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 Lai[1], Lanning Wang[1,2], Qizhong Wu[1,2], Yizhou Yang[3], and Fang Wang[4]

[1]College of Global Change and Earth System Science, Faculty of Geographical Science, Beijing Normal University, Beijing 100875, China
[2]Joint Center for Earth System Modeling and High Performance Computing, Beijing Normal University, Beijing 100875, China
[3]National Supercomputing Center, Wuxi 214026, China
[4]CMA Earth System Modeling and Prediction Centre (CEMC), Beijing 100081, China

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

Abstract. The development of numerical models are constrained by the limitations of high performance computing (HPC). ~~Low~~ While low precision computations can significantly reduce computational costs, ~~but inevitably~~ they may introduce rounding errors ~~, which~~ that can affect computational accuracy under certain conditions. Quasi double-precision algorithm can compensate for rounding errors by keeping corrections, thereby achieving the low numerical precision while maintaining result accuracy. This paper applies the algorithm to the Model for Prediction Across Scales-Atmosphere (MPAS-A) and evaluate its performance across four test cases. The results demonstrate that, after reducing numerical precision to single precision (from 64 bits to 32 bits), the application of quasi double-precision algorithm can achieve results comparable to double-precision computations. The ~~round-off error~~ bias of surface pressure ~~is reduced~~ are reduced respectively by 68%, 75%, 97% ~~,~~ and 96% in cases, the memory has been reduced by almost half, while the computation increases only ~~2%,~~ 6.0%, 0.3%, 2.2%, and 17.8% respectively, significantly reducing computational cost. ~~The work substantiates~~ This research demonstrates that the quasi double-precision algorithm provides both effectiveness and ~~inexpensive computation in numerical modelsby using quasi double-precision algorithm~~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 ~~this context~~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 ~~current~~ 2015's systems. The development of future high-performance computing is crucial for the continued advancement of numerical weather forecasting. Therefore, to meet this technological challenge, the design of code and the selection of algorithms must ~~prioritize~~ focus on the optimization of floating-point operations and memory usage~~to meet this technological challenge~~ (Hatfield et al. 2019).

Mixed precision is a critical 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 not only significantly decreases memory requirements but also substantially reduces the computational and communication costs in numerical software projects such as climate modeling. Employing lower precision numerical representations is a feasible option for reducing the computational costs of complex numerical models (Dawson et al. 2017). Specifically, we define low precision computations as those that utilize a limited number of significant digits (less than 64 bits) during numerical operations, which can significantly reduce the computational resources required while potentially introducing rounding errors. Consequently, 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. (2016) 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, thereby providing a crucial reference for subsequent researchers. Dawson et al. (2018) expanded the scope of mixed-precision methods, demonstrating their applicability to simple thermal diffusion models, provided that 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 non-hydrostatic model, particularly in 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 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. ~~In the same year,~~ Oriol Tintó et al. (2019) applied mixed-precision methods to the European ocean simulation core (NEMO). They discovered that 95.8% of the 962 variables could be computed using ~~half~~ 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 at al. (2020) converted the radiation component of the atmospheric 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 at al. (2022) further investigated the feasibility of reduced-precision computing. He conducted 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 (23 bits) sufficed for most computational needs, and in numerous cases, half precision (10 bits) could also achieve the desired results. This provides an important reference for adopting lower-precision computing in various models in the future. This year, Hugo et al. (2024) further substantiated the effectiveness of mixed-precision methods in the regional weather and climate model COSMO. He 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 of the runtime of the model's hydrostatic solver, non-hydrostatic 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 rounding 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 $A = 0.7315 * 10^3$ (a large number) and ~~$B = 0.4506 * 10^{-5}$~~ $B = 0.4506 * 10^{-5}$ (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 rounding errors. These errors can accumulate over successive computations, leading to a degradation in model accuracy or even complete failure. Therefore, addressing the rounding errors induced by low-precision computations is a critical area for further research.

In 1951, Gill (1951) proposed a fourth-order, four-step explicit Runge-Kutta method aimed at correcting rounding errors during computation. This method constructs auxiliary variables at each step to compensate for the rounding 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 1996). These methods rely on recursive summation and incorporate correction terms to reduce rounding errors. In 1965, Møller (1965) and Kahan (1965) respectively proposed the quasi double-precision method and the Kahan method. 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 quasi double-precision method 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 rounding errors are primarily employed in the step-by-step integration of ordinary differential equations (Thompson et al. 1970; Tomonori et al. 1995; Dmitruk et al. 2023). However, their validation in numerical models remains uncertain. Considering the broader applicability of the quasi double-precision method, which can be utilized for recursive summation in any format, and its superior performance in high-performance computing environments compared to the Kahan method (Kahan 1965), this study aims to implement the quasi double-precision method in MPAS-A model. ~~By addressing the sum of large and small numbers during the time integration process,~~ The application of the ~~single-precision version improved with the~~ Moller method 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 works 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, only applied in IFS by Váňa et al. (2016). 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 error compensation was applied to key variables. By using error compensation methods (quasi double-precision~~algorithm achieves basically consistent results comparable to those of double precision .~~), we can maintain integration stability comparable to that applying double precision scheme while significantly reducing memory requirements by lowering the numerical precision of all variables and improved the accuracy comparable to that 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: Section 2 introduces the quasi double-precision algorithm, the MPAS model, application of quasi double-precision algorithm in MPAS-A, and the experimental design and configuration. Section 3 provides case study in MPAS. Section 4 presents conclusions and discussion of the experiments.

2 Methodology, model and experiments

2.1 Quasi double-precision algorithm

The quasi double-precision algorithm, proposed by Møller et al. (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 quasi double-precision 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 sum, particularly in computers where truncation operations are not followed by proper rounding.

A brief introduction to the algorithm is as follows, with a detailed derivation available follows Møller et al. (1965). Define the floating-point numbers u, v, s, and c, where in each step of the time integration, s=u+v. By introducing a correction variable

```
1 :    u ≔ initial u;
2 :    c ≔ 0;
3 : L: v ≔ (< evalution of v >) + c;
4 :    s ≔ u + v;
5 :    c ≔ (v − (s − u)) + (u − (s − (s − u)))
6 :    u ≔ s
7 :    go to L:
```

Figure 1. The quasi double-precision algorithm in case of a step-by-step integration.

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

Figure 2. The quasi double-precision algorithm adding a precondition of magnitude.

c before computing sum (*s*) of *u* and *v* in each step, the final *s* is adjusted to reduce rounding errors. This algorithm is illustrated in Figure 1.

The process can be viewed as *v* being continuously accumulated onto *u*; however, in numerical models' 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 Figure 2.

It is important to note that the applicability of the quasi double-precision algorithm has been thoroughly analyzed (Møller et al. 1965), cases of inapplicability are exceedingly rare. Considering the numerous sum algorithm and integration 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, non-hydrostatic atmospheric numerical model developed by 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.

The ~~dynamical core of~~ MPAS-A solves the fully compressible, nonhydrostatic equations of motion (Skamarock et al. ~~2012). These fully compressible nonhydrostatic equations are transformed based on geometric height vertical coordinates. The solver employs a split-explicit time integration scheme as described by Klemp et al. (2007) . The time integration scheme utilizes Runge-Kutta methods with a large time step, while for the acoustic modes, a smaller time step forward-backward method is employed (~~ 2011). 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).

~~MPAS-A currently offers two time integration schemes: a second-order Runge-Kutta method and a third-order Runge-Kutta method, which can be configured through namelist parameters. The default setting is~~ The algorithm applied here primarily addresses the rounding 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 ~~second-order Runge-Kutta method, and this study uses the default setting for experiments. The numerical schemes used in MPAS-A are very similar to those used in the Advanced Research WRF (ARW) model. The main differences are that the ARW model uses a rectangular grid and hydrostatic pressure (mass)vertical coordinates. Additionally, MPAS employs a vector-invariant form of the horizontal momentum equations and a more general version of the WRF transport scheme as given by Skamarock and Gassmann (~~time integration scheme. For the spatial discretization scheme, please refer to Skamarock et al. (2011), and it will not be introduced upon here.

The formulation of the scheme can be considered in on dimension as equation Wicker and Skamarock (2002):

$$\frac{\partial \phi}{\partial t} = \mathrm{RHS}_{\phi} \tag{1}$$

The variable ∅ 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):

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = \mathrm{RHS}_{\phi} \tag{2}$$

Where superscript represent the time step, and subscript represent the position of grid zone.

The two-order Runge-Kutta time scheme is used in MPAS-A ~~solver utilizes the physics suite from the Advanced Research WRF (ARW)model, with a particular focus on the physical configurations usedin the ARW Nested Regional Climate Model~~

~~(WRF-NRCM)applications and tropical cyclone prediction experiments.~~ as described in Gear et al. (1971):

$$\phi^* = \phi^t + \frac{\Delta t}{2} \cdot \mathrm{RHS}(\phi^t) \tag{3}$$

$$\phi^{**} = \phi^t + \frac{\Delta t}{2} \cdot \mathrm{RHS}(\phi^*) \tag{4}$$

$$\phi^{t+\Delta t} = \phi^t + \Delta t \cdot \mathrm{RHS}(\phi^{**}) \tag{5}$$

In this study, the 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 the latest version currently released.

2.3 Application of quasi double-precision algorithm in MPAS-A

~~Quasi-double-precision has been validated~~ According to Equation Eq.(3), (4) and (5), it can be observed that in the time integration ~~process of differential equations (Møller et al. 1965). The primary objective of this section is to demonstrate how to apply~~ scheme, each step involves the process of adding tends on the basic field ϕ^t. In numerical models, the basic field is generally much larger than the tends, 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 quasi double-precision algorithm ~~to the time integration process in MPAS-A~~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 in spatial discretization remains uncertain, therefore, we will not apply it in this context.

~~The application of quasi-double-precision algorithm in MPAS-A. (a): The application framework of MPAS-A. (b): An example of adding quasi-double-precision algorithm in MPAS-A.~~

~~By analyzing the application framework of MPAS-A (Figs. 3a), it can be observed that the modules containing~~ Based on the application principles of the algorithm, which involve the processes of adding large and small numbers as well as the time integration ~~schemes are : (1)the gravity wave and acoustic wave calculation module,and~~ process, we have established a strategy for applying the quasi double-precision algorithm within the MPAS-A. Specific improvements are provided based on

Eq.(6),Eq.(7),Eq.(8) and Eq.(9):

$$\frac{\partial V_H}{\partial t} = -\frac{\rho_d}{\rho_m}\left[\nabla_\zeta\left(\frac{p}{\zeta_z}\right) - \frac{\partial z_H p}{\partial \zeta}\right] - \eta \mathbf{k}\times\mathbf{V}_H - \nu_H \nabla_\zeta \cdot \mathbf{V} - \frac{\partial \Omega \nu_H}{\partial \zeta} - \rho_d \nabla_\zeta K - eW\cos\alpha_r - \frac{\nu_H W}{r_e} + \mathbf{F}_{\mathbf{V}_H} \tag{6}$$

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

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

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

The meaning of each variable in the equations exactly follows Skamarock et al. (~~2)the scalar transport process. This study focuses on dynamic core, involving the gravity wave and acoustic wave, so we close the scalar transport in all cases. In the gravity wave and acoustic wave, the core variables calculated through time integration are horizontal momentum at cell edge (u~~2012), so that we don't repeating explanation. For a numerical model, the most crucial variables are the prognostic variables. Therefore, In the MPAS-A model we applied the quasi double-precision algorithm to the time integration process of these prognostic variables, including horizontal momentum (V_H) , dry air density (~~rho~~$\tilde{\rho}_d$), potential temperature (~~theta~~Θ_m) and vertical velocity ~~at vertical cell faces (w)~~ (W) , that is , the process in red of Eq. (6), (7), (8) and (9) (Only the predictive equations for the dynamic core are presented here, without the scalar transport). This study focuses on 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.

~~These variables involve two time integration processes: In acoustic_steps, the small quantity is calculated, and is then added to big quantity in large-time-step-tendency, and the two parts both use quasi double-precision algorithms respectively. Figure 3 shows an application method adding u_tend to u and as an example (Figs. 3b) , other variables are the same.~~

2.4 Experimental design and configuration

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

To assess the application effect of the quasi double-precision algorithm, this study employs four test cases, including two idealized cases (Jablonowski and Williamson baroclinic wave and super-cell) and two real cases (with initial conditions generated using GFS data at 2014-09-10_00) using two different resolutions. To prevent the influence of other factors, the basic parameters of all cases are kept consistent, including the Number of acoustic steps per full RK step, config dynamics split steps, and config number of sub steps (integer), among others.

3 Results and analysis

In this section, we introduce the Spatial RMSE and MAE(the accuracy indicators), and show results across four cases, include two ideal scenarios: Jablonowski and Williamson baroclinic wave and super-cell, as well as two real case (with initial conditions generated using GFS data) using two different resolutions. By using the RMSE and MAE for quantitative comparison, the differences between the benchmark and control experiments are used to evaluate the effectiveness of the quasi double-precision algorithm in reducing round-off error.

3.1 Spatial RMSE and MAE

To quantify the difference between the simulations using SGL, QDP, and DBL, (used as the benchmark), we calculate the spatial root-mean-square error (RMSE). First, for each grid point, the temporal averages of the variables (e.g., surface pressure, 500hPa 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 double-precision experiment (DBL), following (10):

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

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, 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 Mean Absolute Error (MAE) to assess the magnitude of the difference between the control experiments (SGL and QDP) and the benchmark double-precision experiment (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 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 (11):

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|M_i - C_i| \tag{11}$$

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

As shown in Table 1 (spatial RMSE) and Table 2 (MAE), the addition of the quasi-double precision algorithm consistently improves accuracy (compared to single precision) across all cases.For specific analysis, please refer to the following contents(The results of RMSE and MAE are consistent, so to avoid duplication, only the results of RMSE are analyzed in the following text).

Table 1. **The spatial RMSE** values of surface pressure compared to DBL for cases, unit: Pa. Note: JW wave = Jablonowski & Williamson baroclinic wave;SC = Super-cell; RD-120/240 = Real data with total domain size of 120/240 km.

Case name	SGL	QDP
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	$6.68 * 10^{-2}$	$2.25 * 10^{-3}$

Table 2. **The MAE** values of surface pressure compared to DBL for cases, unit: Pa. Note: JW wave = Jablonowski & Williamson baroclinic wave;SC = Super-cell; RD-120/240 = Real data with total domain size of 120/240 km.

Case name	SGL	QDP
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	$5.52 * 10^{-2}$	$1.94 * 10^{-3}$

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), 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 in a pressure-based vertical coordinate system (Jablonowski and Williamson 2006). The ~~configuration follows the specifications published on the MPAS website~~experimental configuration is consistent with the test case presented by Jablonowski and Williamson (2006), with a time step of 450 seconds, 26 vertical levels, ~~resolution~~ total domain size of 120 km × 120 km, and an integration period of 15 days.

The ~~round-off error~~ bias begins to appear at the tenth day. Starting from the tenth day, the ~~round-off error of kinetic energy and surface pressure~~ bias of total energy and total mass caused by SGL can be reduced by using Quasi double-precision (Figs. ~~4a, 4b~~3a, 3b). Unlike SGL, where the ~~error~~ bias increases rapidly after more than 10 days, QDP has a very small ~~error~~ bias

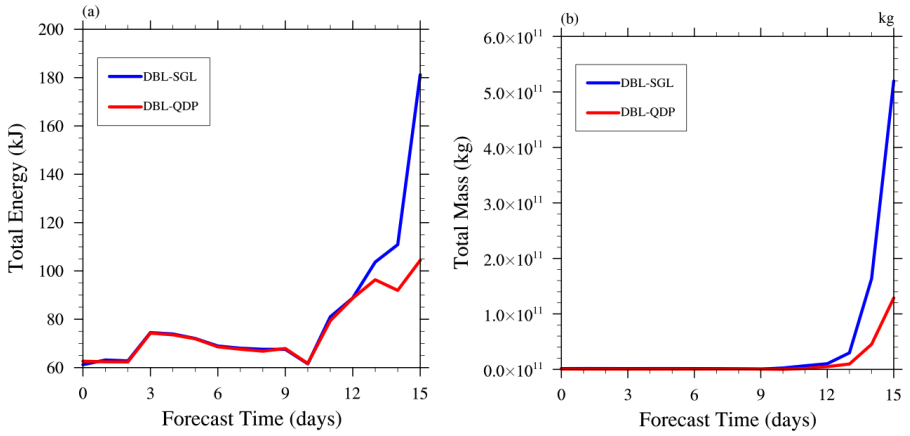

Figure 3. The time evolution of difference between DBL and SGL, as well as difference between DBL and QDP of (a)~~kinetic energy~~Total Energy, (b)~~surface pressure~~ Total Mass in case of Jablonowski and Williamson baroclinic wave.

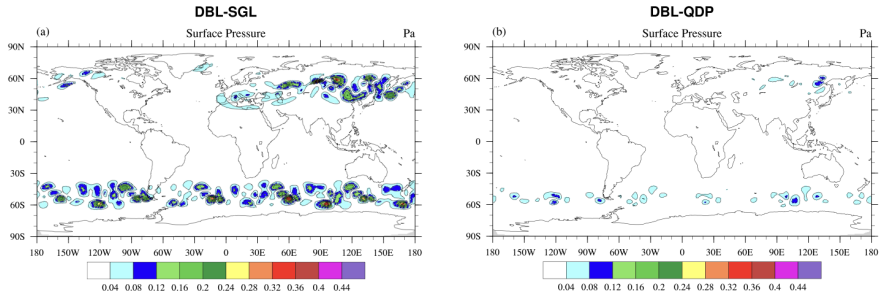

Figure 4. Spatial distributions of averaged (1-15days) difference of surface pressure (units: Pa) between DBL and (a) SGL simulations, (b) QDP simulations in case of Jablonowski and Williamson baroclinic wave.

compared to double precision. Therefore, it can be considered that QDP can be used to replace double precision in medium range weather forecast.

It can be found that SGL can increase the round-off error in all regions(Figs. ~~5a~~4a), especially in high-latitude regions, such as Southern Ocean westerly belt, its high wind speed increase ~~round-off~~ error caused by SGL, but instability caused by high wind speeds is more important. Surprisingly, the ~~round-off error~~ bias can be reduced significantly in QDP(Figs. ~~5b~~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~~, but rather that~~. Instead, the improvement effect is more pronounced in areas with larger errors. The spatial RMSE of surface pressure between DBL and SGL is ~~$3.42*10^{-}2$~~ $3.42*10^{-2}$ Pa, as well as ~~$1.09*10^{-}2$~~ $1.09*10^{-2}$ Pa between DBL and QDP, the error reduced by 68%.

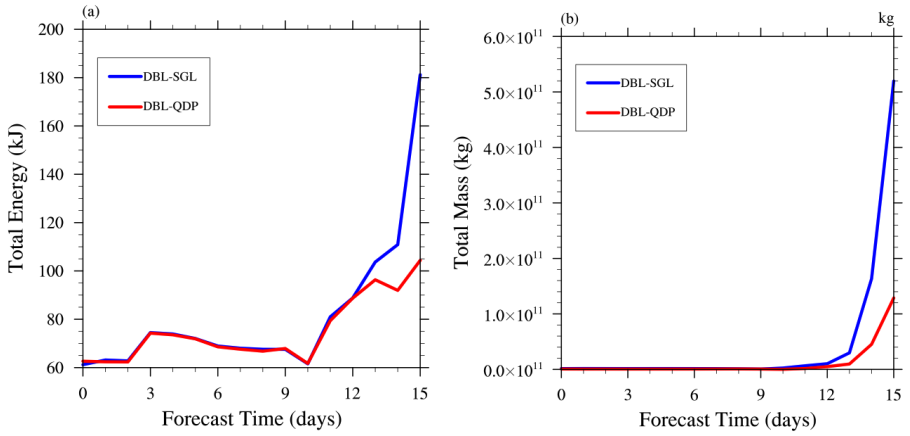

Figure 5. The time evolution of difference between DBL and SGL, as well as difference between DBL and QDP of (a) ~~kinetic~~ Total energy, (b) ~~surface pressure~~ Total mass in case of super-cell.

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 days of integration. These errors arise from factors such as rounding errors due to reduced numerical precision and energy loss during the propagation process. The quasi double-precision 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 on evaluating the improvements provided by the compensation algorithm in addressing rounding errors.

3.3 Super-cell case

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

In this case, the reduction of ~~kinetic energy in round-off~~ Total energy in error is not significant in QDP (Figs. ~~6a~~5a), except for the initial time, all others showed larger errors than SGL. But the errors of both are ~~very small and can be ignored. For surface_pressure~~ negligible. For total mass (Figs. ~~6b~~5b), the ~~round-off~~ error caused by SGL can obtain effective improvement in QDP. This improvement exists throughout the entire integration period.

Figure ~~7~~ 6 shows the spatial distribution of perturbation theta, an important variable in numerical models, when reducing the numerical precision from double (Figs. ~~7a~~6a) to single (Figs. ~~7b~~6b), it displays differences, it indicates a significant increase in round-off error. In QDP, this difference can be compensated(Figs. ~~7c~~6c). The spatial RMSE of surface pressure between DBL

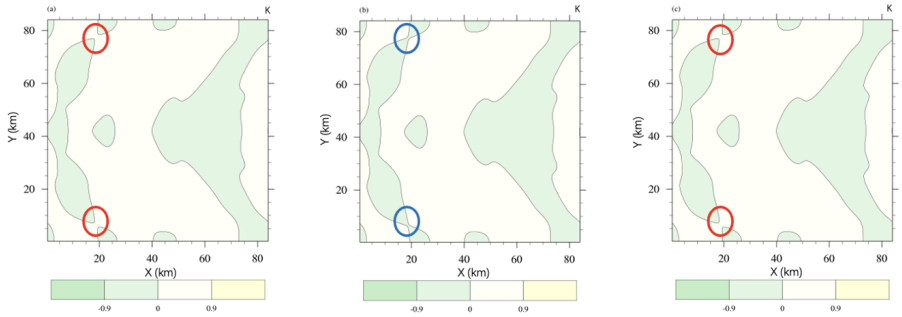

Figure 6. Perturbation theta in super-cell development at 5400s in the (a) DBL simulation, (b) SGL simulation and (c) QDP simulation (bias has reduced), unit: K, the circle ~~represent~~ represents the ~~most clear error~~pattern bias (the same color means the consistent value).

and SGL is ~~$8.95 * 10^{-}4$~~ $8.95 * 10^{-4}$ Pa, as well as ~~$2.19 * 10^{-}4$~~ $2.19 * 10^{-4}$ Pa between DBL and QDP, the error reduced by 75%.

3.4 Real data cases

In this section, we will show the results from two cases using different ~~resolution~~resolutions. The settings include a time step of 720 seconds, 55 vertical levels, ~~resolution of~~ the total domain size are 240 km × 240 km and 120 km × 120 km, and an integration period of 15 days. (Except for the resolution, all other configurations are exactly the same)

~~At the initial stages of the integration process(Figs. 8), both the SGL and QDP have minimal rounding errors in different resolutions. Differences in error~~ Consistent with the analysis presented in Section 3.2, errors are relatively small in the early stages and begin to emerge after ~~500 steps. QDP can reduce errors generated by SGL within certain integration time, although not consistently throughout. Overall, QDP demonstrates an ability to reduce rounding errorscaused by SGL~~140 hours. This increase is attributed to the accumulation of round-off errors and energy loss over time. The effects become more pronounced beyond 140 hours. Overall, the quasi double-precision algorithm demonstrates a certain level of improvement in addressing these errors. The case with ~~resolution~~ the total domain of 240 km × 240 km (Figs. ~~8a~~7a) show the larger error than 120 km × 120 km (Figs. ~~8b~~7b), and the error can be reduced in QDP caused by SGL.

Figure ~~9 and 11~~ 8 and 10 show spatial distributions of surface pressure with different resolution, the total domain aise are 240 km × 240 km (Fig. ~~9~~8) and 120 km × 120 km (Fig. ~~11~~10). The error has reduced throughout the all region, and the improvement effect is very obvious. From a spatial perspective, the case of SGL with ~~resolution~~ the total domain size of 240 km × 240 km (Figs. 9a) show the larger error than 120 km × 120 km (Figs. ~~11a~~10a), and the errors both can be reduced by QDP (Figs. ~~9a and 11a~~8a and 10a). The spatial RMSE of surface pressure with 240 km × 240 km between DBL and SGL is ~~$6.68 * 10^{-}2$~~ $6.68 * 10^{-2}$ Pa, as well as ~~$2.25 * 10^{-}3$~~ $2.25 * 10^{-3}$ Pa between DBL and QDP, the error reduced by 97%. The spatial RMSE of surface pressuse with 120 km × 120 km between DBL and SGL is ~~$6.33 * 10^{-}2$~~ $6.33 * 10^{-2}$ Pa, as well as ~~$2.25 * 10^{-}3$~~ $2.25 * 10^{-3}$ Pa between DBL and QDP, the error reduced by 96%.

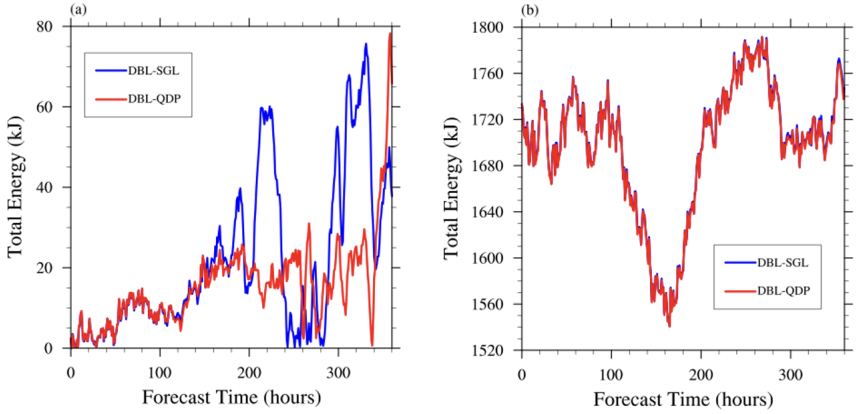

Figure 7. The ~~time~~ temporal evolution of spatially averaged difference of ~~kinetic~~ total energy between DBL and SGL, as well as difference between DBL and QDP in case of real data, with resolution of (a) 240 km × 240 km, (b) 120 km × 120 km.

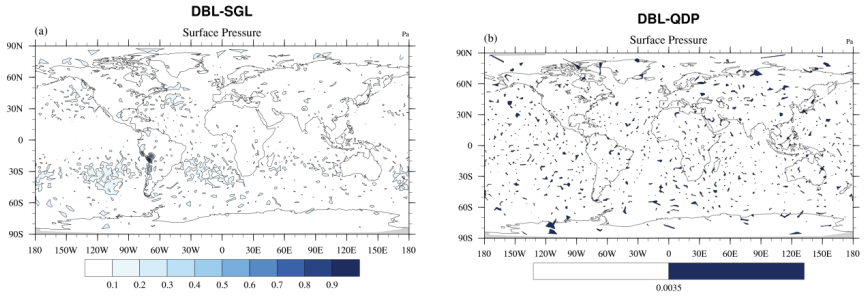

Figure 8. Spatial distributions of averaged (1-15days) difference of surface pressure (units: Pa) between DBL and (a) SGL simulation, (b) QDP simulation (~~resolution~~the total domain size: 240 km × 240 km). The RMSE of surface pressure between DBL and (a) SGL simulation is ~~$6.68 * 10^{-2}$~~ $6.68 * 10^{-2}$ Pa, (b) QDP simulation is ~~$2.25 * 10^{-3}$~~ $2.25 * 10^{-3}$ Pa.

Figure ~~10 and 12~~ 9 and 11 show spatial distributions of 500 hPa height with different ~~resolution~~total domain size, 240 km × 240 km (Fig. ~~10~~9) and 120 km × 120 km (Fig. ~~12~~11), The error improvement effect is consistent with surface pressure. The spatial RMSE of 500 hPa height with 240 km × 240 km between DBL and SGL is ~~$2.80 * 10^{-1}$~~ $2.80 * 10^{-1}$ m, as well as ~~$1.40 * 10^{-1}$~~ $1.40 * 10^{-1}$ m between DBL and QDP, the error reduced by 50%. The spatial RMSE of with 120 km × 120 km between DBL and SGL is ~~$4.35 * 10^{-3}$~~ $4.35 * 10^{-3}$ Pa, as well as ~~$1.90 * 10^{-3}$~~ $1.90 * 10^{-3}$ Pa between DBL and QDP, the error reduced by 56%.

4 ~~Conclusions and discussion~~

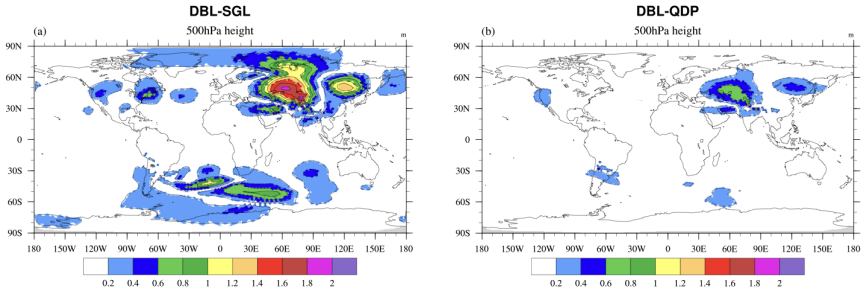

Figure 9. Spatial distributions of averaged (1-15days) difference of 500 hPa height (units: m) between DBL and (a) SGL simulation, (b) QDP simulation (~~resolution~~the total domain size: 240 km × 240 km). The RMSE of 500 hPa height between DBL and (a) SGL simulation is ~~$2.80 * 10^{-1}$~~ $2.80 * 10^{-1}$ m , (b) QDP simulation is ~~$1.40 * 10^{-1}$~~ $1.40 * 10^{-1}$ m(round-off error has reduced).

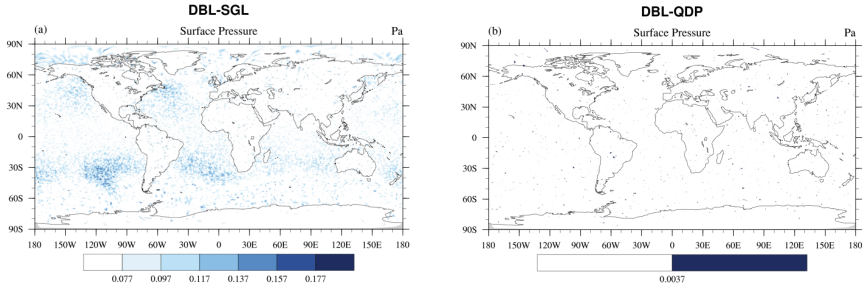

Figure 10. ~~distributions~~ Distributions of averaged (1-15days) difference of surface pressure (units: Pa) between DBL and (a) SGL simulation, (b) QDP simulation (~~resolution~~the total domain size: 120 km × 120 km). The RMSE of surface pressure between DBL and (a) SGL simulation is ~~$6.33 * 10^{-2}$~~ $6.33 * 10^{-2}$ Pa, (b) QDP simulation is ~~$2.25 * 10^{-3}$~~ $2.25 * 10^{-3}$ Pa. (The color bars in (a) and (b) are different)

~~In this study, we applied the~~ In this research, we focus on the processes of summing the basic field and trends. When the resolution is increased, the 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 quasi double-precision algorithm ~~to MPAS-A. we discovered that , the~~ more pronounced. Thus, it is evident from Figure 8 that as the resolution increases, the improvement achieved by quasi double-precision algorithm ~~can effectively reduce the errors introduced by using low precision through the iterative process of time integration. In different cases (including idealized and real data cases) , the number of iterations of the correction variables varies are different, leading to differences in error improvement.~~ also enhances.

On the other hand, it is important to note that the propagation of rounding errors is not immediately apparent over short time scales. However, as the number of iterations increases, these errors can become more significant. The quasi double-precision algorithm employs compensation mechanisms that help mitigate the propagation of these errors.

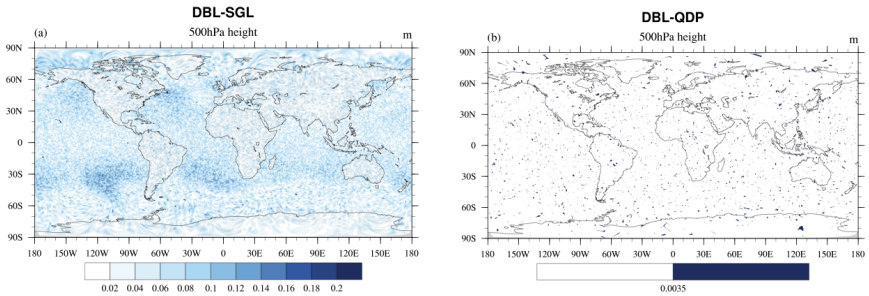

Figure 11. Spatial distributions of averaged (1-15days) difference of 500 hPa height (units: m) between DBL and (a) SGL simulation, (b) QDP simulation (~~resolution~~the total domain size: 120 km × 120 km). The RMSE of 500 hPa height between DBL and (a) SGL simulation is ~~$4.35 * 10^-3$~~ $4.35 * 10^{-3}$ m , (b) QDP simulation is ~~$1.90 * 10^-3$~~ $1.90 * 10^{-3}$ m.

3.1 Computational performance

In comparison with the SGL, although there is a slight increase in runtime, it is minimal, at only 6.0% (Jablonowski and Williamson baroclinic wave), 0.3% (Super-cell), 2.2% (Real data with resolution of 120km) and 17.8% (Real data with resolution of 240km) (Table 3). This slight increase is attributed to the addition of a small number of global variable arrays when using quasi double-precision. And compared to DBL, QDP demonstrated relatively better performance across different cases, reducing the runtime by 28.6% (Jablonowski and Williamson baroclinic wave), 28.5% (Super-cell), 21.1% (Real data with total domain size of 120km) and 5.7% (Real data with total domain size of 240km) (Table 3).

Table 3. Comparative Analysis of Computational Efficiency: DBL vs SGL vs **QDP**

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	19126 s	14765 s	**15092 s**	-21.1%	+2.2%
RD-240	1397 s	1118 s	**1317 s**	-5.7%	+17.8%

Note: JW wave = Jablonowski & Williamson baroclinic wave;

SC = Super-cell; RD-120/240 = Real data with total domain size of 120/240 km

4 Conclusions and discussion

Although the Moller method (Quasi double-precision) has been extensively employed for the temporal integration of ordinary differential equations, its application within the context of realistic numerical models remains unexplored. This study addresses this gap by presenting a novel implementation of the Moller method, thereby expanding its scope and potential impact within the field. The algorithm can compensate for round-off errors by keeping corrections in addition of large and small numbers. And in numerical models, the basic field is generally much larger than the tends, which aligns with the principles of quasi double-precision, as well as the time integration process. Based on the it, we have established a strategy for applying the quasi double-precision algorithm within the MPAS-A. Through the implementation of quasi double-precision methods, we maintain accuracy similarly to the tests using double precision and achieve comparable integration stability to the tests comparing to single precision tests. The error of surface pressure of 4 cases are reduced by 68%, 75%, 97%, 96% (see Section 3). Overall, QDP using quasi ~~double-precision~~ double- precision algorithm demonstrates higher accuracy than the SGL, suggesting the potential for applying quasi double-precision algorithm in numerical models.

We don't apply it to spatial discretization process, because 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 in spatial discretization remains uncertain, therefore, we don't consider it in this context.

While mixed-precision approaches, where certain variables retain double precision for stability (e.g., Chen et al., 2024), are common for reducing numerical precision in models, and they don't consider the error compensation. This study distinguishes itself by implementing single precision for all model variables and applying error compensation for critical variables.

When applied the quasi double-precision algorithm in MPAS-A, we achieved to reduce all double precision to single precision, although increased ~~3~~ few local variables and ~~1 array in every time-integration variables~~arrays in every time- integration variable, these have little impact on the overall memory reduction. In general, memory has been reduced by almost half, ~~while the computation increases only 2%~~with a corresponding computational increase of just 6.0%, 0.3%, 2.2%, and 17.8% in the respective cases (see section 3.4), demonstrating a substantial improvement in computational efficiency.

Nevertheless, there are some limitations to the application of quasi double-precision algorithm. Firstly, the algorithm relies on iterative process of time integration, its effectiveness is dependent on the number of time iterations, the more iterations, the better the error compensation. Secondly, although the quasi double-precision algorithm partially reduces the round-off errors of low-precision calculations, it still shows error compared to the double-precision version, making it less suitable for experiments requiring high precision. Additionally, applying quasi double-precision algorithm must bring other variables, increasing the complexity to a certain degree.

Currently, the quasi double-precision algorithm is only ~~applied~~ implemented in the time integration scheme ~~in~~ of the dynamic core of the MPAS-A model, without ~~considering tracer transport, but the process is more sensitive for the precision . Future research will attempt to apply the quasi double-precision algorithm to this part.~~ factoring in tracer transport. However, the tracer transport process involves numerous operations where large and small numbers are added together, making it more sensitive to precision requirements.

~~Currently, we~~ We have applied quasi double-precision algorithm to ideal and real data tests with low and medium resolution, the impact on high resolution has not been studied yet. On the other hand, the tracer is also a part of the atmosphere dynamic core, which describes the transport of tracer and may be more sensitive to accuracy. In the future, we will apply quasi double-precision algorithm to the tracer to analyze and study its sensitivity.

Code and data availability. Model code and plotting data related to this manuscript is available at: https://doi.org/10.5281/zenodo.13765422. Details regarding the code structure and instructions for running the code are provided in the supplementary material, which can be downloaded and viewed in Fig. S1. This figure provides a visual overview of the code organization. The information of steps how to execute the simulations can be found in README file in each test case folder.

Author contributions. JYL and LNW developed the code of applying quasi double-precision algorithm to MPAS-A and design 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 authors declare that they have no conflict of interest.

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.

References

Bauer, P., Thorpe, A., and Brunet, G.: The quiet revolution of numerical weather prediction, Nature, 525(7567):47-55, doi:10.1038/nature14956, 2015.

Cotronei, A., Slawig, T.: Single-precision arithmetic in ECHAM radiation reduces runtime and energy consumption, Geoscientific Model Development, 13(6):2783-2804, dio:10.5194/gmd-13-2783-2020, 2020.

Chen, Siyuan., Zhang, Yi., Wang, Yiming., Liu, Zhang., Li, Xiaohan., and Xue, Wei.:Mixed-Precision Computing in the GRIST Dynamical Core for Weather and Climate Modelling: Geosci. Model Dev., doi: https://doi.org/10.5194/gmd-2024-68, 2024.

Dawson, A., and Peter D, Düben.: rpe v5: An emulator for reduced floating-point precision in largenumerical simulations, Geoscientific Model Development Discussions, 10(6):1-16, doi:10.5194/gmd-2016-247, 2017.

Dawson, A., Peter D, Düben., Macleod D, A., and Palmer, TN.: Reliable low precision simulations in land surface models, Climate dynamics.DOI:10.1007/S00382-017-4034-X, 2018.

Dmitruk, B., Przemysaw, Stpiczyński.:Improving accuracy of summation using parallel vectorized Kahan's and Gill-Mller algorithms, Concurrency and Computation: Practice and Experience, doi:10.1002/cpe.7763, 2023.

Dmitruk, B., Przemysaw, Stpiczyński.: Improving accuracy of summation using parallel vectorized Kahan's and Gill-Mller algorithms, Concurrency and Computation: Practice and Experience, doi:10.1002/cpe.7763, 2023.

Gear, C. W.: Numerical initial value problems in ordinary differential equations[M]. Englewood Cliffs, N.J: Prentice-Hall, 1971.

Gill, S.: A process for the step-by-step integration of differential equations in an automatic digital computing machine, Proc. Cambridge Philos. Soc. 47, 96-108, 1951.

Hatfield, S., Peter D, Düben., Palmer, T., and Chantry, M.: Accelerating High-Resolution Weather Models with Deep-Learning Hardware, Platform for Advanced Scientific Computing, doi:10.1145/3324989.3325711, 2019.

Higham, N.: Accuracy and Stability of Numerical Algorithms. SIAM; 1996.

Hugo, Banderier., Christian, Zeman., David, Leutwyler., Stefan, Rüdisühli., Christoph, Schär.: Reduced floating-point precision in regional climate simulations: an ensemble-based statistical verification: Geosci. Model Dev., 17, 5573–5586, 2024 doi: https://doi.org/10.5194/gmd-17-5573-2024, 2024.

Kahan, W.: Pracniques: further remarks on reducing truncation errors. ACM, 1965.

Klemp J B, Skamarock W C, Dudhia J.: Conservative split-explicit time integration methods for the compressible nonhydrostatic equations. Mon Wea Rev, 135: 2897-2913, 2007.

Møller, O.: Quasi double-precision in floating point addition, Bit Numerical Mathematics, 5(1):37-50, doi:10.1007/BF01975722, 1965.

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, Monthly Weather Review, MWR-D-17-0257.1, dio:10.1175/MWR-D-17-0257.1, 2018.

Oriol Tintó, Prims., Acosta, M. C., Moore, A. M., Castrillo, M., 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, Geoscientific Model Development, 12(7):3135-3148, dio:10.5194/gmd-12-3135-2019, 2019.

Paxton, E. A. D. A. M., CHANTRY, M. A. T. T. H. E. W., KLOWER, M. I. L. A. N. SAFFIN., and L. E. O. PALMER, T. I. M.: Climate Modeling in Low Precision: Effects of Both Deterministic and Stochastic Rounding, Journal of Climate, 35(4):1215-1229, 2022.

S, Gill.: A process for the step-by-step integration of differential equations in an automatic digital computing machine, Mathematical Proceedings of the Cambridge Philosophical Society, doi:10.1017/s0305004100026414, 2008.

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 Tesselations and C-Grid Staggering, Monthly Weather Review, 240(9):3090-3105, doi:10.1175/MWR-D-11-00215.1, 2011.

Thompson., Robert, J.:Improving round-off in Runge-Kutta computations with Gill's method, Communications of the Acm, 13(12):739-740.DOI:10.1145/362814.362823, 1970.

Tomonori, Kouya., Hideko, Nagasaka.: On the correction method of round-off errors in the Yang's Runge-Kutta method, The Japan Society for Industrial and Applied Methematics, 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, Monthly Weather Review, 145, 495-502, doi:10.1175/MWR-D- 16-0228.1, 2016.

Wicker L J, Skamarock W C.: Time-splitting methods for elastic models using forward time schemes. Mon Wea Rev, 130: 2088-2097, 2002.