GPU-accelerated atmospheric chemical kinetics using single precision in the ECHAM/MESSy (EMAC) model v2.55 with MEDINA v2.0

Kyriacos Sophocleous[1], Timo Kirfel[2], Domenico Taraborrelli[2], and Theodoros Christoudias[1]

[1]The Cyprus Institute, Nicosia, Cyprus
[2]Institute of Climate and Energy Systems, ICE-3: Troposphere, Forschungszentrum Jülich GmbH, Jülich, Germany

Correspondence: Domenico Taraborrelli (d.taraborrelli@fz-juelich.de) and Theodoros Christoudias (t.christoudias@cyi.ac.cy)

Abstract. Atmospheric chemical kinetics puts a major computational burden on Earth system models, yet its embarrassingly parallel nature of the problem structure makes it well suited to GPU acceleration. Here we present a single-precision floating point (FP32) GPU implementation of the MECCA chemical kinetics submodule in the ECHAM/MESSy Atmospheric Chemistry (EMAC) model. It is enabled by the MEDINA code-to-code compiler v2.0 and compatible with the Rosenbrock-family numerical solvers generated by the Kinetic Pre-Processor (KPP). The approach exploits FP32 arithmetic inside the GPU kernels while preserving double precision in the host EMAC workflow via explicit type conversion on the GPU.

We evaluate the numerical fidelity and achieved performance in both an idealised box-model configuration and in full EMAC production simulations on the JUWELS Booster system. Across tested mechanisms and resolutions, FP32 reproduces the double-precision CPU reference to within $< 0.004\%$ in total-species-mass differences, with discrepancies confined to chemically depleted tracers near machine representation limits. In production-scale runs, FP32 further reduces the average chemical-kinetics execution time by about 30% relative to the FP64 GPU implementation and lowers the memory footprint of FP-intensive data structures. Chemistry calculations in FP32 enhance scalability for high-resolution and chemically complex simulations.

1 Introduction

Advances in numerical models have underpinned climate and atmospheric science over recent decades, allowing better understanding of the Earth system and improved predictability under future climate change scenarios (Flato et al., 2013). Coupled Earth System Models (ESMs) integrate interacting submodels for simulating different processes of the climate system. While ESMs have become more sophisticated, with greater capabilities and performance, in large part due to the advancements in CPU clock speeds and core counts, there are still limitations in complexity and spatial resolution for climate-scale simulations. These limitations are related to the time-to-solution requirements, power consumption demands, and storage and memory availability (Palmer, 2014; Dongarra et al., 2020).

A common ESM bottleneck is the simulation of atmospheric chemical kinetics (Christou et al., 2016). The numerical integration of the stiff chemical ordinary differential equations (ODEs) remains computationally demanding despite recent optimization of the adaptive time-step control (Dreger et al., 2025). While the number of function evaluations was reduced by up to 43% for gas-phase chemistry, the overall computational time reduction was lower, at around 11%, while also requiring some adjustments to the original code.

The problem is embarrassingly parallel allowing concurrent solving over model grid partitions (Alvanos and Christoudias, 2017a). Due to this property, offloading the chemical kinetics component of ESM simulations on Graphic Processing Units (GPU) accelerators can greatly reduce the required solution time. Such developments are documented in the literature for up to 20× speedup of the chemical kinetics computational kernel when compared to the equivalent CPU implementation (Linford et al., 2017; Alvanos and Christoudias, 2019).

Architecturally, GPU processors are designed for massive parallel processing, featuring thousands of small, efficient cores optimised for handling simultaneous tasks. They utilise a Streaming Multiprocessor (SM) hierarchy, employing the Single Instruction, Multiple Threads (SIMT) model to execute data-intensive workloads. Because of their characteristics, GPUs are now widely adopted for heavy computing tasks including Machine Learning (ML), Artificial Intelligence (AI) and High Performance Computing (HPC) applications.

The main limitation of GPUs is memory bandwidth and capacity, as they face severe performance degradation when workloads are memory-bound (Williams et al., 2009; Dongarra et al., 2020; Christoudias et al., 2021). One under-exploited aspect of GPU design in this context, is that they deliver markedly higher floating point performance for 32-bit single precision (FP32) than 64-bit double precision (FP64) floating point operations, while also halving memory footprint. These benefits are especially valuable for the memory demanding, numerically intensive chemical kinetics simulations, and exploiting FP32 offers the prospect of additional speedups, reduced memory consumption, and improved energy efficiency.

In this paper, we present the development of version 2.0 of the GPU-accelerated implementation of the MECCA atmospheric chemical kinetics submodule in EMAC (MEDINA) that now supports the use of single precision arithmetic (FP32) while managing to preserve numerical stability and accuracy. We evaluate the achieved computational performance (including time-to-solution, scaling and memory usage) and numerical accuracy of the single precision implementation against standard double-precision (FP64) CPU reference.

1.1 Floating point representation

In climate-chemistry modeling, physical parameters such as species concentrations, reaction constants, and sunlight intensity are all represented as floating point numbers, and simulating chemical processes involves extensive use of floating point operations.

The IEEE 754 Standard defines floating point encoding and functionality (IEEE, 2008). The standard provides foundational formats to enable consistent calculations across platforms and sharing of floating point data while also specifies how floating point arithmetic results should be approximated. Floating point representation might have an impact on application accuracy when working with imprecise outcomes. In order to get the best performance with the precision necessary for any particular

application, it is crucial to take into account a variety of factors in a heterogeneous computing environment where numerical operations are carried out on more than one hardware architectures.

The fundamental binary floating point formats of 32 and 64 bits are called single and double precision floating point representations respectively. Under the IEEE 754 standard, double-precision (64-bit) floating-point arithmetic provides 15–17 significant decimal digits of precision and a dynamic range of approximately $10^{\pm 308}$, while single-precision (32-bit) arithmetic provides 7 significant decimal digits and a reduced dynamic range of approximately $10^{\pm 38}$ (IEEE, 2008). Double precision has been commonly the representation used in Earth System Model simulations, as it is natively supported by recent generations of CPUs and enables greater numerical stability and precision. Lately, with the advent of GPU accelerators, the use of reduced precision has been increasingly explored in geoscientific modelling, to balance the trade-off between numerical accuracy and computational cost (Higham, 2002; Clune et al., 2017; Sophocleous and Christoudias, 2022).

Even though GPUs support floating point numbers and operations in both single and double precision, they are optimised for single precision data structures arithmetic due to their historic development for graphics and very recently for machine-learning workloads. Using single precision can effectively halve the requirements compared to double precision for GPU global and cache memories, allowing more efficient data transfer over the memory hierarchy and the Arithmetic Logic Unit. Additionally, the amount of single precision cores is greater (typically double) compared to double precision cores for each multiprocessor that also contains in addition a set of 32-bit registers that are partitioned among the warps. Due to this architectural design, throughput of single precision instructions is much higher than the equivalent for double precision (Whitehead and Fit-Florea, 2017).

1.2 The MEDINA code

The ECHAM/MESSy Atmospheric Chemistry (EMAC) model is a numerical chemistry and climate simulation system that includes sub-models describing tropospheric and middle atmosphere processes and their interaction with oceans, land and human influences (Jöckel et al., 2006). It uses the second version of the Modular Earth Submodel System (MESSy2) to link multi-institutional computer codes (Jöckel et al., 2010). The core atmospheric model is the 5th generation European Centre Hamburg general circulation model (ECHAM5, Roeckner et al., 2006). The physics subroutines of the original ECHAM code have been modularized and reimplemented as MESSy submodels and have continuously been further developed. Only the spectral transform core, the flux-form semi-Lagrangian large scale advection scheme, and the nudging routines for Newtonian relaxation are remaining from ECHAM.

To compute the concentrations of and interactions between different species in the atmosphere, EMAC employs the module MECCA (Sander et al., 2019), which incorporates the Kinetic Pre-Processor (KPP) general-purpose atmospheric chemistry open-source software tool (Sandu and Sander, 2006; Damian et al., 2002). KPP numerically integrates atmospheric chemical kinetics, which are expressed as a coupled system of ordinary differential equations (ODE). KPP operates on chemical reactions written in a domain-specific language to output C and FORTRAN compatible code. The generated ODE solver enables temporal integration of the Jacobian representing the complete kinetic system.

MEDINA is a software code that automatically generates Compute Unified Device Architecture (CUDA NVIDIA, 2013) kernels to numerically integrate atmospheric chemical kinetics in the EMAC model (Alvanos and Christoudias, 2017b). A compiler parses FORTRAN code generated by the Kinetic PreProcessor (KPP) and outputs CUDA API-compatible code to be compiled with the NVIDIA compiler suite. All Rosenbrock-family solvers (Hairer and Wanner, 1991) supported by KPP are supported by MEDINA.

Performance evaluation of MEDINA v1.0, using Fermi and Pascal CUDA-enabled graphics processing unit (GPU) accelerators, shows achieved speed-ups of 4.5× and 20.4×, respectively, of the kernel execution time (Alvanos and Christoudias, 2017a). A node-to-node real-world production performance comparison shows a 1.75× speed-up over the non-accelerated model using the KPP three-stage Rosenbrock solver. In all cases, the accuracy and correctness of the accelerated implementation are evaluated by comparing to the CPU-only code of the application. The median relative difference is found to be less than 10^{-10} when comparing the output of the accelerated kernel the CPU-only code (Alvanos and Christoudias, 2019). Kerkweg et al. (2025) achieved much higher speedup (in the range 5.14× to 7.69×) for larger vectors of chemical species, resulting in greater computational load that can benefit from the capabilities of the GPUs.

As proposed in Sophocleous and Christoudias (2022), employment of single precision data structures for chemical kinetics simulations can effectively halve memory requirements without compromising the accuracy of results with respect to double precision, with maximum relative difference of $\sim 1\%$ in proof-of-concept initial investigations. Porting this concept to an architecture optimised for single precision, like GPUs, can thus be expected to improve performance in terms of time-to-solution, memory requirements and scalability.

The release of MEDINA v2.0 has added capability to output CUDA code that supports offloading of computational kernels on GPUs that integrate atmospheric chemical kinetics in single precision. This paper presents the design and development of the software code (Sec. 2.1), experimental determination of the speed-up achieved against the GPU-accelerated double precision version of the software and against the original CPU code in a box model, and validation of the accuracy of the results, and profiling of the generated GPU-kernels (Sec. 3.1). Production-ready model simulations with MEDINA v2.0 in the EMAC model are also presented (Sec. 3.2) with evaluation of accuracy, speed-up and scalability on the Juelich Supercomputing Centre Juwels Booster HPC system (Sec. 3.2.1).

2 Methodology

2.1 Implementation

In the box model configuration, the whole simulation is output in single precision. In order to enable storing of data in single precision, but also keep the option of double precision, a new data type called `REAL` is introduced. `REAL` is interpreted as "double" by default. It can also be interpreted as "float" if at compilation the flag `__SINGLEPREC` is passed:

$$REAL = \begin{cases} float, & \text{if (__SINGLEPREC) defined} \\ double, & \text{otherwise} \end{cases}$$

All floating point number data structures of the simulations have been transformed to `REAL`, thus according to the flag passed during compilation can either behave as double precision or single precision data structures during runtime. In addition, all special math functions are replaced by their single precision counterparts during the compilation.

During integration of the new version of MEDINA with the EMAC model, one of the challenges is typecasting. As reaction rates, species concentrations and other variables used by chemical kinetics integration are in double precision in the rest of the modules, single precision chemical kinetics code cannot be straightforwardly embedded in the EMAC model. Hence, the data structures have to be typecast from double to single precision before the simulation of chemical kinetics and vice versa after the completion of the operation in order to ensure that the kernels are run in single precision arithmetic, while all the other submodules operate on the data in double precision.

For this reason, two approaches have been tested:

1. Typecasting is performed on the host CPU and arguments are transferred to the device for the CUDA kernel in single precision. This implementation is optimal for data transfer on the GPU devices and can be easily unified with the EMAC model. In addition, it benefits from native single precision floating point numbers operations on the GPU, and lower memory requirements that enable better scalability and more complex and/or higher-resolution simulations. The drawback of this approach is that, due to the bulk of data required to be typecast, all the speed-up achieved by the use of single precision is contradicted by the time spent on the procedure of typecasting.

2. Offloading the chemical kinetics data in double precision to the CUDA kernel and perform the typecast to single precision within the GPU. Performing the transformation on the device allows massive parallelisation for faster typecasting by alleviating the barrier that appears when typecasting on the CPU. The main drawback of this approach is that it requires $\sim 1.5\times$ memory footprint on the device as memory is allocated for the data that is passed to the kernel in double precision, with an additional half of the total memory also required to hold the variables in single precision prior to computation. ~~The resulting species concentrations are also typecast back to double precision on the device prior to copying back to host~~Indicatively, experimenting with the box model configuration on a vector length (VL_GLO) of 57600, showed for the baseline Double Precision simulation 958MBs of Device Memory requirement. For Single Precision with host-side typecasting the requirement nearly halves to 490MBs, whereas on device-casting Single Precision the footprint is increased to 1446MBs. In this case two additional CUDA adapter functions have been implemented in the MEDINA kernel for typecasting:

   - `__global__ void doubleToFloat(float *out, double *in, int n)`
   - `__global__ void floatToDouble(double *out, float *in, int n)`

Both approaches are outlined in more detail in Fig. 1. After extensively testing the two approaches, typecasting on CPU prior to GPU kernel execution was found to require overall more execution time. While this approach is straightforward to integrate and benefits from the reduction of memory requirements on the device side, it is inferior in terms of performance. Benchmark tests have shown that although the GPU kernel execution time was faster when compared to the typecasting-on-GPU approach, any

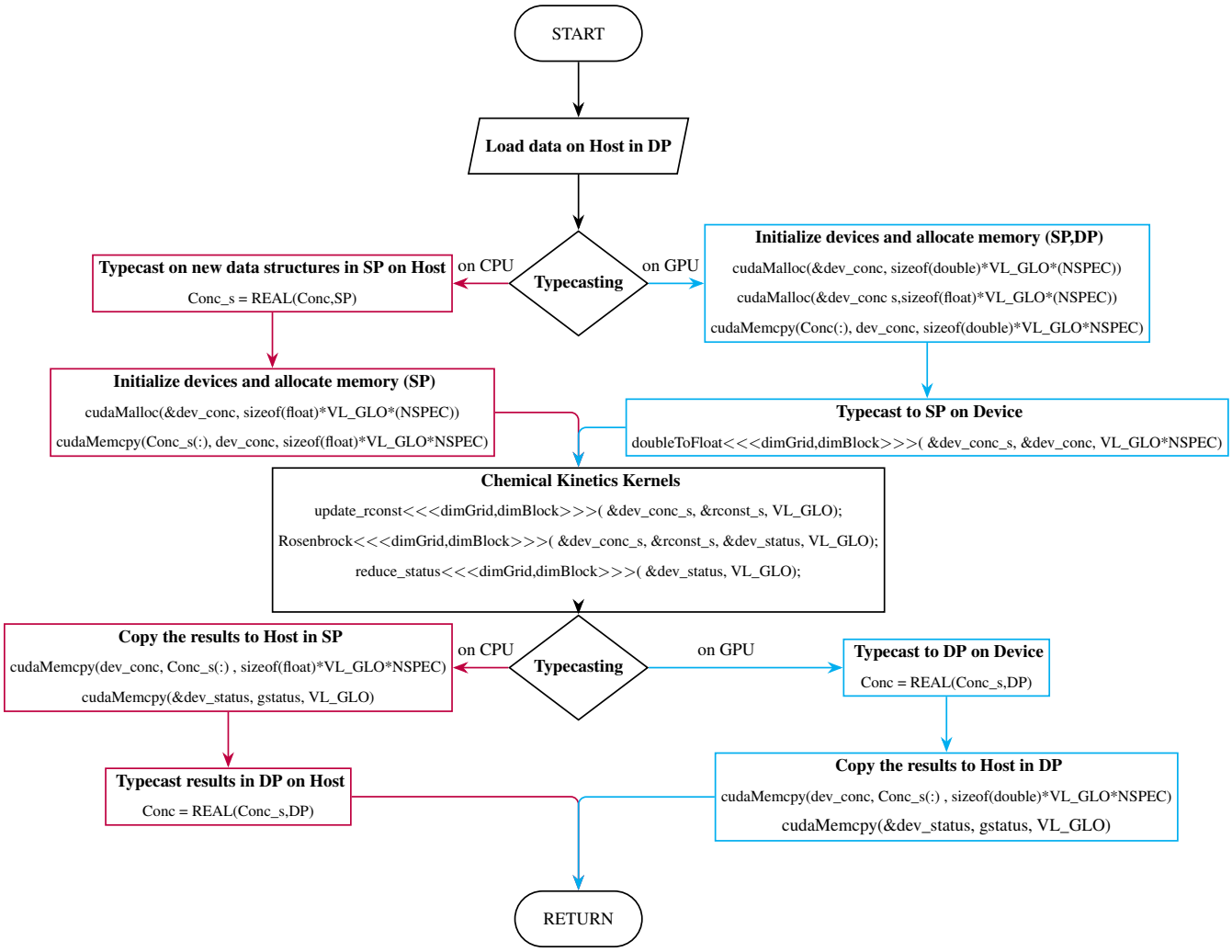


Figure 1. Single precision GPU-accelerated kernels with typecasting on CPU (Purple, Left Branch) or GPU (Cyan, Right Branch).

Table 1. Node configuration on the Juelich Supercomputing Centre Juwels Booster

CPU	Cores	RAM (GB)	TFlops	GPU	Cores	VRAM (GB)	TFlops
AMD EPYC 7402	24	512	1.075	4 × NVidia A100	4 × 6912	4 × 40	9.7 (DP) / 19.5 (SP)

speedup obtained was outweighed by the typecasting operations on the host CPU, resulting in less than 1% overall acceleration for the chemistry module when compared to pure Double Precision, and hence rendering this approach inefficient. Thus, the benchmarking results presented (Sec. 3) are based on the second approach, where typecasting operations are offloaded to the GPU and are executed massively in parallel.

2.2 Model configuration

MEDINA includes five numerical integrators commonly used in atmospheric chemical kinetics models: the two-, three- and four- stage Rosenbrock family solvers (ros2, ros3, ros4) and the four- and six- stage Rodas family solvers (rodas3 and rodas4). The software also includes a unit test with a chemical mechanism in a box model configuration. This chemical mechanism includes 74 chemical species - 73 variable species and 1 fixed, and 77 chemical reactions and is reproduced a user defined number of times (`VL_GLO`) to mimic the column vector element computational structure of an atmospheric model. The box model simulation unit test is carried out for 30 days.

For the production EMAC model runs, we applied EMAC (MESSy version 2.55.0) in the T42L90MA and T10690MA resolutions, i.e. with spherical truncations of T42 and T106 (corresponding to a quadratic Gaussian grid of approximately 2.8 and 1.1 degrees in latitude and longitude respectively) and 90 vertical hybrid pressure levels up to 0.01 hPa. The chemical mechanism used includes 360 reactions and 159 chemical species, with 156 species active and 3 fixed based on the configuration described in detail in Jöckel et al. (2016). With this configuration, simulations were run for one month (31 days). The experiments were carried out on the Juelich Supercomputing Centre Juwels Booster HPC nodes. The configuration of each Booster node is presented in Table 1.

3 Results

MEDINA has been extensively tested in terms of accuracy and performance. All the available solvers have been benchmarked for the non-accelerated CPU version and the GPU-accelerated versions of double and single precision. The box model configuration has been benchmarked on a single node of the Juwels Booster with the utilization of 1 (of the 4 available) GPU against a single CPU core. Scaling tests of the EMAC model have been conducted with up to 24 nodes with full available resources (24 CPU cores and 4 GPUs per node).

3.1 Box model

The accuracy of the results of the box model has been verified for the final concentrations of the species after 30-day simulations, by comparing the output by single precision GPU-accelerated simulations against the results of CPU-only reference simulations in double precision. Fig. 2 shows on logarithmic axes the final concentrations of all species as output by single precision (FP32) GPU-accelerated code against the double precision (FP64) ~~CPU reference~~GPU (left) and CPU reference (right). The points align along the 1:1 line across the full range spanning fifty orders of magnitude, indicating excellent agreement for both the configurations. We present here the results of the three-stage Rosenbrock solver (`Ros3`)~~, but~~ ; all the available solvers have been tested and the results are similar.

To quantify differences, we used the integrated unit test comparison post-processing routine in MEDINA, which computes, for each species, a relative difference between the GPU and CPU end-state and reports whether it exceeds a user-defined threshold. In FP64-vs-FP64 control runs, no species exceeded the nominal limit of 5%, and in fact were at 0.0004% or lower. In FP32-vs-FP64 comparisons, all species with non-negligible volume mixing ratios met the accuracy criterion, other than species that are chemically depleted and take values below the FP32 machine limit of representation. In those cases the absolute volume mixing ratio discrepancies are lower than 10^{-38} $^{-35}$ $\mathrm{molec/cm^3}$, and remain negligible compared to any atmospherically relevant value.

As these logarithmic scale scatter plots span 50 orders of magnitude, minor deviations might be obscured. Thus, two additional metrics are introduced to assess the solver's fidelity. The Normalized Mean Error (NME) 1 and the integrated bias 2. The NME measures the absolute magnitude of the deviations relative to the reference, defined as:

$$NME[\%] = \frac{\sum |x_i^{SP} - x_i^{DP}|}{\sum x_i^{DP}} \times 100 \tag{1}$$

where x_i^{SP} and x_i^{DP} represent the final concentrations (or masses) of species i for the single-precision and double-precision implementations, respectively.

The bias is used to identify any systematic over- or under-prediction, which indicates an artificially created or destroyed mass. Bias is expressed as

$$Bias[\%] = \frac{\sum (x_i^{SP} - x_i^{DP})}{\sum x_i^{DP}} \times 100 \tag{2}$$

As displayed in the two plots, the NME is exceptionally low at $1.16 \times 10^{-5}\%$, which is also complemented by a negligible integrated bias of $-3.7 \times 10^{-6}\%$, confirming that single precision manages to strictly conserve mass and does not systematically miss-predicts concentrations.

One of the main benefits of using Single Precision in GPU-accelerated code is the potential for improved time-to-solution over Double Precision. To evaluate the performance gain, we conducted a workload saturation analysis for the box model for all five implemented Rosenbrock solvers by comparing the single precision (FP32) version of the code against the previous

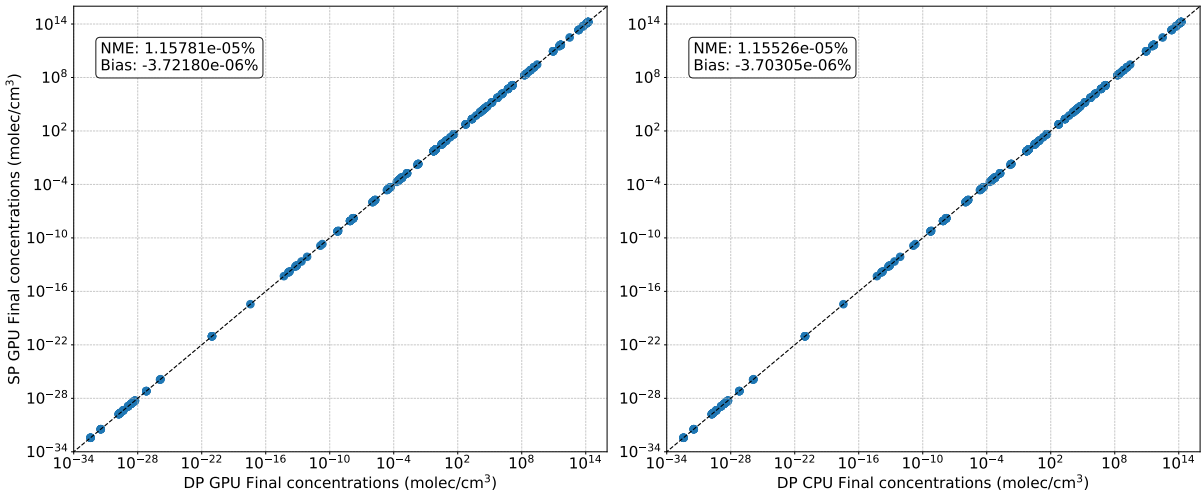


Figure 2. Final concentrations ($\mathrm{molec/cm^3}$) from the FP32 GPU implementation versus the FP64 GPU solution (left) and CPU reference (right) for all species using the three-stage Rosenbrock solver for the box model configuration.

standard FP64. For this assessment, the computational resources were fixed to a single Nvidia A100 GPU, while the size of the problem expressed as the number of independent grid boxes (VL_GLO) was increased. To measure performance, the analysis relies on the throughput (R, Eq. 3) metric, defined as the size of the problem (VL_GLO is the number of vertical model columns offloaded per kernel) over the required execution time for the size of the problem. This metric removes the trivial growth in execution time with increasing workload and provides a direct measure of weak-scaling behaviour:

$$R = \frac{VL_GLO}{T_{VL_GLO}} \tag{3}$$

Fig. 3 illustrates the solver throughput against varying vector lengths (VL_GLO).FP32 consistently achieves higher throughput than the double-precision implementation for all solvers and all size problems. For both precisions, throughput increases with the number of grid boxes, indicating improved GPU utilisation. Our analysis demonstrates that single precision GPU chemical kinetics not only manages better time-to-solution when compared to double precision, but also scales more efficiently. FP32 in MEDINA is shown to achieve better utilisation of the GPU particularly for large problems relevant for production-scale atmospheric composition simulations. Throughput effectively saturates at greater than $\sim 300,000$ cells for both representations, as the memory limit of the hardware for this mechanism has been reached.

To further complement this assessment, a weak scaling performance evaluation of MEDINA chemical kinetics integration was conducted for up to 64 GPUs (16 nodes). For this experiment, the workload per GPU was held constant at $VL_GLO = 576.000$ grid cells, while the size of the problem increases proportionally to the number of the employed GPUs. Two metrics have been utilized for this analysis: i) the processing rate (P_N) , which is defined as

$$P_N = \frac{N \times VL_GLO}{T_N}$$

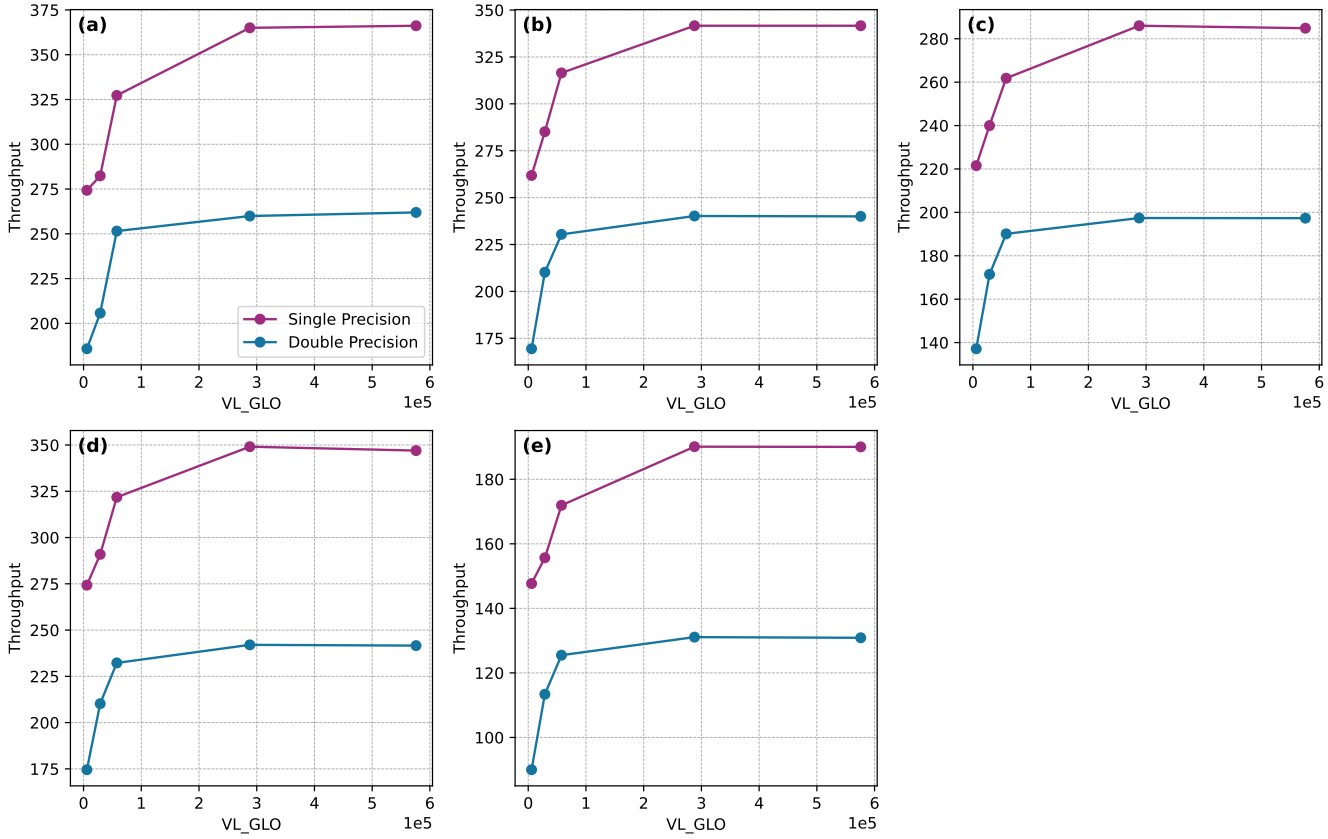


Figure 3. Throughput of GPU-accelerated chemical kinetics. VL_GLO as a function of workload size divided by execution time for the Ros2 (a), Ros3 (b), Ros4 (c), Rodas3 (d), and Rodas4 (e) solvers. Single-precision (FP32) and double-precision (FP64) GPU implementations are compared.

where N is the total number of GPUs and T_N is the required execution time, and ii) the weak scaling efficiency (E_{weak})

$$E_{weak} = \frac{T_1}{T_N} \times 100$$

where T_1 is the time-to-solution for a single GPU, and T_N the time-to-solution for N GPUs.

Figure 4 shows the weak scaling behaviour of the two representations, single- and double-precision, for the GPU implementation of the box model. The processing rate (P_N) increases with the number of GPUs, as shown in the left panel. The FP32 single precision implementation constantly sustains a higher processing rate compared to FP64 across all node configurations. The right panel of Fig. 4 presents the weak scaling efficiency (E_{weak}), with high scalability for both representations, as both manage an efficiency of over 99% for up to 64 GPUs. These results indicate that the execution time remains almost constant regardless of the total number of nodes employed, as the embarrassingly parallel nature of the problem means there is no communication overhead between GPUs.

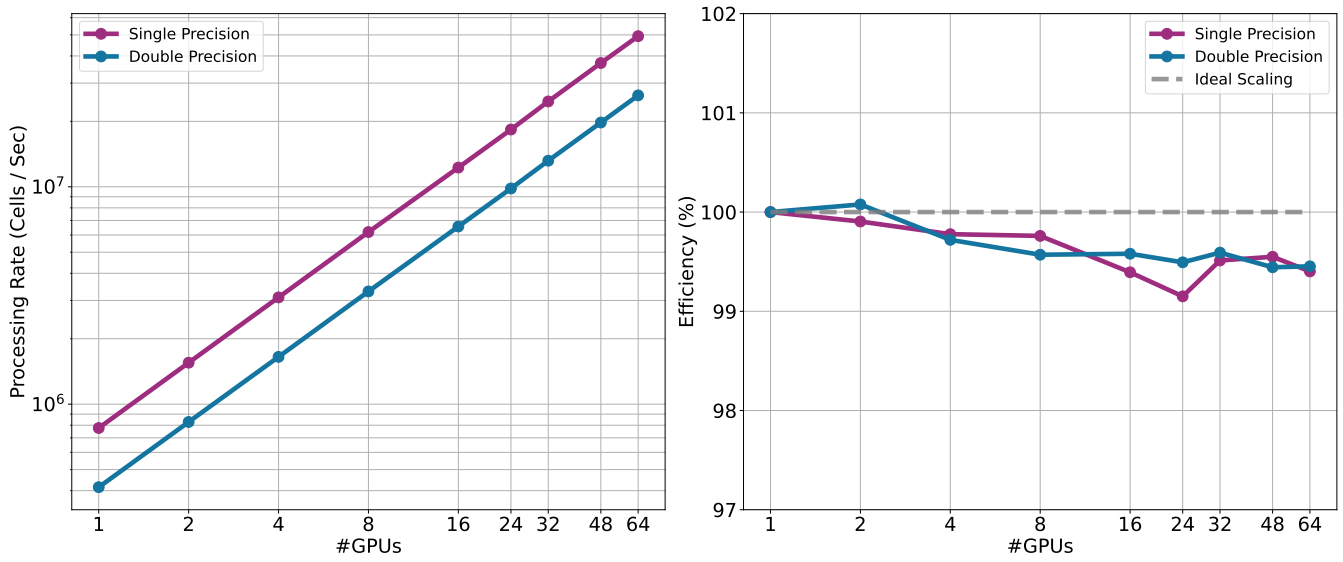


Figure 4. Weak-scaling performance of the GPU-accelerated chemical kinetics. The processing rate (left) and weak scaling efficiency (right) are shown as a function of the number of GPUs used, on a comparison of up to 64 GPUs.

We used the NVIDIA Nsight Systems (NVIDIA, 2023b) and NVIDIA Nsight Compute (NVIDIA, 2023a) performance analysis tools for understanding and optimizing the performance of applications running on NVIDIA GPUs to gain more quantitative insights on performance, and identify possible bottlenecks. The GPU-accelerated box model code has been profiled on the HPC system for all five available solvers for both the FP64 and the FP32 implementations with grid-point vector length $VL_GLO = 576,000$. As expected, most of the required simulation time ($\sim$ 99%) is spent in four procedures: allocation of memory (`cudaMalloc`), copying the data from the host to the device (`cudaMemcpy_HtoD`), integration of the sparse linear system (Rosenbrock solver), and copying the data back from the device to the host (`cudaMemcpy_DtoH`). Nsight Systems has shown that up to 94.5% of the required runtime is spent on the integration and only $\sim$5-5.5% of the time is required for the other processes. Profiling has also revealed improved kernel execution time for the Single Precision implementation compared to Double, with a speedup (Eqn. 4) of ~~42%–45%~~ $1.42\times$–$1.45\times$ depending on the selected solver (Table 2).

$$S[\%] = \frac{t_{DP} - t_{SP}}{t_{SP}} \times 100 \frac{t_{DP}}{t_{SP}} \tag{4}$$

Additionally, as shown in Tab. 3, the single precision implementation achieves a reduction in memory traffic of more than 63% in L2 cache $\leftrightarrow$ DRAM processes, more than 20% in L1 cache $\leftrightarrow$ L2 cache processes, and more than 33% in L2 stitch edge cache memory. This happens not only because memory requirements are lower for single precision, but also because hit rates for both L1/Tex and L2 Cache memories are higher for single compared to double precision (Fig. 5b).

Table 2. Required execution time for the Integrator CUDA Kernel as output by NVIDIA Nsight Compute for all five supported numerical solvers in Double (DP) and Single Precision (SP).

Solver	DP (s)	SP (s)	Speedup (~~%~~×)
Ros2	2.40	1.69	~~42.01~~ 1.420
Ros3	2.61	1.83	~~42.62~~ 1.426
Ros4	3.22	2.22	~~45.05~~ 1.451
Rodas3	2.60	1.79	~~45.25~~ 1.453
Rodas4	4.93	3.42	~~44.15~~ 1.442

Table 3. Memory transfer size between L1/L2 cache memories, L2/DRAM and L2 stitch edge cache memory in box model for Double and Single Precision using Ros3 solver.

Memory	Precision	Read (GB)	Write (GB)	Total (GB)	Reduction
L1 ↔ L2	Double	1198.08	642.29	1840.37	
	Single	1157.12	301.45	1458.57	20.75%
L2 ↔ DRAM	Double	354.94	525.37	880. 31	
	Single	181.26	142.31	323.57	63.24%
L2 stitch edge	Double			1167.36	
	Single			776.27	33.05%

Fig. 5 presents memory-related processes of GPU-accelerated chemical kinetics. For this analysis, the 3-step Rosenbrock-family (Ros3) solver was employed. The breakdown shows that the time spent in memory operations is slightly improved for FP32 and that typecasting consumes a marginal amount of the total time-to-solution (Fig. 5a). Fig. 5b presents kernel-level memory metrics as captured by NVIDIA Nsight Compute. Single Precision manages better throughput for both L1/Tex and L2 Cache memories, while hit rates are improved by more than 10%, indicating better utilisation of the GPU memory hierarchy.

While the theoretical peak performance difference between FP32 and FP64 on the NVIDIA A100 is reported as 2× roughly (Table1), the benchmarked speedup does not exceed 45% 2. KPP-generated Rosenbrock solvers do not only perform floating-point arithmetic operations. Integration of chemical kinetics relies heavily on sparse matrix operations that require extensive integer arithmetic and array indexing. Additionally, adaptive time-stepping and other control-flow tasks within the solver require logical evaluations to be managed. These computations remain non-accelerated and require the same computational effort to be performed regardless of the employed precision. This is also reflected on memory traffic behaviour (Table 3). The substantial 63.24% reduction in L2-to-DRAM traffic indicates that FP32 is a more efficient fit for the GPU fast L2 cache. However, the same number of fetch instructions are needed to load/store the data. The GPU still spends time on other processes, none of which is sped up with the utilization of FP32, so the attainable kernel acceleration remains below the theoretical 2× hardware maximum.

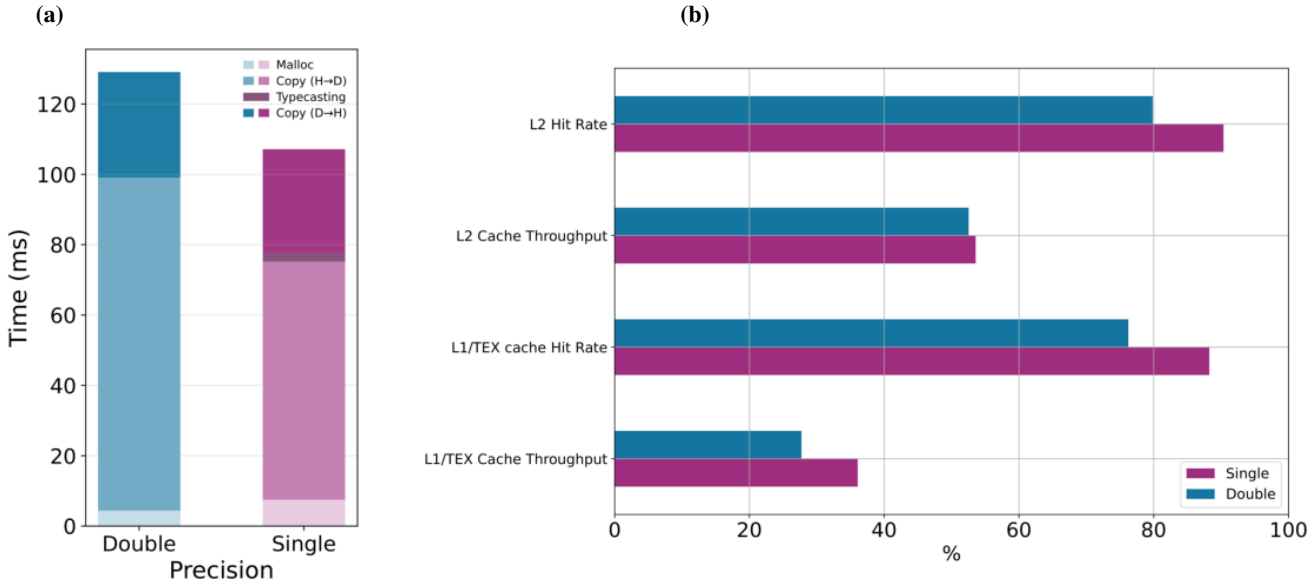


Figure 5. Profiling analysis of GPU-accelerated chemical kinetics for the Box Model using the Ros3 solver and $VL_GLO = 576,000$. (a) Time consumption of memory-related processes obtained with NVIDIA Nsight Systems, including memory allocation (malloc), host-to-device data transfer, typecasting, and device-to-host data transfer for single- (FP32) and double-precision (FP64). (b) L1/Tex Cache and L2 Cache Throughput and Hit Rates obtained with NVIDIA Nsight Compute for FP32 and FP64 implementations.

3.2 EMAC model

This section evaluates the numerical accuracy of FP32 GPU implementation within the global EMAC model against the FP64 CPU and GPU versions. Fig. 6 demonstrates the relative difference M_i in total atmospheric mass of individual chemical species between the accelerated GPU version of the code in single precision and the CPU reference version.

$$M_i[\%] = \frac{M_{CPU} - M_{GPU_32}}{M_{CPU}} \tag{5}$$

The relative difference M_i in Eq. 5 is expressed as the difference between the mass output from the FP64 CPU simulation and FP32 GPU version over the mass computed by the reference CPU version. This analysis presents the 80 species with the highest relative differences. For all species, the maximum relative difference remains below 0.004%, indicating excellent numerical agreement, well within the inherent systematic uncertainties in reaction rates. All other species with lower relative differences are presented in Appendix Fig. A1.

In Fig. 7 we compare the final mass of all species as output by Single Precision GPU version against the Double Precision GPU version using the global EMAC model in full production mode over a one-month simulation. Again, we find consistent agreement across the range of masses spanning close to fifty orders of magnitude, with values almost perfectly aligned along the 1:1 equivalence. This result demonstrates that single-precision arithmetic preserves the numerical integrity of the GPU-

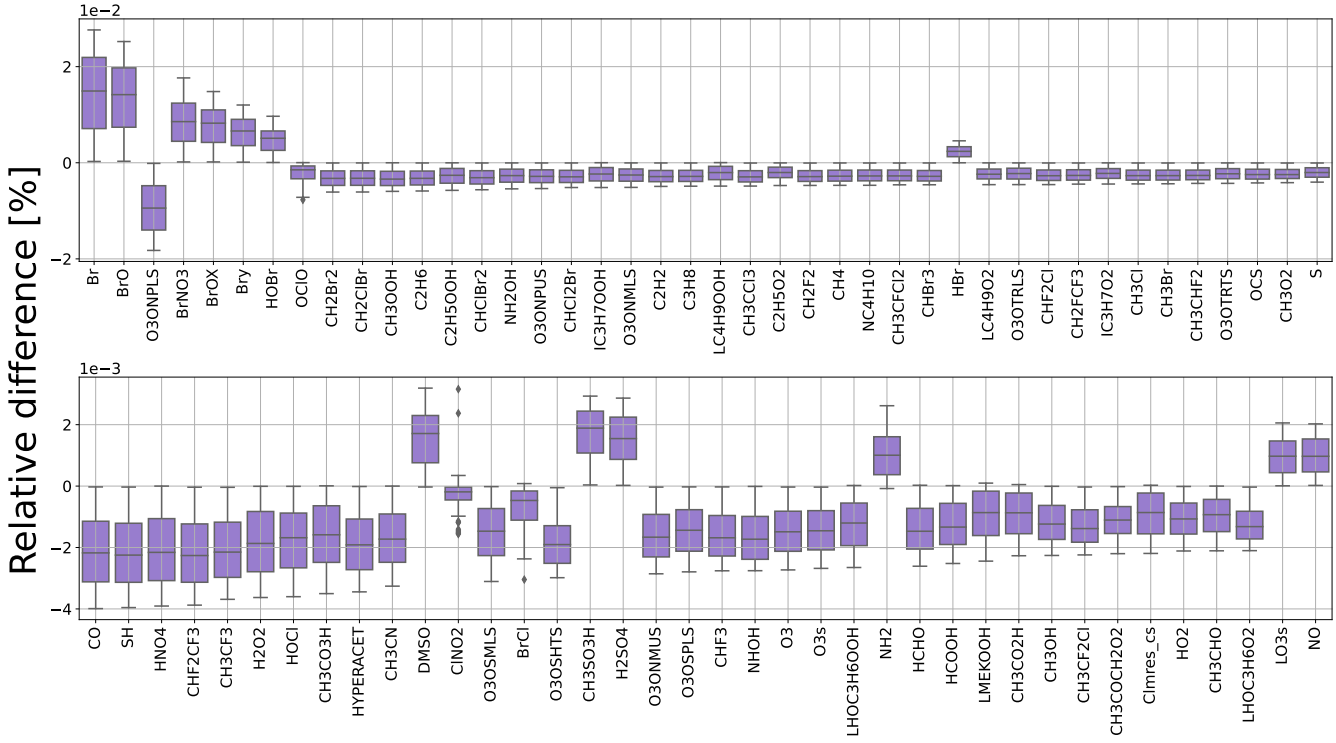


Figure 6. Relative difference (%) in total mass of species between the FP32 GPU implementation and the FP64 CPU reference in the EMAC model.

accelerated chemical kinetics within EMAC. Results are essentially identical for FP32 against FP64 on GPUs, paving the way for use in real-world production simulations.

The visual agreement across the full domain is also confirmed mathematically with the statistical deviations NME (Eq. 1) and Bias(Eq. 2). As shown within the plot, these are exceptionally small, measuring at 1.19×10^{-8} for the former and at 1.12×10^{-8} for the latter, demonstrating that no systematic mass drift or numerical divergence occurs with the employment of single precision solver for global-scale simulations over extended periods.

As demonstrated in previous studies, GPU-accelerated double precision chemistry remains numerically stable over year-long integrations (Alvanos and Christoudias, 2019). The same behaviour is expected for the single-precision implementation due to the tightly constrained error profile and the consistent mass conservation observed across the multiple diurnal cycles within the one-month simulations that have been extensively tested for the purpose of this research. Due to the significant computational demands, the extended required simulation time for year-long runs, and the recent decommissioning of the HPC cluster that was used for this research, the experimental verification of this argument is deferred to future work.

Moreover, we evaluate the spatial consistency of chemical species between the two precisions (Fig. 8). The surface concentrations of key species, the hydroxyl radical (OH), the ozone (O_3), and Nitrate (NO_3) are compared along with their relative

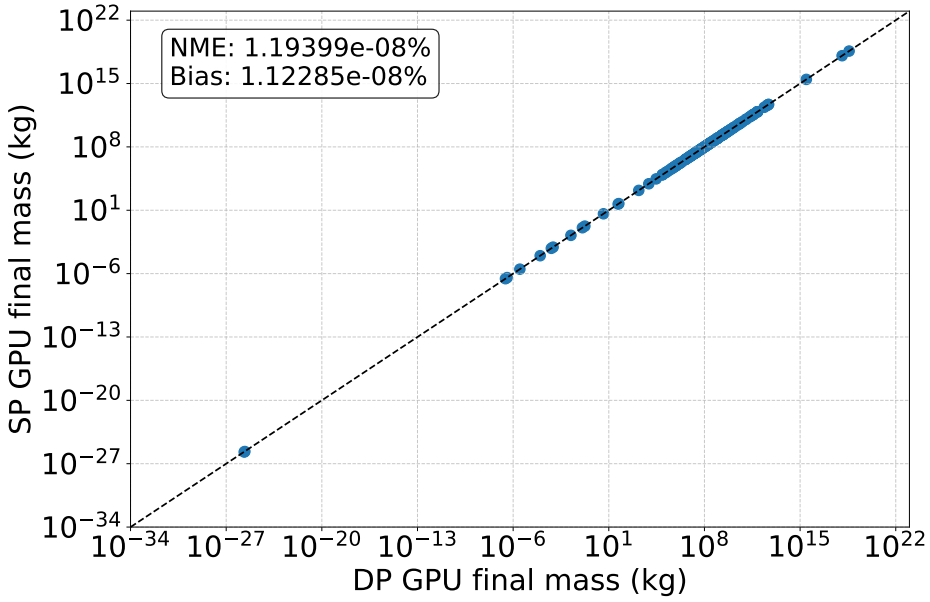


Figure 7. Final mass (kg) of species from the GPU single precision against GPU double precision simulation in the global EMAC model.

differences (Eq. 5) at the end of the simulation. The spatial patterns for these species, are in excellent agreement between the two simulations, with relative differences remaining low across the domain and consistent with the species-wise mass differences shown in Fig. 6. Elevated relative differences are confined to regions of very low concentrations, where absolute values are close to zero. These results confirm that single-precision GPU chemistry accurately reproduces the double-precision CPU solution.

3.2.1 Strong scaling

A strong-scaling assessment of MEDINA v2.0 in full production EMAC simulations was conducted on the JUWELS Booster system (Alvarez, 2021) to evaluate the scalability and time-to-solution benefits of single-precision GPU chemistry. All experiments were carried out using the following configuration on JUWELS Booster: twenty-four (24) CPU cores and four (4) GPUs per node. For T42 resolution the experiments were carried out on 1, 2, 4, 6 and 8 JUWELS Booster nodes, while for T106 1, 2, 4, 8, 12, 16, 20 and 24 nodes were used.

Fig. 9 presents the scaling of the execution time (t_s) associated with the MEDINA chemical kinetics integration for the FP32 and FP64 GPU implementations. Each violin plot represents the set of individual, per-timestep execution times accumulated over the course of the complete 31-day production run. These integration times are directly measured via the mecca_physc variable of the standard MESSy qtimer diagnostic tool, which records the wall-clock time spent in the chemical physical routine at every model time step.

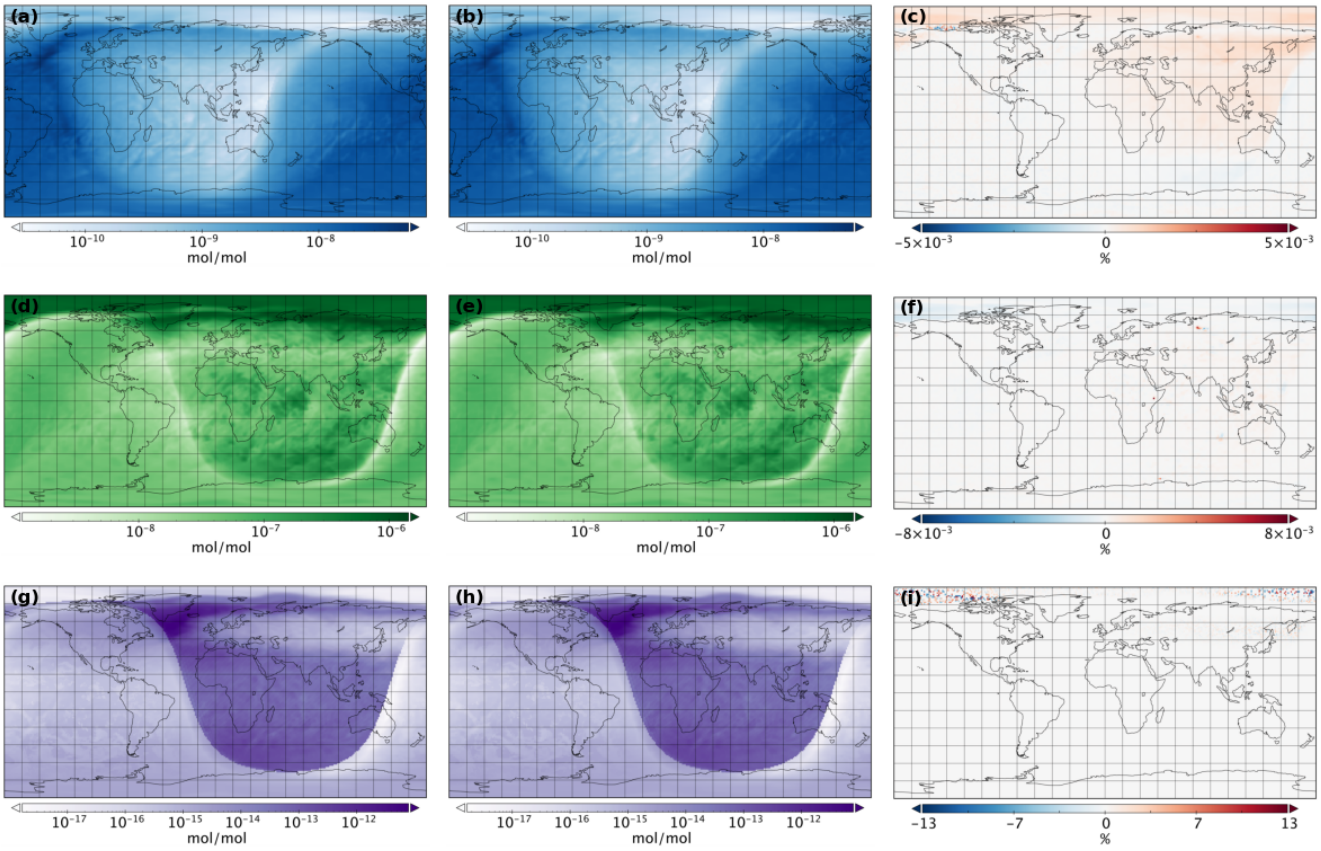


Figure 8. Surface concentrations of OH, O_3, and NO_3 for the single-precision GPU implementation ((a), (d), (g) respectively) the double-precision CPU reference ((b),(e),(h) respectively) and their relative differences ((c),(f),(i) respectively) in the EMAC model.

The left panel (Fig. 9a) corresponds to simulations at T42 horizontal resolution, while the right panel (Fig. 9b) shows the results for the finer T106 resolution. Across all configurations and for both resolutions, the execution-time distributions for the FP32 implementation are consistently reduced compared to FP64, indicating markedly faster chemical kinetics integration when Single Precision is used. In terms of the average execution time (AET, defined in Eq. 6 as the mean of the per-timestep execution times t_s over the total number of steps N_{steps}), FP32 achieves a consistent $\sim 30\%$ improvement when compared to FP64, for all numbers of Nodes/GPUs used and for both model resolutions, demonstrating robust scalability of the single-precision implementation. This performance gain derives from increased throughput coming from the GPU architecture optimisation for single-precision floating-point operations.

$$AET = \frac{\sum_{s=1}^{N_{steps}} t_s}{N_{steps}} \tag{6}$$

The overall strong-scaling performance for the chemical kinetics component of EMAC at T42 and T106 resolutions is presented in Fig. 10. The speedup (S_i) is defined in Eqn. 7 as the ratio of the time-to-solution on a given number of nodes

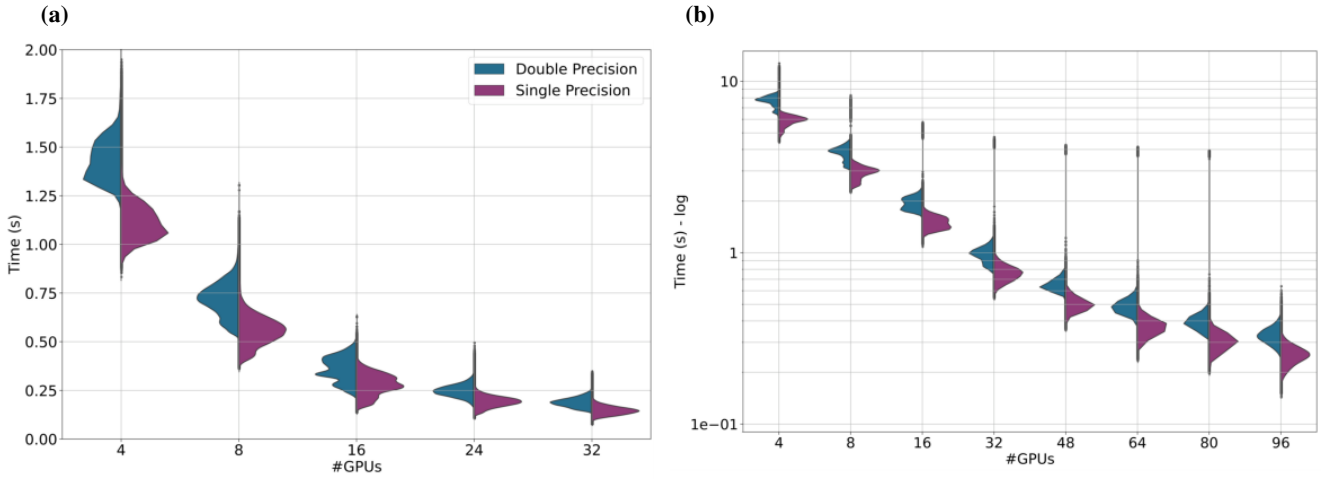


Figure 9. Distribution of execution time for the MEDINA chemical kinetics integration loop using FP32 and FP64 GPU implementations. (a) T42 resolution with experiments conducted on 1 to 8 nodes; (b) Results for T106 resolution using 1 to 24 nodes on the JUWELS system. he distributions represent the individual per-timestep execution times (t_s) recorded by the qtimer submodel (mecca_physc) accumulated over a 31-day simulation.

(t_i) to the time-to-solution of the Double Precision implementation on a single node (t_1^{DP}). For both resolutions and all Node configurations, the FP32 GPU implementation scores higher overall speedup compared to FP64, with the performance gap increasing as more resources are provided.

$$S_i = \frac{t_i}{t_1^{DP}} \tag{7}$$

To assess the GPU performance gains for the full EMAC model, we evaluated the total simulation wall-clock time. We note that there already a benefit of a factor of $times$1.75 speedup in DP (Alvanos and Christoudias, 2019), and with the additional 30% speedup in SP, chemical kinetics account for only 17.7% of the total runtime at T42 resolution and 11.3% at T106 – from up to 90% on CPU-only configurations (Christou et al., 2016). The approximate 30% performance gain in SP from MEDINA yields substantial reduction in the total time-to-solution of 252/25.4 to 9.8/6.2 minutes in T106/T42 respectively. Fig. 11 shows the total wall-clock time in minutes for one-month simulation in T42 and T106 resolutions, separating the time spent in the accelerated chemistry solver from the rest of the EMAC runtime. For T42 resolution, the employment of Single precision saves an additional 25.4 minutes of total execution time on 4 GPUs, and maintains proportionally higher time saving with more nodes, as the rest of the model scales on the CPU. In the computationally more demanding T106 resolution, over 4 hours (250 minutes) are saved off the total simulation wall-clock time for 4 GPUs. As the model scales on multiple nodes, the 30% gain in chemical kinetics compared to DP is maintained, resulting in significant speedups with the scalability for the rest of the EMAC components on CPU. A detailed analysis of the substantial speedup already achieved in DP is presented in (Alvanos and Christoudias, 2019). For chemistry schemes that require even larger fraction of the total EMAC runtime, for instance in Pozzer et al. (2022), the overall wall-clock improvement is expected to be even larger.

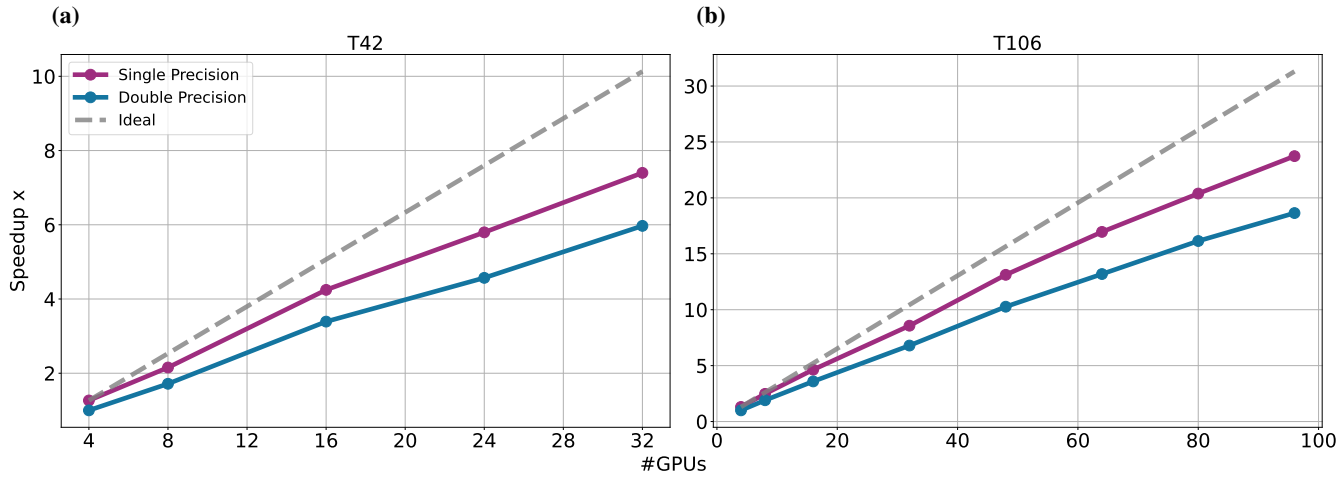


Figure 10. Strong-scaling speedup of the EMAC model on JUWELS for T42 (left) and T106 (right) resolutions, comparing FP64 and FP32 GPU chemical kinetics, against the number of GPUs used.

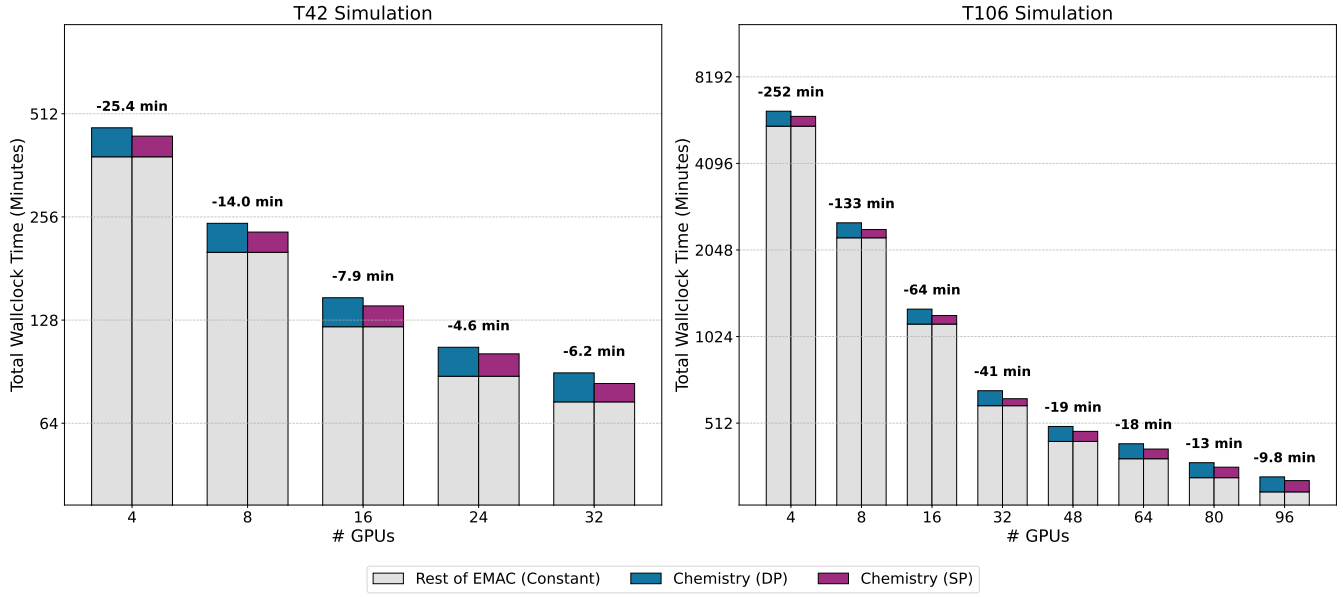


Figure 11. Total wall-clock time required for the global EMAC model simulations. The time consumption is broken down into the constant non-chemistry components and the chemical kinetics integration for both the single-precision (FP32) and double-precision (FP64) GPU implementations. Results are shown for the T42 (left) and T106 (right) resolutions as a function of the number of GPUs used, with the absolute execution time savings indicated above each configuration

Our results demonstrate that single-precision GPU chemical kinetics significantly improves both the performance and scalability of the EMAC ESM. The benefits are sustained for both resolutions and all node configurations, making FP32 GPU chemistry with MEDINA v.2.0 an effective approach for accelerating large-scale Earth system simulations without compromising numerical accuracy.

4 Conclusions

This paper presents the development of MEDINA v2.0, with the added feature to produce single-precision (FP32) CUDA kernels for atmospheric chemical kinetics ready to be integrated in the ECHAM/MESSy (EMAC) Earth system model. We evaluate the single-precision GPU implementation of the MECCA chemical kinetics submodule and demonstrate that it can reliably replace the FP64 GPU reference, delivering significant performance improvements while preserving numerical accuracy.

Accuracy assessments, for both the box-model configuration and full production-grade EMAC simulations, have shown excellent agreement between the newly developed FP32 GPU version of the code and the FP64 CPU and GPU references. Differences in species average mixing rations and total atmospheric mass remain below 0.004% in all cases, with larger relative discrepancies to be found for species that are chemically depleted at the machine limit.

In terms of performance, the FP32 GPU achieves better speedup, scalability, and memory utilization compared to its FP64 counterpart. For the box-model configuration, single precision consistently achieves lower normalised execution times for all Rosenbrock-family solvers and problem sizes. By utilising NVidia profilers (Nvidia Nsight Compute and Nvidia Nsight Systems) we have shown that FP32 performes better in memory operations and cache efficiency, with higher throughput and hit rates for both L1/Tex and L2 cache memories. Although GPU typecasting introduces additional overhead, the enabled parallel execution manages to keep the required execution time low, preserving the gains of kernel execution in single precision.

In production-scale EMAC simulations on the JUWELS Booster system, the FP32 implementation achieves an average performance gain of approximately 30% in chemical-kinetics execution time relative to the FP64 GPU version, for both T42 and T106 resolutions and for all node configurations. Strong-scaling experiments have demonstrated that the acceleration obtained is consistent and increases with system size, underlining the suitability of single-precision GPU chemistry for large-scale, high resolution simulations.

Medina v2.0 has exhibited great performance gains. However, the implementation currently only generates CUDA kernels and hence, only NVIDIA GPUs are supported. These devices were specifically targeted for this development due to the MEDINA legacy code that allowed for quick refactoring and development, and the availability of NVIDIA devices on the JUWELS booster supercomputer. As the HPC and Earth System Modelling community adapts different heterogenous architectures featuring AMD and Intel accelerators, portability is becoming a critical requirement. In the future, MEDINA, as an automated code-to-code compiler, can potentially address this limitation, by extending the framework to other programming models and open standards. This would ensure that the accelerated chemical kinetics simulations are vendor-agnostic and portable across different supercomputing environments and heterogenous architectures.

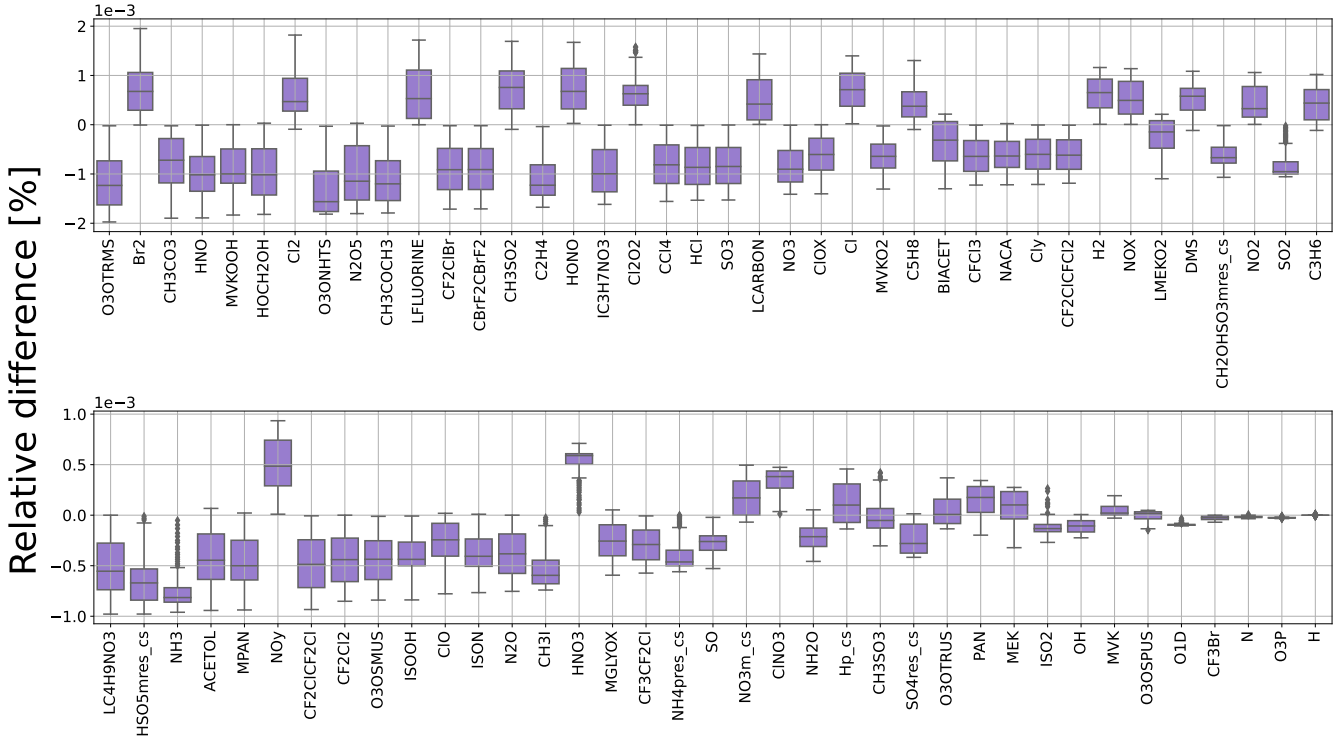


Figure A1. Relative difference (%) in total mass of additional species between the FP32 GPU implementation and the FP64 CPU reference in the EMAC model.

As Earth system models rely on heterogeneous GPU-accelerated architectures, reduction in computational cost and performance gain without accuracy compromising is essential. The presented approach demonstrates that single-precision GPU-accelerated chemical kinetics can be safely integrated into EMAC and provide a framework with better time-to-solution and memory management when compared to the nowadays standard Double Precision, improving efficiency and scalability of computationally intensive model components.

Appendix A: Additional Results

Code and data availability. The MEDINA FORTRAN to CUDA code-to-code compiler is developed in CUDA and Python and is included in the EMAC model. A consortium of institutions continuously develops the EMAC model. The usage of EMAC and access to the source code are licensed to all affiliates of institutions which are members of the MESSy Consortium. Institutions can become a member of the MESSy Consortium by signing the MESSy Memorandum of Understanding. More information can be found on the MESSy Consortium website (http://www.messy-interface.org). The code presented here was developed based on MESSy version 2.55.0 and will be available in the next official release (version 2.56). The exact version of the MEDINA code used to produce the results used in this paper is archived on the Zenodo repository under under the MIT lisence; DOI: https://doi.org/10.5281/zenodo.19882619 (Christoudias, 2026). All data sources for the results presented are available with DOI:10.5281/zenodo.21442083 (Christoudias and Sophocleous, 2026).

Author contributions. Software, Investigation, Formal Analysis, Visualization: K.S. Writing - Original Draft: K.S. Writing - Review & Editing: T.C., D.T., T.K. Resources: T.C., D.T. Conceptualization, Methodology, Funding acquisition, Supervision: T.C.

Competing interests. The authors declare that they have no competing interests.

Disclaimer. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

Acknowledgements. This study has been partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement no. 856612 (EMME-CARE) and no. 857645 (NI4OS). The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS (Alvarez, 2021) at Jülich Supercomputing Centre (JSC).

References

Alvanos, M. and Christoudias, T.: GPU-accelerated atmospheric chemical kinetics in the ECHAM/MESSy (EMAC) Earth system model (version 2.52), Geoscientific Model Development, 10, 3679–3693, https://doi.org/10.5194/gmd-10-3679-2017, 2017a.

Alvanos, M. and Christoudias, T.: MEDINA: MECCA Development in Accelerators – KPP Fortran to CUDA Source-to-Source Preprocessor, Journal of Open Research Software, 5, 13, https://doi.org/10.5334/jors.158, 2017b.

Alvanos, M. and Christoudias, T.: Accelerating Atmospheric Chemical Kinetics for Climate Simulations, IEEE Transactions on Parallel and Distributed Systems, 30, 2396–2407, 2019.

Alvarez, D.: JUWELS cluster and booster: exascale pathfinder with modular supercomputing architecture at juelich supercomputing centre, Journal of large-scale research facilities JLSRF, 7, A183–A183, 2021.

Christou, M., Christoudias, T., Morillo, J., Alvarez, D., and Merx, H.: Earth system modelling on system-level heterogeneous architectures: EMAC (version 2.42) on the Dynamical Exascale Entry Platform (DEEP), Geoscientific Model Development, 9, 3483–3491, https://doi.org/10.5194/gmd-9-3483-2016, 2016.

Christoudias, T.: MEDINA v.2, https://doi.org/10.5281/zenodo.19882619, 2026.

Christoudias, T. and Sophocleous, K.: GPU-accelerated atmospheric chemical kinetics using single precision in the ECHAM/MESSy (EMAC) model v2.55 with MEDINA v2.0: Data sources, https://doi.org/10.5281/zenodo.21442083, 2026.

Christoudias, T., Kirfel, T., Kerkweg, A., Taraborrelli, D., Moulard, G.-E., Raffin, E., Azizi, V., Oord, G. v. d., and Werkhoven, B. v.: GPU Optimizations for Atmospheric Chemical Kinetics, in: The International Conference on High Performance Computing in Asia-Pacific Region, pp. 136–138, 2021.

Clune, T. L., Dennis, J. M., Edwards, J., and Vertenstein, M.: Computational and scientific challenges in global climate modeling, Bulletin of the American Meteorological Society, 98, 1531–1544, https://doi.org/10.1175/BAMS-D-16-0014.1, 2017.

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

Dongarra, J., Hittinger, J., Bell, J., Chacon, L., Falgout, R., Heroux, M., Hovland, P., Ng, E., Webster, C., and Wild, S.: Applied mathematics research for exascale computing, SIAM Review, 62, 3–64, https://doi.org/10.1137/18M1191022, 2020.

Dreger, R., Kirfel, T., Pozzer, A., Rosanka, S., Sander, R., and Taraborrelli, D.: Optimized step size control within the Rosenbrock solvers for stiff chemical ordinary differential equation systems in KPP version 2.2.3_rs4, Geoscientific Model Development, 18, https://doi.org/10.5194/gmd-18-4273-2025, 2025.

Flato, G., Marotzke, J., Abiodun, B., Braconnot, P., Chou, S. C., Collins, W., Cox, P., Driouech, F., Emori, S., Eyring, V., et al.: Evaluation of climate models, Climate Change 2013: The Physical Science Basis, pp. 741–866, 2013.

Hairer, E. and Wanner, G.: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Springer, Berlin, 1991.

Higham, N. J.: Accuracy and Stability of Numerical Algorithms, SIAM, 2002.

IEEE: IEEE Standard for Floating-Point Arithmetic, Tech. rep., IEEE, Piscataway, NJ, USA, https://doi.org/10.1109/IEEESTD.2008.4610935, 2008.

Jöckel, P., Tost, H., Pozzer, A., Brühl, C., Buchholz, J., Ganzeveld, L., Hoor, P., Kerkweg, A., Lawrence, M. G., and Sander, R.: The atmospheric chemistry general circulation model ECHAM5/MESSy1: Consistent simulation of ozone from the surface to the mesosphere, Atmospheric Chemistry and Physics, 6, 5067–5104, https://doi.org/10.5194/acp-6-5067-2006, 2006.

Jöckel, P., Kerkweg, A., Pozzer, A., Sander, R., Tost, H., Riede, H., Baumgaertner, A., Gromov, S., and Kern, B.: Development cycle 2 of the Modular Earth Submodel System (MESSy2), Geoscientific Model Development, 3, 717–752, https://doi.org/10.5194/gmd-3-717-2010, 2010.

Jöckel, P., Tost, H., Pozzer, A., Kunze, M., Kirner, O., Brenninkmeijer, C. A., Brinkop, S., Cai, D. S., Dyroff, C., Eckstein, J., et al.: Earth system chemistry integrated modelling (ESCiMo) with the modular earth submodel system (MESSy) version 2.51, Geoscientific Model Development, 9, 1153–1200, 2016.

Kerkweg, A., Kirfel, T., Do, D. H., Griessbach, S., Jöckel, P., and Taraborrelli, D.: The MESSy DWARF (based on MESSy v2.55.2), Geoscientific Model Development, 18, 1265–1286, https://doi.org/10.5194/gmd-18-1265-2025, 2025.

Linford, J. C., Sandu, A., and Carmichael, G. R.: Accelerated chemical kinetics simulations using GPUs, Atmospheric Environment, 163, 37–49, https://doi.org/10.1016/j.atmosenv.2017.05.030, 2017.

NVIDIA: CUDA C Programming Guide, 2013.

NVIDIA: NVIDIA Nsight Compute User Guide, https://docs.nvidia.com/nsight-compute/2023.1/NsightCompute/index.html, 2023a.

NVIDIA: NVIDIA Nsight Systems User Guide, https://docs.nvidia.com/nsight-systems/UserGuide/, 2023b.

Palmer, T. N.: More reliable forecasts with less precise computations: A fast-track route to cloud-resolved weather and climate simulators?, Philosophical Transactions of the Royal Society A, 372, https://doi.org/10.1098/rsta.2013.0391, 2014.

Pozzer, A., Reifenberg, S. F., Kumar, V., Franco, B., Kohl, M., Taraborrelli, D., Gromov, S., Ehrhart, S., Jöckel, P., Sander, R., Fall, V., Rosanka, S., Karydis, V., Akritidis, D., Emmerichs, T., Crippa, M., Guizzardi, D., Kaiser, J. W., Clarisse, L., Kiendler-Scharr, A., Tost, H., and Tsimpidi, A.: Simulation of organics in the atmosphere: evaluation of EMACv2.54 with the Mainz Organic Mechanism (MOM) coupled to the ORACLE (v1.0) submodel, Geoscientific Model Development, 15, 2673–2710, https://doi.org/10.5194/gmd-15-2673-2022, 2022.

Roeckner, E., Brokopf, R., Esch, M., Giorgetta, M., Hagemann, S., Kornblueh, L., Manzini, E., Schlese, U., and Schulzweida, U.: Sensitivity of simulated climate to horizontal and vertical resolution in the ECHAM5 atmosphere model, Journal of Climate, 19, 3771–3791, 2006.

Sander, R., Baumgaertner, A., Cabrera-Perez, D., Frank, F., Gromov, S., Grooß, J.-U., Harder, H., Huijnen, V., Jöckel, P., Karydis, V. A., Niemeyer, K. E., Pozzer, A., Riede, H., Schultz, M. G., Taraborrelli, D., and Tauer, S.: The community atmospheric chemistry box model CAABA/MECCA-4.0, Geoscientific Model Development, 12, https://doi.org/10.5194/gmd-12-1365-2019, 2019.

Sandu, A. and Sander, R.: Technical note: Simulating chemical systems in Fortran90 and Matlab with the Kinetic PreProcessor KPP-2.1, Atmospheric Chemistry and Physics, 6, 187–195, https://doi.org/10.5194/acp-6-187-2006, 2006.

Sophocleous, K. and Christoudias, T.: Reduced-Precision Chemical Kinetics in Atmospheric Models, Atmosphere, 13, 1418, 2022.

Whitehead, N. and Fit-Florea, A.: Floating Point and IEEE-754 Compliance for NVIDIA GPUs, Tech. rep., NVIDIA, 2017.

Williams, S., Waterman, A., and Patterson, D.: Roofline: An insightful visual performance model for multicore architectures, Communications of the ACM, 52, 65–76, https://doi.org/10.1145/1498765.1498785, 2009.