2012, 2012.

Li, J., Chen, X., Wang, Z., Du, H., Yang, W., Sun, Y., Hu, B., Li, J., Wang, W., Wang, T., Fu, P., and Huang, H.: Radiative and heterogeneous chemical effects of aerosols on ozone and inorganic aerosols over East Asia, Sci. Total Environ., 622-623, 1327-1342, https://doi.org/10.1016/j.scitotenv.2017.12.041, 2018.

Li, M., Liu, H., Geng, G., Hong, C., Liu, F., Song, Y., Tong, D., Zheng, B., Cui, H., Man, H., Zhang, Q., and He, K.: Anthropogenic emission inventories in China: a review, Natl. Sci. Rev., 4, 834-866, 10.1093/nsr/nwx150, 2017.

Linford, J. C., Michalakes, J., Vachharajani, M., and Sandu, A.: Automatic Generation of Multicore Chemical Kernels, IEEE T. Parall. Distr., 22, 119-131, 10.1109/tpds.2010.106, 2011.

Liu, J., Han, Y., Tang, X., Zhu, J., and Zhu, T.: Estimating adult mortality attributable to PM2.5 exposure in China with assimilated PM2.5 concentrations based on a ground monitoring network, Sci. Total. Environ., 568, 1253-1262, https://doi.org/10.1016/j.scitotenv.2016.05.165, 2016.

Milton, L. A. and White, A. R.: The potential impact of bushfire smoke on brain health, Neurochem. Int., 139, 104796, 10.1016/j.neuint.2020.104796, 2020.

NVIDIA: CUDA C++ Programming Guide Version 10.2, NVIDIA Corporation, https://docs.nvidia.com/cuda/archive/10.2/pdf/CUDA_C_Programming_Guide.pdf (last access: 26 May 2025), 2020.

Nenes, A., Pandis, S. N., and Pilinis, C.: ISORROPIA: A New Thermodynamic Equilibrium Model for Multiphase Multicomponent Inorganic Aerosols, Aquat. Geochem., 4, 123-152, 10.1023/A:1009604003981, 1998.

Podrascanin, Z.: Setting-up a Real-Time Air Quality Forecasting system for Serbia: a WRF-Chem feasibility study with different horizontal resolutions and emission inventories, Environ. Sci. Pollut. R., 26, 17066-17079, 10.1007/s11356-019-05140-y, 2019.

Quevedo, D., Do, K., Delic, G., Rodríguez-Borbón, J., Wong, B. M., and Ivey, C. E.: GPU Implementation of a Gas-Phase Chemistry Solver in the CMAQ Chemical Transport Model, ACS ES&T Air, 2, 226-235, 10.1021/acsestair.4c00181, 2025.

Sandu, A., Verwer, J. G., Van Loon, M., Carmichael, G. R., Potra, F. A., Dabdub, D., and Seinfeld, J. H.: Benchmarking stiff ode solvers for atmospheric chemistry problems-I. implicit vs explicit, Atmos. Environ., 31, 3151-3166, https://doi.org/10.1016/S1352-2310(97)00059-9, 1997.

Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Barker, D.M., Duda, M. G., Huang, X. Y., Wang, W., and Powers, J. G.: A Description of the Advanced Research WRF Version3 (No.NCAR/TN-475CSTR), University Corporation for Atmospheric Research, NCAR, https://doi.org/10.5065/D68S4MVH, 2008.

Sun, J., Fu, J. S., Drake, J. B., Zhu, Q., Haidar, A., Gates, M., Tomov, S., and Dongarra, J.: Computational Benefit of GPU Optimization for the Atmospheric Chemistry Modeling, J. Adv. Model. Earth. Sy., 10, 1952-1969, https://doi.org/10.1029/2018MS001276, 2018.

Tang, Y., Campbell, P. C., Lee, P., Saylor, R., Yang, F., Baker, B., Tong, D., Stein, A., Huang, J., Huang, H.-C., Pan, L., McQueen, J., Stajner, I., Tirado-Delgado, J., Jung, Y., Yang, M., Bourgeois, I., Peischl, J., Ryerson, T., Blake, D., Schwarz, J., Jimenez, J.-L., Crawford, J., Diskin, G., Moore, R., Hair, J., Huey, G., Rollins, A., Dibb, J., and Zhang, X.: Evaluation of the NAQFC driven by the NOAA Global Forecast System (version 16): comparison with the WRF-CMAQ during the summer 2019 FIREX-AQ campaign, Geosci. Model Dev., 15, 7977-7999, 10.5194/gmd-15-7977-2022, 2022.

Thompson, T. M. and Selin, N. E.: Influence of air quality model resolution on uncertainty associated with health impacts, Atmos. Chem. Phys., 12, 9753-9762, 10.5194/acp-12-9753-2012, 2012.

Top500: Supercomputing Top500 list, TOP500 international organization, https://www.top500.org/lists/top500/2024/11/ (last access: 26 May 2025), 2024.

Váňa, F., Düben, P., Lang, S., Palmer, T., Leutbecher, M., Salmond, D., and Carver, G.: Single Precision in Weather Forecasting Models: An Evaluation with the IFS, Mon. Weather Rev., 145, 495–502, https://doi.org/10.1175/mwr-d-16-0228.1, 2017.

Walcek, C. J. and Aleksic, N. M.: A simple but accurate mass conservative, peak-preserving, mixing ratio bounded advection algorithm with FORTRAN code, Atmos. Environ., 32, 3863-3880, https://doi.org/10.1016/S1352-2310(98)00099-5, 1998.

Wang, H., Chen, H., Wu, Q., Lin, J., Chen, X., Xie, X., Wang, R., Tang, X., and Wang, Z.:

GNAQPMS v1.1: accelerating the Global Nested Air Quality Prediction Modeling System (GNAQPMS) on Intel Xeon Phi processors, Geosci. Model Dev., 10, 2891-2904, 10.5194/gmd-10-2891-2017, 2017.

Wang, H., Lin, J., Wu, Q., Chen, H., Tang, X., Wang, Z., Chen, X., Cheng, H., and Wang, L.: MP CBM-Z V1.0: design for a new Carbon Bond Mechanism Z (CBM-Z) gas-phase chemical mechanism architecture for next-generation processors, Geosci. Model Dev., 12, 749-764, 10.5194/gmd-12-749-2019, 2019.

Wang, W., Chen, H., Wang, Z., Li, J., Chen, X., Yu, F., Fan, X., Zhao, S., Hu, B., Wang, W., Tang, X., Wang, Z., Ge, B., and Wu, J.: Development and evaluation of photolysis and gas-phase reaction scheme in EPICC-model: Impacts on tropospheric ozone simulation, Atmos. Environ., 359, https://doi.org/10.1016/j.atmosenv.2025.121373, 2025.

Wesely, M. L.: Parameterization of surface resistances to gaseous dry deposition in regional-scale numerical models, Atmos. Environ.(1967), 23, 1293-1304, https://doi.org/10.1016/0004-6981(89)90153-4, 1989.

Wu, H., Tang, X., Wang, Z., Wu, L., Lu, M., Wei, L., and Zhu, J.: Probabilistic Automatic Outlier Detection for Surface Air Quality Measurements from the China National Environmental Monitoring Network, Adv. Atmos. Sci., 35, 1522-1532, 10.1007/s00376-018-8067-9, 2018.

Wu, Q. Z., Xu, W. S., Shi, A. J., Li, Y. T., Zhao, X. J., Wang, Z. F., Li, J. X., and Wang, L. N.: Air quality forecast of PM_{10} in Beijing with Community Multi-scale Air Quality Modeling (CMAQ) system: emission and improvement, Geosci. Model Dev., 7, 2243-2259, 10.5194/gmd-7-2243-2014, 2014.

Yarwood, G., Shi, Y., Beardsley, R.: Impact of cb6r5 mechanism changes on air pollutant modeling in Texas. Final report prepared for the Texas Commission on Environmental Quality, Austin, TX. https://www.tceq.texas.gov/airquality/airmod/project/pj_report_pm.html, 2020.

Yang, J., Qu, Y., Chen, Y., Zhang, J., Liu, X., Niu, H., and An, J.: Dominant physical and chemical processes impacting nitrate in Shandong of the North China Plain during winter haze events, Sci. Total Environ., 912, 169065, https://doi.org/10.1016/j.scitotenv.2023.169065, 2024.

Zaveri, R. A. and Peters, L. K.: A new lumped structure photochemical mechanism for large-scale applications, J. Geophys. Res.-Atmos., 104, 30387-30415, 10.1029/1999jd900876, 1999.

https://doi.org/10.5194/egusphere-2025-2918
Preprint. Discussion started: 22 September 2025

Zhang, J., Lian, C., Wang, W., Ge, M., Guo, Y., Ran, H., Zhang, Y., Zheng, F., Fan, X., Yan, C., Daellenbach, K. R., Liu, Y., Kulmala, M., and An, J.: Amplified role of potential HONO sources in O3 formation in North China Plain during autumn haze aggravating processes, Atmos. Chem. Phys., 22, 3275-3302, 10.5194/acp-22-3275-2022, 2022.

Zhang, L., Brook, J. R., and Vet, R.: A revised parameterization for gaseous dry deposition in air-quality models, Atmos. Chem. Phys., 3, 2067-2082, 10.5194/acp-3-2067-2003, 2003.

Zhu, T., Tang, M., Gao, M., Bi, X., Cao, J., Che, H., Chen, J., Ding, A., Fu, P., Gao, J., Gao, Y., Ge, M., Ge, X., Han, Z., He, H., Huang, R.-J., Huang, X., Liao, H., Liu, C., Liu, H., Liu, J., Liu, S. C., Lu, K., Ma, Q., Nie, W., Shao, M., Song, Y., Sun, Y., Tang, X., Wang, T., Wang, T., Wang, W., Wang, X., Wang, Z., Yin, Y., Zhang, Q., Zhang, W., Zhang, Y., Zhang, Y., Zhao, Y., Zheng, M., Zhu, B., and Zhu, J.: Recent Progress in Atmospheric Chemistry Research in China: Establishing a Theoretical Framework for the "Air Pollution Complex", Adv. Atmos. Sci., 40, 1339-1361, 10.1007/s00376-023-2379-0, 2023.