



Enhancing single precision with quasi-double precision: achieving double-precision accuracy in the Model for Prediction Across Scales – Atmosphere (MPAS-A) version 8.2.1

Jiayi Lai¹, Lanning Wang^{1,2}, Qizhong Wu^{1,2}, Yizhou Yang³, and Fang Wang⁴

¹College of Global Change and Earth System Science, Faculty of Geographical Science, Beijing Normal University, Beijing 100875, China

²Joint Center for Earth System Modeling and High-Performance Computing, Beijing Normal University, Beijing 100875, China

³National Supercomputing Center, Wuxi 214026, China

⁴CMA Earth System Modeling and Prediction Centre (CEMC), Beijing 100081, China

Correspondence: Lanning Wang (wangln@bnu.edu.cn) and Yizhou Yang (yang.yizhou@outlook.com)

Received: 24 September 2024 – Discussion started: 27 September 2024

Revised: 31 December 2024 – Accepted: 3 January 2025 – Published: 25 February 2025

Abstract. The limitations of high-performance computing (HPC) significantly constrain the development of numerical models. Traditional numerical models often employ double precision to ensure result accuracy, but this comes at a high computational cost. While using lower precision can substantially reduce computational expenses, it may introduce round-off errors that can affect accuracy under certain conditions. The quasi-double-precision algorithm (QDP algorithm) compensates for these round-off errors by maintaining corrections, thus improving result accuracy. To investigate the effectiveness of this algorithm at enhancing the accuracy of numerical model results, this paper applies it to the single-precision version of the Model for Prediction Across Scales – Atmosphere (MPAS-A), and its performance is evaluated across two idealized and two real-data cases. The results show that the application of the QDP algorithm reduces the surface pressure bias by 68 %, 75 %, 97 %, and 96 % in the respective cases. Compared to double-precision experiments, the runtime is reduced by 28.6 %, 28.5 %, 21.1 %, and 5.7 %. This study demonstrates that the QDP algorithm provides effective and cost-efficient computational capabilities for numerical models.

1 Introduction

Since the advent of modern computers in the 1950s, numerical-simulation-based weather and climate modeling has emerged as one of the most effective methods for exploring weather and climate systems, providing a new platform for numerical model research (Bauer et al., 2015). However, in order to achieve more accurate and precise simulation results, numerical weather and climate models are evolving towards higher resolutions and more complex physical parameterization schemes (Bauer et al., 2015). With the integration of increasingly complex modules to meet diverse requirements, numerical weather and climate models have developed rapidly, and the next generation of these models will feature unprecedented resolution and complexity (Hatfield et al., 2019). In light of these circumstances, the demand for more powerful high-performance computing (HPC) systems and more efficient computational methods has become particularly urgent. As noted by Bauer et al. (2015), the computational tasks of future numerical model prediction (NMP) systems are expected to be 100 to 1000 times greater than those of 2015's systems. To bridge the gap between hardware advancements and application performance, the design of code and the selection of algorithms must focus on the optimization of floating-point operations and memory usage (Hatfield et al., 2019).

Mixed precision is a promising research direction in optimizing computational resources within numerical models. By reducing the bit width required for number representation and thereby lowering the precision of floating-point numbers, mixed-precision methods enable storage and computations to be performed with fewer bits. This approach reduces the computational and communication costs in numerical simulations such as climate modeling. Lower-precision numerical representations are a feasible approach to reducing computational costs in complex numerical models (Dawson and Düben, 2017). Low-precision computations, defined as operations utilizing fewer than 64 bits of significance, remarkably decrease resource requirements but may introduce round-off errors. To address this challenge, the study of mixed-precision techniques has emerged.

In recent years, notable advancements have been made in the application of mixed-precision computing in numerical weather and climate models. Váňa et al. (2017) investigated the implementation of mixed-precision computing in the Integrated Forecast System (IFS) prediction model. They employed double precision in certain regions while utilizing lower precision in others. This approach significantly enhanced computational efficiency by an average of 40 % while maintaining acceptable error margins. Dawson et al. (2018) expanded the scope of mixed-precision methods, demonstrating their applicability to simple thermal diffusion models, while key state variables are stored and updated with higher precision. For more complex real-world land surface schemes, they showed that using lower precision for the majority of computations while ensuring high-precision processing of state variables could still meet the requisite accuracy standards. Concurrently, Nakano et al. (2018) conducted an in-depth study on the dynamical core of the global compressible nonhydrostatic model, particularly the baroclinic wave tests by Jablonowski and Williamson. Nakano et al. (2018) opted to use double precision for grid geometry calculations and single precision for other components. The results indicated that this strategy not only successfully simulated the growth of baroclinic waves with minimal error but also reduced runtime by 46 %. This study further corroborated the efficacy of mixed-precision computing in dynamical core calculations. Hatfield et al. (2019) applied mixed-precision computing to the Legendre transform in the IFS, successfully implementing half-precision computations. Remarkably, this modification reduced the computational cost to 25 % of that in the double-precision reference test, significantly lowering computational overhead. This achievement underscored the substantial potential of mixed-precision computing in large-scale numerical prediction models. Tintó Prims et al. (2019) applied mixed-precision methods to the Nucleus for European Modelling of the Ocean (NEMO). They discovered that 95.8 % of the 962 variables could be computed using single precision. Additionally, in the Regional Ocean Modeling System (ROMS), all 1146 variables could be computed using single precision, with 80.7 % of them even using half

precision. This finding suggests that mixed-precision methods have extensive applicability in ocean modeling. Cotronei and Slawig (2020) converted the radiation component of the atmosphere in the European Centre Hamburg model (ECHAM) to a single-precision algorithm, resulting in an approximately 40 % acceleration in radiation calculations. This result indicates that applying single-precision computing in atmospheric models can significantly enhance computational efficiency while preserving computational accuracy to a reasonable extent. Paxton et al. (2022) further investigated the feasibility of reduced-precision computing, conducting tests in the Lorenz system, shallow water approximation over a ridge, and the simplified parameterized coarse-resolution spectral global atmospheric model SPEEDY. The findings revealed that single precision sufficed for most computational needs, and in numerous cases, half precision was also able to achieve the desired results. This provides an important reference for adopting lower-precision computing in various models in the future. In 2024, Banderier et al. (2024) further substantiated the effectiveness of mixed-precision methods in the regional weather and climate model COSMO. The study found that the differences between double-precision and single-precision simulations were minimal, typically detectable only in the initial few hours or days of the simulation. However, single-precision simulations reduced computational costs by approximately 30 %. In the same year, Chen et al. (2024) applied the principle of limited iterative development to identify equations that were insensitive to precision in weather and climate modeling tests, modifying them from double precision to single precision. This optimization resulted in a reduction in the runtime of the model's hydrostatic solver, nonhydrostatic solver, and tracer transport solver by 24 %, 27 %, and 44 %, respectively, thereby substantially enhancing computational efficiency. In summary, mixed-precision computing exhibits broad application prospects and potential advantages in numerical weather and climate modeling. By flexibly applying varying precision computing methods while ensuring predictive accuracy, it is feasible to significantly enhance computational efficiency and reduce computational costs.

When utilizing mixed-precision computation, low-precision calculations inevitably introduce round-off errors, particularly when adding numbers with significantly different magnitudes. In such scenarios, the limited precision can cause the larger number to effectively “swallow” the smaller number, thereby compromising the accuracy of the result. For instance, consider the variables $A = 0.7315 \times 10^3$ (a large number) and $B = 0.4506 \times 10^{-5}$ (a small number). If the precision of the result is reduced to 4 significant digits, the outcome will be 0.7315×10^3 , with the large number effectively overshadowing the small one. This phenomenon is especially pertinent in numerical modeling, where the introduction of biases into fundamental fields often necessitates the addition of large and small numbers, inherently causing round-off errors. These errors can accumulate over

successive computations, leading to a degradation in model accuracy or even complete failure. Therefore, this issue cannot be overlooked.

Some methods have been developed to address the round-off errors. In an early study, Gill (1951) proposed a fourth-order, four-step explicit Runge–Kutta method aimed at correcting round-off errors during computation. This method constructs auxiliary variables at each step to compensate for the round-off errors generated, thereby further refining the results to achieve higher precision. However, this method is not applicable to other forms of numerical solutions. In addition to this, compensated summation methods can enhance the accuracy of summation by utilizing the floating-point precision supported by lower-level hardware (Higham, 2002). These methods rely on recursive summation and incorporate correction terms to reduce round-off errors. Subsequently, Møller (1965) and Kahan (1965) proposed the quasi-double-precision (QDP) algorithm and the Kahan algorithm, respectively. The primary idea behind both methods is to make slight adjustments to the total sum to avoid the precision loss caused by adding a small, precise value to a much larger one in floating-point addition. The QDP algorithm has been validated in solving ordinary differential equations using the fourth-order Runge–Kutta method (Møller, 1965), where the error after precision reduction is essentially minimized to zero.

Currently, methods for compensating round-off errors are primarily employed in the step-by-step integration of ordinary differential equations (Thompson, 1970; Tomonori and Hideko, 1995; Dmitruk and Stpiczyński, 2023). However, their validation in numerical models remains uncertain. Considering the broader applicability of the QDP algorithm, which can be utilized for recursive summation in any format, and its superior performance in HPC environments compared to the Kahan algorithm (Kahan, 1965), this study aims to implement the QDP algorithm in the Model for Prediction Across Scales – Atmosphere (MPAS-A). The application of the QDP algorithm to a realistic numerical model, as presented in this study, represents a novel contribution to the field, with no prior research exploring this specific implementation.

Most work involving numerical models that reduce numerical precision adopt a mixed-precision scheme, where some variables use single precision while others remain in double precision to ensure integration stability, as demonstrated in the work of Chen et al. (2024). Currently, there are very few studies that almost entirely employ low precision (32 bit) in numerical models; this was only applied in IFS by Váňa et al. (2017). However, they only utilize single precision without considering error compensation for it. In this study, all variables in the numerical model were implemented using single precision, and the QDP algorithm was applied to key variables. Using the QDP algorithm, we can maintain integration stability comparable to that of applying the double-precision scheme while significantly reducing memory requirements

```

1 :  $u := \text{initial } u;$ 
2 :  $c := 0;$ 
3 :  $L: v := (\langle \text{evaluation of } v \rangle) + c;$ 
4 :  $s := u + v;$ 
5 :  $c := (v - (s - u)) + (u - (s - (s - u)))$ 
6 :  $u := s$ 
7 :  $go\ to\ L;$ 

```

Figure 1. The iterative process of the QDP algorithm in a step-by-step integration.

by lowering the numerical precision of all variables and improving the accuracy compared to applying the single precision. This approach not only reduces communication pressure but also allows for substantial increases in computational speed through vectorization optimization. The structure of this paper is as follows: Sect. 2 introduces the QDP algorithm, the MPAS model, application of the QDP algorithm in MPAS-A, and the experimental design and configuration. Section 3 provides a case study in MPAS. Section 4 presents conclusions and discussion of the experiments.

For clarity, the abbreviations and glossary shown in Table 1 are used throughout the entire paper.

2 Methodology, model, and experiments

2.1 Quasi-double-precision algorithm

The QDP algorithm, proposed by Møller (1965), aims to address the precision loss that occurs when adding small values to large values in floating-point arithmetic. This precision loss typically arises from coarse-truncation operations. The QDP algorithm reduces round-off errors by keeping corrections. Primarily applied in the step-by-step integration of ordinary differential equations, the algorithm significantly corrects rounding errors in the sum, particularly on computers where truncation operations are not followed by proper rounding.

A brief introduction to the algorithm is as follows, with a detailed derivation available in Møller (1965). Define the floating-point numbers u , v , s , and c , where at each step of the time integration, $s = u + v$. By introducing a correction variable c before computing the sum (s) of u and v in each step, the final s is adjusted to reduce round-off errors. This algorithm is illustrated in Fig. 1. It illustrates the iterative process (lines 3–7) and its role in error compensation, which enhances accuracy in time integration.

The process can be viewed as v being continuously incorporated into u ; however, in numerical model computations, it is impossible to ensure that u is always greater than v . To enhance the precision of the correction process, a precondition of magnitude comparison is added to the algorithm, as shown in Fig. 2.

It is important to note that the applicability of the QDP algorithm has been thoroughly analyzed (Møller, 1965); cases

Table 1. Key abbreviations and the glossary used in this study.

Abbreviation	Definition/description
MPAS-A	The atmospheric component of the Model for Prediction Across Scales (MPAS)
DBL	Double-precision simulation, used as the benchmark
SGL	Single-precision simulation
QDP	Single-precision simulation with the QDP applied (in this study, we apply the quasi-double precision method only to single-precision simulations. For convenience, we refer to the simulations using the QDP method in single precision as QDP).
QDP algorithm	Quasi-double precision, an algorithm to compensate for round-off errors in low-precision computations (to distinguish between the quasi-double-precision algorithm and single-precision simulations utilizing the quasi-double-precision algorithm, we refer to the QDP algorithm as the QDP algorithm and the simulations as QDP).

```

1 : c := 0;
2 : L: if u > v then;
3 :   u := initial u;
4 :   v := (< evaluation of v >) + c;
5 :   s := u + v;
6 :   c := (v - (s - u)) + (u - (s - (s - u)))
7 : else
8 :   v := initial v;
9 :   u := (< evaluation of u >) + c;
10 :  s := u + v;
11 :  c := (u - (s - v)) + (v - (s - (s - v)))
12 : go to L;

```

Figure 2. The QDP algorithm with magnitude preconditioning for identifying large and small numbers.

of inapplicability are exceedingly rare. Considering the numerous summation operations involved in numerical models, even if a few inapplicable instances occur, their impact on the overall result is negligible. Therefore, in practical applications, these infrequent cases are typically not considered.

2.2 MPAS-A

MPAS-A is a compressible, nonhydrostatic atmospheric numerical model developed by the National Center for Atmospheric Research (NCAR). It employs an unstructured centroidal Voronoi grid (mesh or tessellation) and a staggered C grid for state variables as the basis for horizontal discretization in the fluid flow solver. MPAS-A consists of two main components: the model, which includes atmospheric dynamics and physics, and the initialization component, which generates initial conditions for the atmosphere and land surface, updates for sea surface temperature and sea ice, and lateral boundary conditions. Both components (model and initialization) are integral constructs within the MPAS software framework and utilize the same drivers and software infrastructure.

MPAS-A solves the fully compressible, nonhydrostatic equations of motion (Skamarock et al., 2012). The spatial discretization uses a horizontal (spherical) centroidal

Voronoi mesh with a terrain-following geometric-height vertical coordinate and C-grid staggering for momentum. The temporal discretization uses the explicit time-split Runge–Kutta technique from Wicker and Skamarock (2002) and Klemp et al. (2007).

The algorithm applied here primarily addresses the round-off error compensation between large and small numbers in addition. Currently, it is only applicable to the time integration process and has not been implemented in the spatial discretization process. Therefore, this section will provide a detailed introduction to the time integration scheme. For the spatial discretization scheme, please refer to Skamarock et al. (2012), as it will not be introduced here.

The formulation of the scheme can be considered in one dimension as Eq. (1):

$$\frac{\partial \phi}{\partial t} = \text{RHS}_\phi. \quad (1)$$

The variable ϕ represents any prognostic variable in the prognostic equations, while RHS represents the right-hand side of the prognostic equations (i.e., the spatial discretization equation). In MPAS-A, a forward-in-time finite difference is used, and it can be written as Eq. (2):

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = \text{RHS}_\phi, \quad (2)$$

where the superscript represents the time step, and the subscript represents the position of grid zone.

The second-order Runge–Kutta time scheme used in MPAS-A is described in Gear (1973) as Eqs. (3), (4), and (5):

$$\phi^* = \phi^t + \frac{\Delta t}{2} \cdot \text{RHS}(\phi^t), \quad (3)$$

$$\phi^{**} = \phi^t + \frac{\Delta t}{2} \cdot \text{RHS}(\phi^*), \quad (4)$$

$$\phi^{t+\Delta t} = \phi^t + \Delta t \cdot \text{RHS}(\phi^{**}). \quad (5)$$

In this study, version 8.2.1 of MPAS-A was used for the following reasons.

1. This research primarily focuses on the accumulation of variables in time integration, specifically the accumulation of time integration variables within the dynamical core. Version 8.2.1 supports the option to close physical processes during model construction, preventing the influence of physical processes on the results of the dynamical core. Therefore, this version was chosen. It should be noted that all cases in this study have closed physical processes.
2. This version supports single-precision operations, reducing the repetitive work of code modification. It is not the only version that supports single precision, but it is the latest version currently released.

2.3 Application of the QDP algorithm in MPAS-A

According to Eqs. (3), (4), and (5), we can observe that in the time integration scheme, each step involves the process of adding trends to the basic field ϕ^t . In numerical models, the basic field is generally much larger than the trends, which aligns with the principles of numerical computation regarding the addition of large and small numbers, as well as the time integration process. It is important to note that the QDP algorithm currently only addresses time integration and has not been validated during the spatial discretization process. The spatial discretization primarily involves subtraction, specifically the subtraction of a small number from a large number or the subtraction of two close values. Whether this algorithm is applicable to spatial discretization remains uncertain; therefore, we will not apply it in this context.

Based on the application principles of the algorithm, which involve the processes of adding large and small numbers as well as the time integration process, we have established a strategy for applying the QDP algorithm within MPAS-A. Specific improvements are provided based on Eqs. (6), (7), (8), and (9):

$$\begin{aligned} \frac{\partial \mathbf{V}_H}{\partial t} = & -\frac{\rho_d}{\rho_m} \left[\nabla_\zeta \left(\frac{p}{\xi_\zeta} \right) - \frac{\partial z_{HP}}{\partial \zeta} \right] - \eta \mathbf{k} \times \mathbf{V}_H \\ & - v_H \nabla_\zeta \cdot \mathbf{V} - \frac{\partial \Omega v_H}{\partial \zeta} - \rho_d \nabla_\zeta K \\ & - eW \cos \alpha_r - \frac{v_H W}{r_e} + \mathbf{F}_{V_H}, \end{aligned} \quad (6)$$

$$\begin{aligned} \frac{\partial W}{\partial t} = & -\frac{\rho_d}{\rho_m} \left[\frac{\partial p}{\partial \zeta} + g \tilde{\rho}_m \right] - (\nabla \cdot \mathbf{v} W)_\zeta \\ & + \frac{uU + vV}{r_e} + e(U \cos \alpha_r - V \sin \alpha_r) + F_W, \end{aligned} \quad (7)$$

$$\frac{\partial \Theta_m}{\partial t} = -(\nabla \cdot \mathbf{V} \theta_m)_\zeta + F_{\Theta_m}, \quad (8)$$

$$\frac{\partial \tilde{\rho}_d}{\partial t} = -(\nabla \cdot \mathbf{V})_\zeta. \quad (9)$$

The meaning of each variable in the equations follows Skamarock et al. (2012), so we do not repeat the explanation

here. For a numerical model, the most crucial variables are the prognostic variables. Therefore, in the MPAS-A model we applied the QDP algorithm to the time integration process of these prognostic variables, including horizontal momentum (V_H), dry air density ($\tilde{\rho}_d$), potential temperature (Θ_m), and vertical velocity (W), that is, the calculation process on the left-hand side of Eqs. (6), (7), (8), and (9) (only the predictive equations for the dynamic core are presented here, without the scalar transport). This study focuses on the dynamic core, involving the gravity wave and acoustic wave, so we turned off the scalar transport in all cases. In order to be understood well, we provide the pseudo-code in the supplement.

2.4 Experimental design and configuration

This study aims to investigate whether the QDP algorithm can effectively compensate for the round-off errors that are caused by reduced numerical precision. Setting DBL as the benchmark experiment, two control experiments are also established: the first control experiment uses SGL, and the second control experiment uses QDP. By comparing the spatial root-mean-square error (spatial RMSE) and spatial mean absolute error (spatial MAE) between these two control experiments and the benchmark experiment, this study evaluates the effectiveness of the QDP algorithm at reducing round-off errors.

To assess the application effect of the QDP algorithm, we selected these two ideal cases and the real-data case because they are the only complete datasets available for download on the MPAS website.

1. *Jablonowski and Williamson baroclinic wave.* A deterministic initial-value test case (Jablonowski and Williamson, 2006) for dry dynamical cores of atmospheric general-circulation models is presented that assesses the evolution of an idealized baroclinic wave in the Northern Hemisphere. The primary objective is to assess the model's efficacy in replicating the typical dynamics of moist atmospheric conditions across various precision settings.
2. *Supercell.* A reduced-radius sphere (Klemp et al., 2015) can be used to assess the behavior of nonhydrostatic processes in global atmospheric dynamical cores, as long as the simulated cases demonstrate good agreement with the corresponding flows in Cartesian geometry, for which analytical solutions are available.
3. *Real data.* We used real data with initial conditions generated using Global Forecast System (GFS) data on 10 September 2014 at 00:00 UTC using two different total domain sizes (120 km \times 120 km and 240 km \times 240 km).

To prevent the influence of other factors, the basic parameters of all cases are kept consistent, including the number

Table 2. Key abbreviations and a glossary regarding the test cases used in this study.

Abbreviation	Definition/description
JW wave	Jablonowski and Williamson baroclinic wave
SC	Supercell
RD-120	Real data with a total domain size of 120×120 km
RD-240	Real data with a total domain size of 240×240 km

of acoustic steps per full Runge–Kutta (RK) step, config dynamics split steps, and config number of sub steps (integer), among others.

For clarity, the abbreviations about test cases as shown in Table 2 are used throughout the entire paper.

3 Results and analysis

In this section, we introduce spatial RMSE and MAE and show results across four cases, including two ideal scenarios, the Jablonowski and Williamson baroclinic wave and supercell, as well as two real cases (with initial conditions generated using GFS data) using two different total domain sizes. Using the spatial RMSE and MAE for quantitative comparison, the differences between the benchmark and control experiments are used to evaluate the effectiveness of the QDP algorithm at reducing round-off error.

3.1 Spatial RMSE and MAE

To quantify the differences between the SGL, QDP, and DBL (used as the benchmark), we calculate the spatial RMSE. First, for each grid point, the temporal averages of the variables (e.g., surface pressure, 500 hPa height) are computed across the entire simulation period for each experiment (SGL, QDP, and DBL). Then, the spatial RMSE is calculated as the root-mean-square difference between the temporally averaged fields of the control experiment (SGL or QDP) and the benchmark DBL, following Eq. (10):

$$\text{Spatial RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (M_i - C_i)^2}, \quad (10)$$

where, N is the total number of grid points, M_i is the temporally averaged value at grid point i for the benchmark double-precision experiment, and C_i is the temporally averaged value at grid point i for the control experiment (SGL or QDP).

In addition to the spatial RMSE, we also calculate the spatial MAE to assess the magnitude of the difference between the control experiments (SGL and QDP) and the benchmark DBL, irrespective of the direction of the difference. Like the spatial RMSE calculation, we first compute the temporal average for each grid point across the entire simulation period

Table 3. The spatial RMSE values of surface pressure compared to DBL for cases (in Pa). Note that JW wave is the Jablonowski and Williamson baroclinic wave, SC is the supercell, and RD-120/240 is real data with a total domain size of 120/240 km.

Case name	SGL	QDP (proposed)
JW wave	3.42×10^{-2}	1.09×10^{-2}
SC	8.80×10^{-4}	2.27×10^{-4}
RD-120	6.33×10^{-2}	2.25×10^{-3}
RD-240	6.68×10^{-2}	2.25×10^{-3}

for each experiment. The MAE is then calculated as the average absolute difference between the temporally averaged fields of the control experiment and the benchmark experiment, following Eq. (11):

$$\text{Spatial MAE} = \frac{1}{N} \sum_{i=1}^N |M_i - C_i|, \quad (11)$$

where N is the total number of grid points, M_i represents the temporally averaged value at grid point i for the benchmark DBL, and C_i represents the temporally averaged value at grid point i for the control experiment (either SGL or QDP).

Spatial RMSE is primarily used to measure the difference between predicted and actual values and is more sensitive to large errors. Spatial MAE calculates the average absolute prediction error and is less sensitive to outliers than spatial RMSE, making it more suitable for conventional error measurements. Therefore, the combination of spatial RMSE and MAE provides a more comprehensive evaluation. When comparing the performance of different experiments, spatial RMSE may be used to quantify differences in extreme values (such as temperature fluctuations, ocean current speeds, etc.), while spatial MAE is used to assess the accuracy of the model's overall trend. Combining both provides a better reflection of the algorithm's performance advantages.

As shown in Table 3 (spatial RMSE) and Table 4 (spatial MAE), the addition of the QDP algorithm consistently improves accuracy (compared to SGL) across all cases. For specific analysis, please refer to the following sections (the results of spatial RMSE and MAE are consistent, so to avoid duplication, only the results of spatial RMSE are analyzed in the following text).

3.2 Jablonowski and Williamson baroclinic wave

This case is a deterministic initial-value test case for dry dynamical cores of atmospheric general circulation models (Jablonowski and Williamson, 2006) that assesses the evolution of an idealized baroclinic wave in the Northern Hemisphere. The initial zonal state is quasi-realistic and entirely defined by analytical expressions, which are steady-state solutions of the adiabatic, inviscid primitive equations

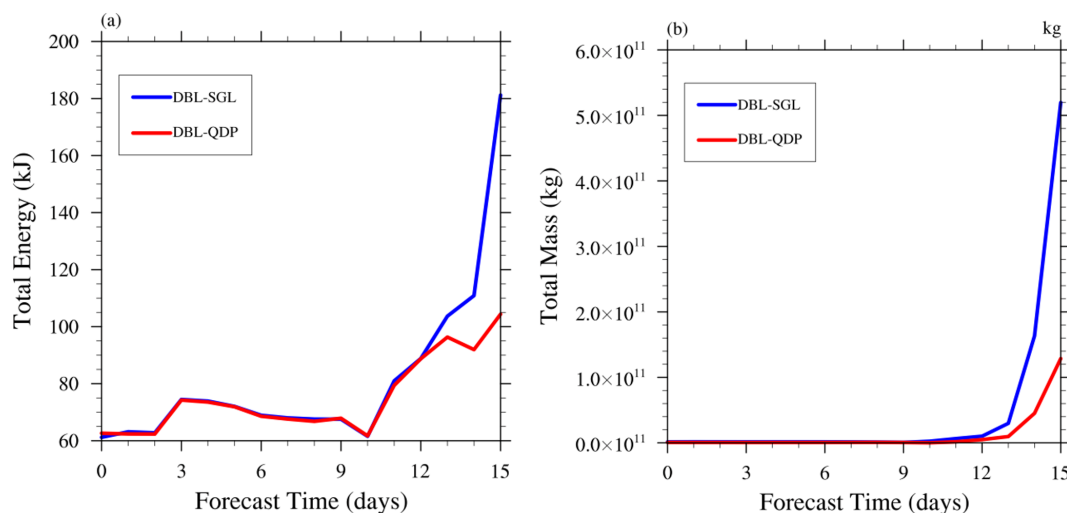


Figure 3. Time evolution of differences in (a) total energy and (b) total mass between DBL and SGL and DBL and QDP (proposed) in the JW wave. Both panels highlight QDP’s superior error compensation, especially over extended integration periods.

Table 4. The spatial MAE values of surface pressure compared to DBL for cases (in Pa). Note that JW wave is the Jablonowski and Williamson baroclinic wave, SC is the supercell, and RD-120/240 is real data with a total domain size of 120/240 km.

Case name	SGL	QDP (proposed)
JW wave	1.29×10^{-2}	3.81×10^{-2}
SC	8.79×10^{-4}	2.26×10^{-4}
RD-120	5.38×10^{-2}	1.95×10^{-3}
RD-240	5.52×10^{-2}	1.94×10^{-3}

in a pressure-based vertical coordinate system (Jablonowski and Williamson, 2006). The experimental configuration is consistent with the test case presented by Jablonowski and Williamson (2006), with a time step of 450 s, 26 vertical levels, total domain size of 120 km × 120 km, and an integration period of 15 d.

The bias begins to appear on day 10. Starting from day 10, the bias of total energy and total mass caused by SGL can be reduced using the QDP algorithm (Fig. 3a and b). Unlike SGL, where the bias increases rapidly after more than 10 d, QDP has a very small bias compared to DBL. Therefore, we believe that QDP can be used to replace DBL in medium-range weather forecasts.

We found that SGL can increase the round-off error in all regions (Fig. 4a); especially in high-latitude regions such as Southern Ocean westerly belt, its high wind speed increased the error caused by SGL, but instability caused by high wind speeds is more important. Surprisingly, the bias can be reduced significantly in QDP (Fig. 4b); it means that QDP can improve stability compared to SGL. It should be emphasized that this does not mean that the higher the wind speed,

the better the improvement effect. Instead, the improvement effect is more pronounced in areas with larger errors. The spatial RMSE of surface pressure between DBL and SGL is 3.42×10^{-2} Pa and 1.09×10^{-2} Pa between DBL and QDP; the error is reduced by 68 %.

The sources of unpredictability, as noted by Bauer et al. (2015), include instabilities that inject chaotic noise at small scales and the upscale propagation of their energy. For the cases examined, both SGL and QDP begin to exhibit errors after 10 d of integration. These errors arise from factors such as round-off errors due to reduced numerical precision and energy loss during the propagation process. The QDP algorithm can reduce the impacts of these errors.

While we acknowledge other potential sources of uncertainty such as initial-condition errors, we have not conducted an in-depth study on them in this research. Our primary focus remains evaluating the improvements provided by the compensation algorithm when addressing round-off errors.

3.3 Supercell case

The test case (Klemp et al., 2015), on a reduced-radius sphere, can evaluate the behavior of nonhydrostatic processes in nonhydrostatic global atmospheric dynamical cores, provided that the simulated cases exhibit good agreement with corresponding flows in Cartesian geometry and for which there are known solutions. The settings include a time step of 3 s, 40 vertical levels, a total domain size of 84 km × 84 km, and an integration period of 2 h.

In this case, for total mass (Fig. 5), the error caused by SGL can be effectively improved in QDP. This improvement exists throughout the entire integration period. Figure 6 shows the spatial distribution of perturbation theta, an important variable in numerical models: when reducing the nu-

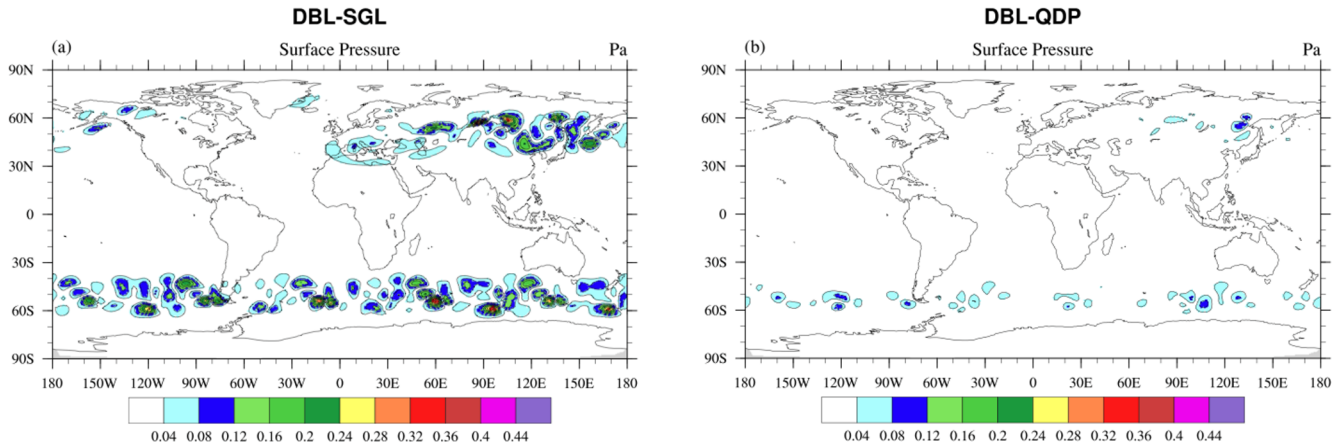


Figure 4. Spatial distributions of 1–15 d averaged surface pressure differences (Pa) in (a) DBL vs. SGL and (b) DBL vs. QDP in the JW wave case. QDP reduces errors more effectively, particularly in midlatitude to high-latitude regions.

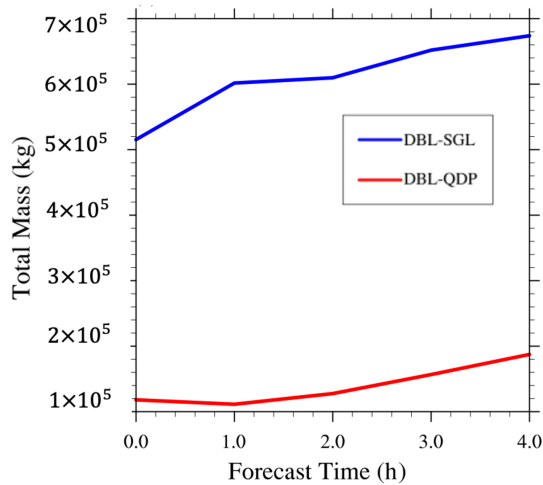


Figure 5. Time evolution of differences in total mass for the supercell case: DBL vs. SGL and DBL vs. QDP. The results highlight the QDP algorithm's effective error compensation, with benefits becoming more pronounced over time.

merical precision from double (Fig. 6a) to single (Fig. 6b), it displays differences and it indicates a significant increase in round-off error. In QDP, this difference can be compensated (Fig. 6c). The spatial RMSE of surface pressure between DBL and SGL is 8.95×10^{-4} Pa and it is 2.19×10^{-4} Pa between DBL and QDP; the error is reduced by 75 %.

3.4 Real-data cases

In this section, we will show the results from two cases using different total domain sizes. The settings include a time step of 720 s, 55 vertical levels, total domain sizes of $240 \text{ km} \times 240 \text{ km}$ and $120 \text{ km} \times 120 \text{ km}$, and an integration period of 15 d (except for the total domain size, all other configurations are exactly the same).

Consistent with the analysis presented in Sect. 3.2, errors are relatively small in the early stages and begin to emerge after 140 h. This increase is attributed to the accumulation of round-off errors and energy loss over time. The effects become more pronounced beyond 140 h. Overall, the QDP algorithm demonstrates a certain level of improvement in addressing these errors. The case with the total domain of $240 \text{ km} \times 240 \text{ km}$ (Fig. 7a) shows an error larger than in the $120 \text{ km} \times 120 \text{ km}$ (Fig. 7b) case, and the error can be reduced in QDP compared to SGL.

Figures 8 and 10 show spatial distributions of surface pressure in the total domain. The error has reduced throughout all the regions, and the improvement effect is very obvious. From a spatial perspective, the case of SGL with the total domain size of $240 \text{ km} \times 240 \text{ km}$ (Fig. 8a) shows a larger error than in the $120 \text{ km} \times 120 \text{ km}$ domain (Fig. 10a) case, and the errors in both were reduced by QDP (Figs. 8b and 10b). The spatial RMSE of surface pressure at $240 \text{ km} \times 240 \text{ km}$ between DBL and SGL is 6.68×10^{-2} Pa, and it is 2.25×10^{-3} Pa between DBL and QDP; the error is reduced by 97 %. The spatial RMSE of surface pressure at $120 \text{ km} \times 120 \text{ km}$ between DBL and SGL is 6.33×10^{-2} Pa, and it is 2.25×10^{-3} Pa between DBL and QDP; the error is reduced by 96 %.

Figures 9 and 11 show spatial distributions of 500 hPa height with different total domain sizes: $240 \text{ km} \times 240 \text{ km}$ (Fig. 9) and $120 \text{ km} \times 120 \text{ km}$ (Fig. 11). The error improvement effect is consistent with surface pressure. The spatial RMSE of 500 hPa height at $240 \text{ km} \times 240 \text{ km}$ between DBL and SGL is 2.80×10^{-1} m, and it is 1.40×10^{-1} m between DBL and QDP; the error is reduced by 50 %. The spatial RMSE at $120 \text{ km} \times 120 \text{ km}$ between DBL and SGL is 4.35×10^{-3} Pa, and it is 1.90×10^{-3} Pa between DBL and QDP; the error is reduced by 56 %.

In this research, we focus on the processes of summing the basic field and trends. When the resolution is increased, the

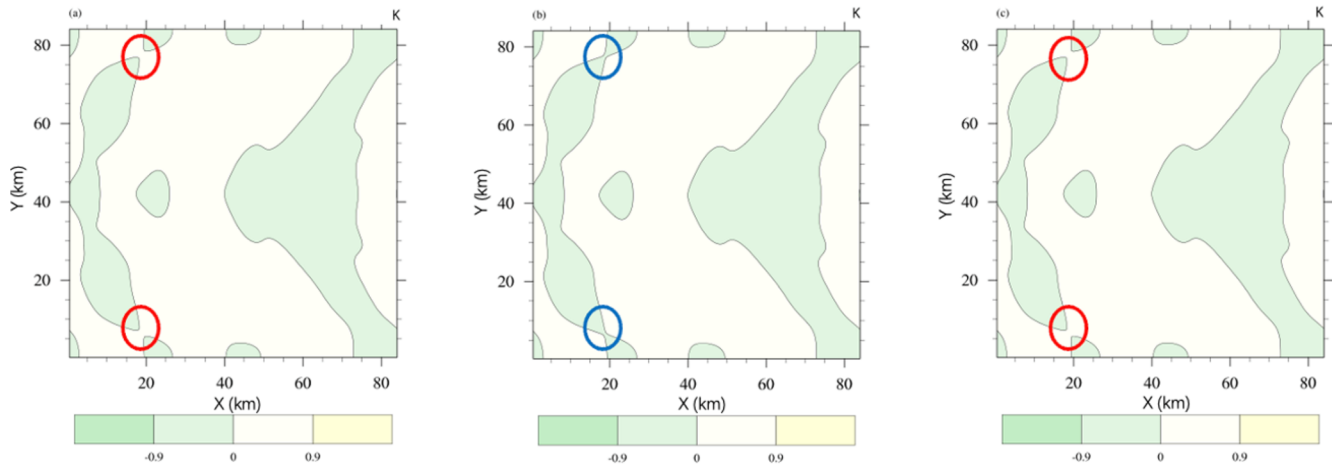


Figure 6. Perturbation theta at 5400 s in supercell development: (a) DBL, (b) SGL, and (c) QDP (in K). The circles highlight pattern biases (consistent values in the same color). Error regions (blue) appear in (b) with single precision, while QDP reduces these errors in (c).

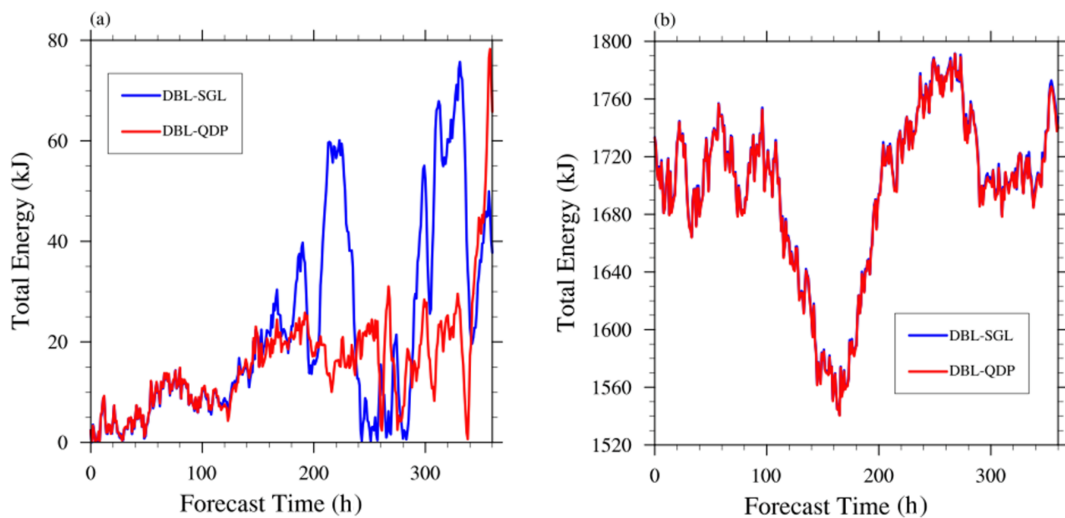


Figure 7. Temporal evolution of total energy differences for real-data simulations: DBL vs. SGL (blue) and DBL vs. QDP (red) at resolutions of (a) 240 km × 240 km and (b) 120 km × 120 km. QDP consistently reduces errors from numerical precision loss, with resolution-dependent improvements.

basic field remains relatively unchanged; however, the trends become smaller. This characteristic aligns with the nature of adding large and small numbers, making the advantages of the QDP algorithm more pronounced. Thus, it is evident that as the resolution increases, the improvement achieved by the QDP algorithm also enhances.

On the other hand, it is important to note that the propagation of round-off errors is not immediately apparent over short timescales. However, as the number of iterations increases, these errors can become more significant. The QDP algorithm employs compensation mechanisms that help mitigate the propagation of these errors.

Due to the current limitations of the MPAS-A website, which only provides a single set of terrain and initial-condition fields for different experiments, our future plan is

to request assistance from the MPAS-A website to construct different terrain and initial-condition fields for a specific experiment. We aim to conduct a sensitivity analysis, particularly for real-data experiments. Provided that computational resources allow, we plan to carry out simulations with different resolutions and initial conditions. This will help lay the data foundation for future uncertainty analysis.

3.5 Computational performance

In comparison with the SGL, there is a slight increase in runtime, although it is minimal: only 6.0 % (JW wave), 0.3 % (SC), 2.2 % (RD-120), and 17.8 % (RD-240) (Table 5). This slight increase is attributed to the addition of a small number of global variable arrays when using the QDP algo-

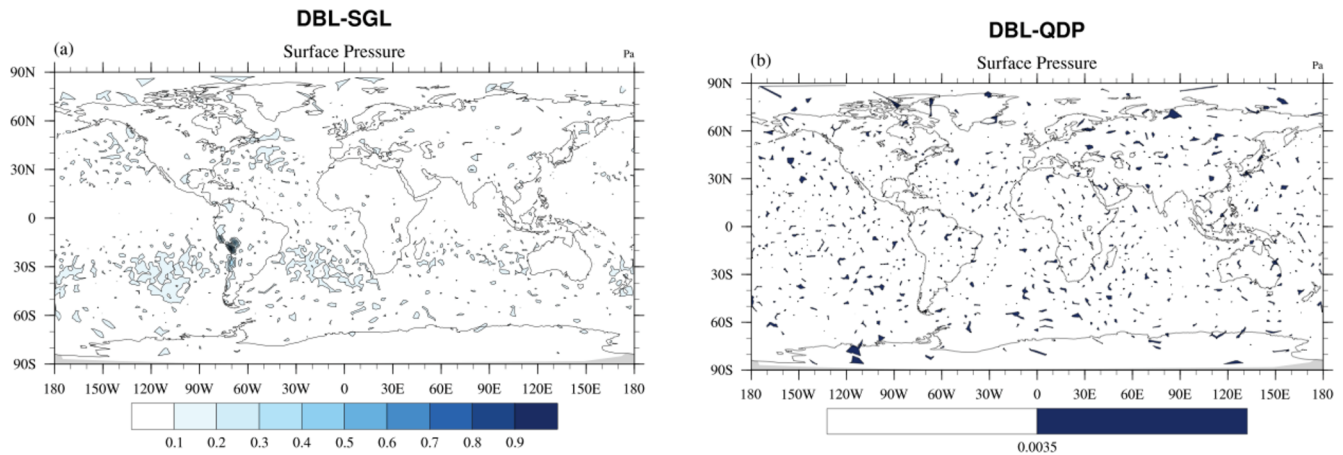


Figure 8. Spatial distributions of averaged (1–15 d) surface pressure differences (Pa) in (a) DBL vs. SGL and (b) DBL vs. QDP (domain size is $240 \text{ km} \times 240 \text{ km}$). The RMSE is $6.68 \times 10^{-2} \text{ Pa}$ for (a) and $2.25 \times 10^{-3} \text{ Pa}$ for (b), highlighting QDP's significant error reduction across regions and its effectiveness at correcting errors by several orders of magnitude.

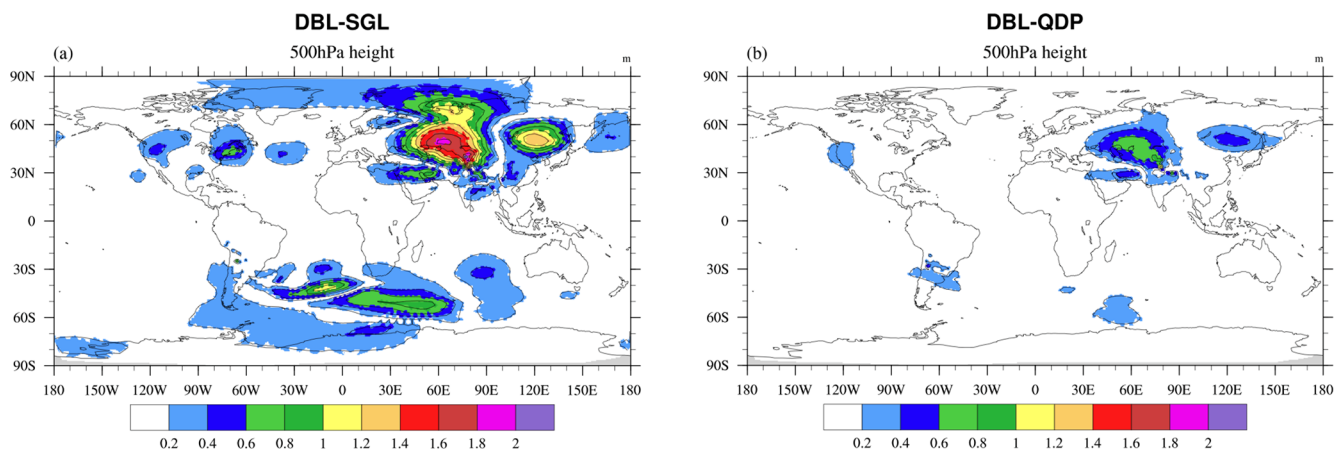


Figure 9. Spatial distributions of averaged (1–15 d) 500 hPa height differences (m) in (a) DBL vs. SGL and (b) DBL vs. QDP (domain size is $240 \text{ km} \times 240 \text{ km}$). The RMSE decreases from $2.80 \times 10^{-1} \text{ m}$ in (a) to $1.40 \times 10^{-1} \text{ m}$ in (b), indicating notable spatial error reduction with QDP.

rithm. Compared to DBL, QDP demonstrated relatively better performance across different cases, reducing the runtime by 28.6 % (JW wave), 28.5 % (SC), 21.1 % (RD-120), and 5.7 % (RD-240) (Table 5).

4 Conclusions and discussion

This study aims to demonstrate the potential application of the QDP algorithm in numerical models. Although QDP algorithms have been widely used for time integration of ordinary differential equations, their application in real-world numerical models has not yet been explored. This research bridges this gap by introducing a novel implementation of the QDP algorithm, thus extending its scope and potential impact in the field of numerical forecasting. As the QDP algorithm manages round-off errors through corrections for

Table 5. Comparative analysis of computational efficiency: QDP vs. DBL and QDP vs. SGL.

Case name	DBL	SGL	QDP (proposed)		
	Runtime	Runtime	Runtime	vs. DBL	vs. SGL
JW wave	1768 s	1191 s	1263 s	−28.6 %	+6.0 %
SC	1507 s	1073 s	1077 s	−28.5 %	+0.3 %
RD-120	19 126 s	14 765 s	15 092 s	−21.1 %	+2.2 %
RD-240	1397 s	1118 s	1317 s	−5.7 %	+17.8 %

Note that JW wave is the Jablonowski and Williamson baroclinic wave, SC is the supercell, and RD-120/240 is real data with a total domain size of 120/240 km.

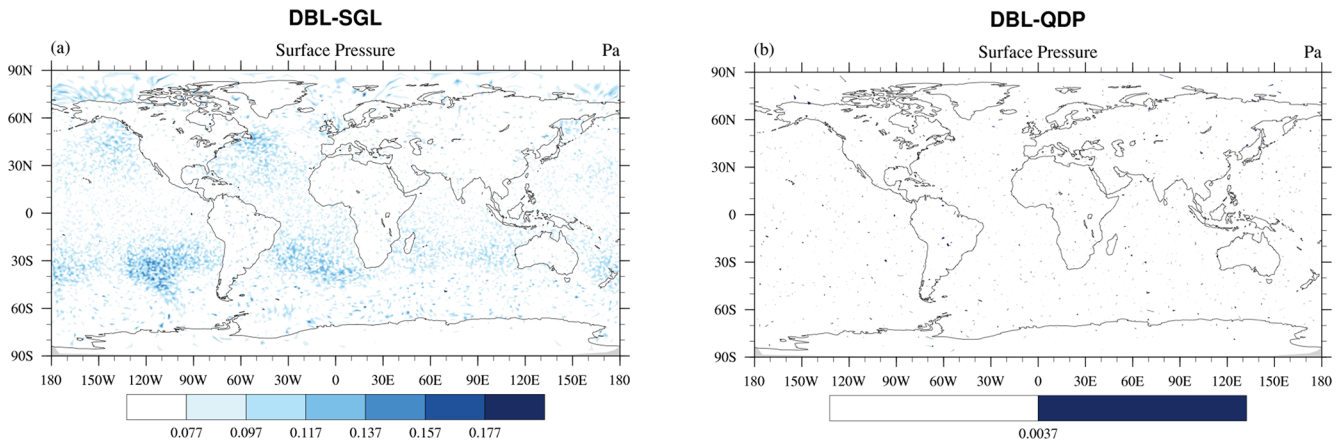


Figure 10. Distributions of averaged (1–15 d) surface pressure differences (Pa) in (a) DBL vs. SGL and (b) DBL vs. QDP (domain size is 120 km × 120 km). The RMSE decreases from 6.33×10^{-2} Pa in (a) to 2.25×10^{-3} Pa in (b). Note that the color bars differ between (a) and (b). QDP significantly reduces errors across all regions.

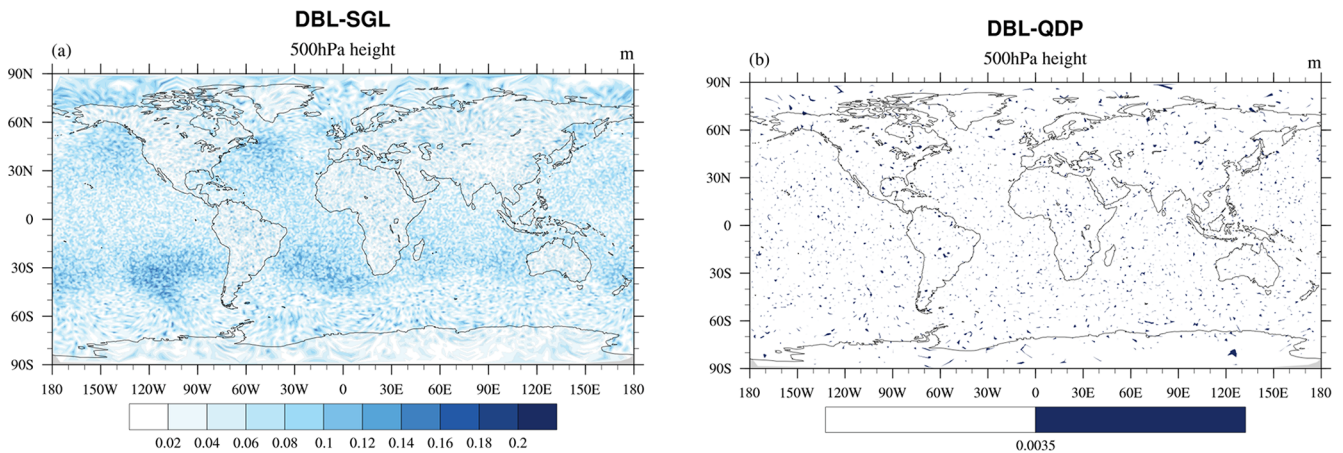


Figure 11. Spatial distributions of averaged (1–15 d) 500 hPa height differences (m) in (a) DBL vs. SGL and (b) DBL vs. QDP (domain size is 120 km × 120 km). The RMSE decreases from 4.35×10^{-3} m in (a) to 1.90×10^{-3} m in (b), consistent with the overall improvement shown in Fig. 10.

both large and small numbers, it aligns with numerical models where basic fields significantly exceed tendency fields. Therefore, we have developed a strategy to apply the QDP algorithm in the MPAS-A model. In this study, the application of the QDP algorithm in the single-precision version reduced surface pressure errors by 68 %, 75 %, 97 %, and 96 % in four cases, while the runtime decreased by 28.6 %, 28.5 %, 21.1 %, and 5.7 % compared to the double-precision experiments (see Sect. 3). Overall, the QDP algorithm achieves a balance between maintaining accuracy and reducing computational cost.

However, the application of the QDP algorithm has certain limitations. First, the algorithm relies on the iterative process of time integration, and its effectiveness depends on the number of iterations; generally, increasing the number of iterations enhances the error compensation. Second, although the application of QDP mitigates errors to some extent, it still ex-

hibits errors when compared to DBL, making it less suitable for situations that require high precision. Furthermore, the application of the QDP algorithm necessitates the introduction of additional variables, which increases the complexity to some degree.

Furthermore, whether the QDP algorithm is applicable to the spatial discretization process requires further investigation. Although floating-point operations such as addition, multiplication, and division are often performed multiple times during spatial discretization, which can introduce round-off errors, these errors do not accumulate and amplify as they do in time integration. This is primarily because spatial computations generally do not involve the repetitive time accumulation process. Additionally, the errors in spatial discretization mainly stem from discretization errors (such as grid resolution) and the choice of discretization methods (e.g., central difference, forward difference), which dif-

fer from round-off errors and are primarily related to the discretization method and grid design. Therefore, the effectiveness of the QDP algorithm in this context may not be as pronounced as it is in time integration. As a result, this study does not apply the QDP algorithm to the spatial discretization process.

In large-scale numerical simulations, the impact of round-off errors cannot be ignored, which is why double precision is commonly employed to maintain the accuracy of results. However, double-precision computations significantly increase computation time and resource consumption, which is often impractical for large-scale simulations. Using the QDP algorithm, we not only reduced memory and communication overhead but also enhanced scalability, particularly in ultra-large parallel simulations where inter-node communication can become a major performance bottleneck. Moreover, while our experiments did not include vectorization strategies, there is considerable potential for further performance improvements.

In future work, we plan to explore vectorization strategies for the QDP algorithm, building on successful implementations of vectorized compensated summation algorithms. Dmitruk and Stpicyński (2023) have efficiently vectorized Kahan and Gill-Møller's compensated summation algorithms using Intel AVX-512 intrinsics, with parallelization handled through OpenMP constructs. Numerical experiments have shown that the vectorized summation algorithm achieves performance comparable to traditional summation algorithms, especially for large problem sizes, while maintaining high accuracy. So we intend to apply similar vectorization techniques to the QDP algorithm in numerical models, utilizing single instruction, multiple data (SIMD) extensions in modern multicore processors to accelerate the computation of compensated summation and other time-stepping algorithms. Future implementations will also include parallelization via easy-to-use constructs like OpenMP's "declare reduction", which can further speed up execution, especially for large-scale problems. However, for smaller problem sizes or when summation is part of a more complex computation, we may find parallelization to be beneficial even at smaller scales. By incorporating these vectorization and parallelization strategies, we aim to significantly enhance the efficiency and accuracy of the QDP algorithm in HPC environments.

Looking ahead, we plan to extend the application of the QDP algorithm to additional components of the MPAS-A model. Currently, the QDP algorithm is implemented only within the time integration scheme of the dynamic core, with no consideration for its application to tracer transport. Tracer transport processes involve numerous operations where both large and small values are added together, making them particularly sensitive to precision requirements. In addition, the QDP algorithm has been applied to ideal and real-data tests at low and medium resolutions, while its performance at high resolutions has not yet been studied. In future work, we will focus on these fields.

Code and data availability. The repository provided includes all relevant code necessary for the study, categorized into four main components.

1. *Download model source code.* This includes source code of MPAS-v8.2.1 for different simulation modes.
 - a. DBL and SGL
 - Both DBL and SGL are available on GitHub: <https://github.com/MPAS-Dev/MPAS-Model/releases/tag/v8.2.1> (Duda, 2024).
 - *Note.* If the source code is obtained via the official GitHub repository of the MPAS model, to build a dycore-only MPAS-A model, users need to comment-out or delete the definition of PHYSICS in the Makefile located in the `src/core_atmosphere/`, e.g., `# PHYSICS=--DDO_PHYSICS`.
 - *Note.* The source code for SGL is identical to DBL. The difference lies in the compilation process, where a specific compilation option is used to enable single-precision execution. To compile the model in single precision, simply add the `PRECISION=single` flag during the build process.
 - DBL and SGL are also available on Zenodo: <https://doi.org/10.5281/zenodo.14576893> (Lai, 2024) (located in `code_and_data/model/DBL/` and `code_and_data/model/SGL/`).
 - b. QDP (proposed)
 - QDP is available on Zenodo: <https://doi.org/10.5281/zenodo.14576893> (Lai, 2024) (located in `code_and_data/model/QDP/`).
 - *Note.* Add the `PRECISION=single` flag during the build process.
2. *Compile model source code.* This step includes generating the executable files for `init_atmosphere` and `atmosphere`.
 - Compile `init_atmosphere` using the following command: `make ifort CORE=init_atmosphere`.
 - Clean previous builds for `atmosphere` (if necessary) using the following command: `make clean CORE=atmosphere`.
 - Compile `atmosphere` using the following command: `make ifort CORE=atmosphere`.
 - To compile the model in single precision, add the `PRECISION=single` flag: `make ifort CORE=atmosphere PRECISION=single`.
3. *Case setup and run.* Specific case setups and configurations used in the experiments are provided, including input files, namelist configurations, and scripts for running idealized and real-world scenarios. These allow users to replicate the exact experiments conducted in the study. The steps are as follows.
 - Download the archive file for the test cases, which includes mesh files, decomposition files, and the namelist file, from the official MPAS website at <http://mpas-dev.github.io/> (MPAS, 2025). The idealized test cases currently available on the official website include the supercell, mountain wave, and Jablonowski and Williamson baroclinic wave. These can also be downloaded directly from

<https://doi.org/10.5281/zenodo.14576893> (Lai, 2024) (located in `code_and_data/test/`).

- Link the `init_atmosphere` and `atmosphere` executable files, compiled in the first part, to the case folder.
 - If the code is downloaded directly from Zenodo, users can run the cases by following the instructions in the README file or directly executing the `run.sh` script. Before running the simulations, adjust the number of nodes in the script according to the available computational resources to optimize performance.
4. *Visualization and postprocessing code.* In order to reproduce the figures in this paper, follow the instructions below.
- For Figs. 3 and 4, run the NCL scripts `ncl time.ncl` and `ncl spatical.ncl`. Navigate to the directory `code_and_data/test/c2_DBL/`.
 - For Figs. 5 and 6, run the NCL scripts `ncl time.ncl` and `ncl spatical.ncl`. Navigate to the directory `code_and_data/test/c5_DBL/`.
 - For Figs. 7a, 8, and 9, run the NCL scripts `ncl time.ncl` and `ncl spatical.ncl`. Navigate to the directory `code_and_data/test/c7_240km_DBL/`.
 - For Figs. 7b, 10, and 11, run the NCL scripts `ncl time.ncl` and `ncl spatical.ncl`. Navigate to the directory `code_and_data/test/c7_120km_DBL/`.

Supplement. The supplement related to this article is available online at <https://doi.org/10.5194/gmd-18-1089-2025-supplement>.

Author contributions. JYL and LNW developed the code applying the quasi-double-precision algorithm to MPAS-A and designed the structure of the manuscript. JYL carried out the simulations and analyzed the results with help from LNW, YZY, FW, QZW, and HQC. All authors gave comments and contributed to the development of the paper.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Acknowledgements. The authors would like to thank the administrator of Beijing Normal University High-Performance Computing for providing the high-performance computing (HPC) environment and technical support. The research work presented in this paper was supported by the National Key Research and Development Program of China (grant no. 2023YFB3002405) and the National Natural Science Foundation of China (grant no. 42375162).

Financial support. This research has been supported by the National Key Research and Development Program of China (grant no. 2023YFB3002405) and the National Natural Science Foundation of China (grant no. 42375162).

Review statement. This paper was edited by Xiaomeng Huang and reviewed by four anonymous referees.

References

- Banderier, H., Zeman, C., Leutwyler, D., Rüdüsühli, S., and Schär, C.: Reduced floating-point precision in regional climate simulations: an ensemble-based statistical verification, *Geosci. Model Dev.*, 17, 5573–5586, <https://doi.org/10.5194/gmd-17-5573-2024>, 2024.
- Bauer, P., Thorpe, A., and Brunet, G.: The quiet revolution of numerical weather prediction, *Nature*, 525, 47–55, <https://doi.org/10.1038/nature14956>, 2015.
- Chen, S., Zhang, Y., Wang, Y., Liu, Z., Li, X., and Xue, W.: Mixed-precision computing in the GRIST dynamical core for weather and climate modelling, *Geosci. Model Dev.*, 17, 6301–6318, <https://doi.org/10.5194/gmd-17-6301-2024>, 2024.
- Cotronei, A. and Slawig, T.: Single-precision arithmetic in ECHAM radiation reduces runtime and energy consumption, *Geosci. Model Dev.*, 13, 2783–2804, <https://doi.org/10.5194/gmd-13-2783-2020>, 2020.
- Dawson, A. and Düben, P. D.: rpe v5: an emulator for reduced floating-point precision in large numerical simulations, *Geosci. Model Dev.*, 10, 2221–2230, <https://doi.org/10.5194/gmd-10-2221-2017>, 2017.
- Dawson, A., Düben, P. D., MacLeod, D. A., and Palmer, T. N.: Reliable low precision simulations in land surface models, *Clim. Dynam.*, 51, 2657–2666, 2018.
- Dmitruk, B. and Stpiczyński, P.: Improving accuracy of summation using parallel vectorized Kahan's and Gill-Møller algorithms, *Concurr. Comput.: Pract. Exp.*, 35, e7763, <https://doi.org/10.1002/cpe.7763>, 2023.
- Duda, M.: MPAS-Model v8.2.1, GitHub [code], <https://github.com/MPAS-Dev/MPAS-Model/releases/tag/v8.2.1> (last access: 26 December 2024), 2024.
- Gear, C.: Numerical initial value problems in ordinary differential equations, *HomeSIAM Review*, 15, 3, <https://doi.org/10.1137/1015088>, 1973.
- Gill, S.: A process for the step-by-step integration of differential equations in an automatic digital computing machine, in: *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 47, Cambridge University Press, 96–108, <https://doi.org/10.1017/S0305004100026414>, 1951.
- Hatfield, S., Chantry, M., Düben, P., and Palmer, T.: Accelerating high-resolution weather models with deep-learning hardware, in: *Proceedings of the platform for advanced scientific computing conference, PASC '19: Proceedings of the Platform for Advanced Scientific Computing Conference, Zurich, Switzerland, 2019*, 1, 1–11, <https://doi.org/10.1145/3324989.3325711>, 2019.
- Higham, N. J.: Accuracy and stability of numerical algorithms, 2nd Edn., SIAM, <https://doi.org/10.1137/1.9780898718027>, 2002.

- Jablonowski, C. and Williamson, D. L.: A baroclinic instability test case for atmospheric model dynamical cores, *Q. J. Roy. Meteorol. Soc.*, 132, 2943–2975, 2006.
- Kahan, W.: Pracniques: further remarks on reducing truncation errors, *Commun. ACM*, 8, 40, <https://doi.org/10.1145/363707.363723>, 1965.
- Klemp, J., Skamarock, W., and Park, S.-H.: Idealized global nonhydrostatic atmospheric test cases on a reduced-radius sphere, *J. Adv. Model. Earth Syst.*, 7, 1155–1177, 2015.
- Klemp, J. B., Skamarock, W. C., and Dudhia, J.: Conservative split-explicit time integration methods for the compressible nonhydrostatic equations, *Mon. Weather Rev.*, 135, 2897–2913, 2007.
- Lai, J.: Enhancing Single-Precision with Quasi Double-Precision: Achieving Double-Precision Accuracy in the Model for Prediction Across Scales-Atmosphere (MPAS-A) version 8.2.1, Zenodo [code], <https://doi.org/10.5281/zenodo.14576893>, 2024.
- Møller, O.: Quasi double-precision in floating point addition, *BIT Numer. Math.*, 5, 37–50, 1965.
- MPAS: MPAS-Atmosphere Idealized Test Cases, <https://mpas-dev.github.io/>, last access: 29 February 2025.
- Nakano, M., Yashiro, H., Kodama, C., and Tomita, H.: Single precision in the dynamical core of a nonhydrostatic global atmospheric model: Evaluation using a baroclinic wave test case, *Mon. Weather Rev.*, 146, 409–416, 2018.
- Paxton, E. A., Chantry, M., Klöwer, M., Saffin, L., and Palmer, T.: Climate modeling in low precision: Effects of both deterministic and stochastic rounding, *J. Climate*, 35, 1215–1229, 2022.
- Skamarock, W. C., Klemp, J. B., Duda, M. G., Fowler, L. D., Park, S.-H., and Ringler, T. D.: A multiscale nonhydrostatic atmospheric model using centroidal Voronoi tessellations and C-grid staggering, *Mon. Weather Rev.*, 140, 3090–3105, 2012.
- Thompson, R. J.: Improving round-off in Runge-Kutta computations with Gill’s method, *Commun. ACM*, 13, 739–740, 1970.
- Tintó Prims, O., Acosta, M. C., Moore, A. M., Castrillo, M., Seradell, K., Cortés, A., and Doblas-Reyes, F. J.: How to use mixed precision in ocean models: exploring a potential reduction of numerical precision in NEMO 4.0 and ROMS 3.6, *Geosci. Model Dev.*, 12, 3135–3148, <https://doi.org/10.5194/gmd-12-3135-2019>, 2019.
- Tomonori, K. and Hideko, N.: On the correction method of round-off errors in the Yang’s Runge-Kutta method, *Jpn. Soc. Indust. Appl. Meth.*, 5, 293–305, 1995.
- 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, 2017.
- Wicker, L. J. and Skamarock, W. C.: Time-splitting methods for elastic models using forward time schemes, *Mon. Weather Rev.*, 130, 2088–2097, 2002.