Reviwer 1

This manuscript presents stage one of a multi-tiered plan to support heterogeneous (mixed CPU/GPU) architectures for running the ICON model. The authors utilize GT4Py, a domain-specific language, to modernize the ICON dynamics core from the existing Fortran code base. The outcome is a more performant code, which is also easier to read and develop compared to the equivalent Fortran implementation. The paper is well written and well reasoned, demonstrating promising results that are on par with the current state of GPU-ready Earth System modeling. I recommend that this manuscript be published, as I have only a few minor questions and technical corrections to suggest.

First, I want to commend the authors for their attention to (a) the hardware-based challenges that arise when running these models at scale, and (b) the importance of robust testing. In my experience, these topics are not typically the most exciting to discuss, but they are essential considerations for any group undertaking a similar effort.

We sincerely thank the reviewer for taking time to reviewer the paper and for appreciating our work.

Minor Comments:

Introduction

1. Paragraph 3: It may be helpful to include node counts when discussing how much of the machine each example used. This additional detail would provide useful context, especially as future machines come online.

   We agree with the reviewer that additional details would be useful. However, adding them to the same paragraph would make the introduction quite involed with details on hardwares. We have therefore repharsed the paragraph with only relevant details and have suggested interested readers to look at Table 1 in Klocke et al. (2025) for more details. Please see lines 62-70 in the revised version.

   Paragraph 6: As noted above, I appreciate the discussion highlighting barriers to running these models at scale.

   Thanks again!

Section 2

1. Not strictly necessary, but it may add valuable context for readers if the authors note that Fortran compiler support is increasingly being deprioritized by vendors, which makes supporting legacy codes on new machines more challenging.

   We appreciate the reviewer for reminding this point. It has been added in the revised introduction on line 89 when discussing softare reliability.

Section 3

1. I may have missed it, but it was unclear whether the plan is to transition entirely away from Fortran after deliverable 3. Could the authors clarify how much of the original Fortran code is expected to remain in the model (e.g., 10%, 25%, or more)?

Thanks for asking this question. We do aim for a Fortran-free driver/infrastructure code in deliverable 3. The no-Fortran infrastructure so far is complete for idealized simulations using dynamical core alone.

As for the model components, we will (likely) keep land-surface parameterization and Ocean in Fortran in the foreseable future.

Section 4

1. General comment: The authors should verify that each "Listing" is correct and that the code blocks would work as expected.

   Thanks for pointing it out. The listings have been checked again. They are correct in what they represent but we do not expect them (e.g., Listing 1) to work by simply copy and pasting.

2. Section 4.3: If I understood correctly, the ported code was tested to within a tolerance error, and bit-for-bit (BFB) agreement was not strictly enforced. Was any BFB enforcement attempted during porting? If not, could the authors justify their decision not to enforce BFB?

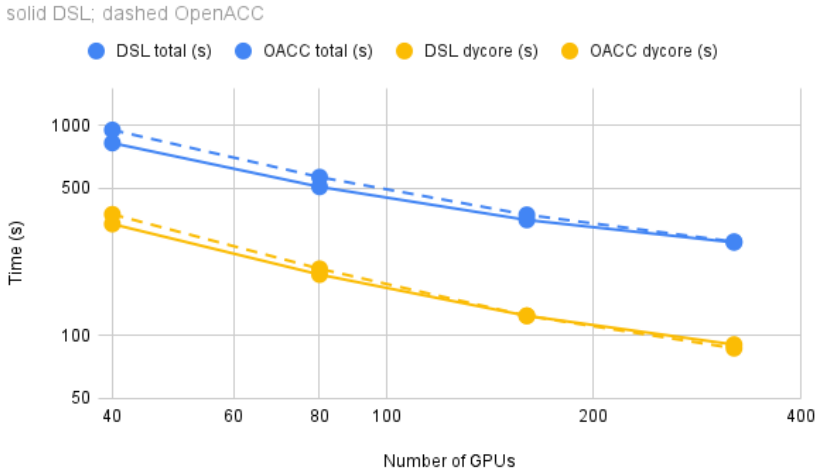
   Thanks for asking this question. Enforcing BFB agreement was not deemed as a sustained testing strategy within the project since we had intentions to combine stencils into larger GT4Py operators/programs for performance tuning through DaCe. Maintaing such a debugging mode would have been difficult. That said, we did attempt BFB agreement during dynamical core porting- we used the -iEEE flag to prevent Fortran from doing non-IEEE 754 compliant transformations of floating point computations. We also switched off fused multiply adds (FMA) on both sides (Fortran and generated CUDA code in gt4py). BFB was achieved in the vast majority of stencils except for a few, even though the codes were correct.

3. Figure 5: Did the authors conduct experiments with runs well beyond 15 timesteps to confirm that the relative error does indeed stabilize?

   Yes, we did and the errors did stabilize.

4. Figure 7: Did the authors examine this data using a log-log plot? If so, was the observed trend not quite linear?

   You are right. Here's the log-log plot for your reference.


solid DSL; dashed OpenACC
DSL total (s)
OACC total (s)
DSL dycore (s)
OACC dycore (s)
1000
500
100
50
Time (s)
40
60
80
100
200
400
Number of GPUs