Responses to Topic editor's Comments for Manuscript egusphere-2024-2986

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

Addressed Comments for Publication to

Geoscientific Model Development

by

The Authors

Dear Editor,

Enclosed herewith is the updated edition of our earlier submission titled “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” assigned the manuscript number egusphere-2024-2986. We extend our sincere appreciation to you for offering invaluable feedback, which has significantly improved the overall quality of our manuscript. In this updated version, we have carefully addressed the suggestions provided by you. Herein, you will discover a summary of the modifications implemented, along with a thorough response to the your comments.

Sincerely,

The Authors

Note: To enhance the legibility of this response letter, editor’s comments are typeset in boxes. Rephrased or added sentences are typeset in color. The respective parts in the manuscript are highlighted to indicate changes.

Response to the editor

Comment 1.1

Clarify Vectorization Benefits: Could you briefly discuss potential vectorization strategies applicable to QDP in a future work section?

Response 1.1:

We express our gratitude to you for your review and valuable feedback. In response to your comment regarding the "potential vectorization strategies," we have included a more thorough discussion in the revised version. This includes an exploration of vectorization strategies applied to QDP and provides an outlook for future research in this area. It can be seen as follows (Line 341-351 in manuscript):

> In future work, we plan to explore vectorization strategies for QDP algorithm, building on successful implementations of vectorized compensated summation algorithms. Dmitruk et al. (2023) have efficiently vectorized 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 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 QDP algorithm in HPC environments.

Comment 1.2

Expand on Algorithm Limitations: Please provide a brief discussion on any known limitations of QDP in spatial discretization.

Response 1.2:

Thank you for your insightful suggestion. We have expanded the manuscript to include a brief discussion on the limitations of the QDP method in spatial discretization. This added section can be found as follows(Line 326-334 in manuscript):

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

Comment 1.3

Enhance Figure Captions: Ensure all figure captions are concise and directly relate to the content of the figures.

Response 1.3:

Thank you for the suggestion regarding the figure captions. We have revised the captions to ensure they are more concise and directly aligned with the content of the figures. This improvement aims to enhance the clarity and effectiveness of the visual information presented. The revised version is as follows.

Figure 1. Iterative Process of QDP algorithm in Step-by-Step Integration.
Figure 2. The QDP algorithm with magnitude preconditioning for Identifying large and small numbers.
Figure 3. Time evolution of differences in (a) total energy and (b) total mass between DBL, SGL, and QDP (proposed) in the JW wave. Both figures highlight QDP's superior error compensation, especially over extended integration periods.
Figure 4. Spatial distributions of 1–15 day averaged surface pressure differences (Pa) between DBL and (a) SGL, (b) QDP in the JW wave case. QDP reduces errors more effectively, particularly in mid- to high- latitude regions.
Figure 5. Time evolution of differences in (a) total energy and (b) total mass for the supercell case: DBL vs. SGL and DBL vs. QDP. The results highlight QDP algorithm's effective error compensation, with benefits becoming more pronounced over time.
Figure 6. Perturbation theta at 5400 s in supercell development: (a) DBL, (b) SGL, and (c) QDP (unit: K). The circle highlights 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.
Figure 8. Spatial distributions of averaged (1–15 days) surface pressure differences (Pa) between DBL and (a) SGL, (b) QDP (domain size: 240 km × 240 km). The Spatial RMSE is 6.68×10^{-2} Pa for (a) and 2.25×10^{-3} Pa for (b), highlighting QDP algorithm's significant error reduction across regions and its effectiveness in correcting errors by several orders of magnitude.

Figure 9. Spatial distributions of averaged (1–15 days) 500 hPa height differences (m) between DBL and (a) SGL, (b) QDP (domain size: 240 km × 240 km). The Spatial RMSE decreases from 2.80×10^{-1} m in (a) to 1.40×10^{-1} m in (b), indicating notable spatial error reduction with QDP algorithm.
Figure 10. Distributions of averaged (1–15 days) surface pressure differences (Pa) between DBL and (a) SGL, (b) QDP (domain size: 120 km × 120 km). The Spatial RMSE decreases from 6.33×10^{-2} Pa in (a) to 2.25×10^{-3} Pa in (b). Note: color bars differ between (a) and (b). QDP significantly reduces errors across all regions.
Figure 11. Spatial distributions of averaged (1–15 days) 500 hPa height differences (m) between DBL and (a) SGL, (b) QDP (domain size: 120 km × 120 km). The Spatial 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.

Comment 1.4

Technical Terminology: Include a list of abbreviations or a glossary for better accessibility to a wider audience.

Response 1.4:

Thank you for your helpful suggestion. We have added a list of abbreviations and a glossary to the manuscript in the relevant section. The relevant details can be found in Table 1 and Table 2 of the manuscript.

Comment 1.5

README File: Consider adding a section on reproducing the study's results, including specific commands used.

Response 1.5:

We appreciate your suggestion to include a section on reproducing the study's results. In the revised version of the manuscript, we have added a detailed content in the "code data availability" part. This addition aims to enhance the transparency and reproducibility of our work. It can be seen as follows.

The provided repository 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)

- DBL and SGL:
  - GitHub: `https://github.com/MPAS-Dev/MPAS-Model/releases/tag/v8.2.1` (last access: 26 December 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.

  – Zenodo: `https://doi.org/10.5281/zenodo.14576893` (located in `code_and_data/model/DBL/` and `code_and_data/model/SGL/`)

- SGL-QDP (proposed):
  – Zenodo: `https://doi.org/10.5281/zenodo.14576893` (located in `code_and_data/model/SGL-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`: Use the following command: make ifort CORE=init_atmosphere
- Clean previous builds for `atmosphere`: (if necessary) Use the following command: make clean CORE=atmosphere
- Compile `atmosphere`: Use the following command: make ifort CORE=atmosphere
- Compile in Single Precision: 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/` (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 at `https://doi.org/10.5281/zenodo.14576893` (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, ensure to adjust the number of nodes

in the script according to the available computational resources to optimize performance.

4. Visualization and Post-Processing Code: In order to reproduce the figures in this paper, follow the instructions below:

- Figure 3 and 4: Run the NCL script by `ncl time.ncl` and `ncl spatical.ncl`. Navigate to the directory: `code_and_data/test/c2_DBL/`.
- Figure 5 and 6: Run the NCL script by `ncl time.ncl` and `ncl spatical.ncl`. Navigate to the directory: `code_and_data/test/c5_DBL/`.
- Figure 7a, 8, and 9: Run the NCL script by `ncl time.ncl` and `ncl spatical.ncl`. Navigate to the directory: `code_and_data/test/c7_240km_DBL/`.
- Figure 7b, 10, and 11: Run the NCL script by `ncl time.ncl` and `ncl spatical.ncl`. Navigate to the directory: `code_and_data/test/c7_120km_DBL/`.