Review of https://doi.org/10.5194/gmd-2024-92.

Title of the Manuscript: Exploring a high-level programming model for the NWP domain using ECMWF microphysics schemes

Manuscript ID: gmd-2024-92

Date: 14/6/2024

Summary of the Manuscript

The paper describes how to use the GT4Py library to implement a representative physical parametrization scheme and its related tangent-linear and adjoint algorithms from the IFS. The main objectives of the study are to demonstrate the correctness and performance-portability of Python rewrites with GT4Py against the reference Fortran code and various ported variants. The paper is part of a larger effort to port weather and climate models to Python with GT4Py, with a particular focus on the IFS prognostic cloud microphysics scheme.

The methods involve benchmarking prototype codes on three different HPCs with diverse hardware and software configurations to ensure robust execution and competitive performance. This includes comparing single and double precision variants. The results show good portability and reasonable performance of the Python rewrites with GT4Py across all tested scenarios.

Overall this paper is a useful contribution towards efforts to achieve performance portability and future-proofing of our model codes. Although I have raised some questions about the scientific correctness of the transferred code, these issues are not essential to the main goal of the paper – which is to explain the software aspects.

GMD aspects for consideration

Does the paper address relevant scientific modelling questions within the scope of GMD?
Yes. This is a development and technical paper detailing technical implementation of a Domain Specific Language.

Does the paper present a model, advances in modelling science, or a modelling protocol that is suitable for addressing relevant scientific questions within the scope of EGU?
Yes. The application here would lead to improved modelling of the atmosphere and physical parametrizations in particular.

Does the paper present novel concepts, ideas, tools, or data?
Yes. The GT4Py tooling is a novel approach to deal with performance portability.

Does the paper represent a sufficiently substantial advance in modelling science?
Yes. The capability demonstrated could potentially lead to more performant and portable codes.

Are the methods and assumptions valid and clearly outlined?
Yes, though some suggestions for improvements have been provided below.

Are the results sufficient to support the interpretations and conclusions?
Yes, though some suggestions for improvements have been provided below.

Is the description sufficiently complete and precise to allow their reproduction by fellow scientists (traceability of results)?
Yes. The codes that have been used to produce these results are accessible and their location has been provided.

Do the authors give proper credit to related work and clearly indicate their own new/original contribution?
Yes, I believe so.

Does the title clearly reflect the contents of the paper?
Yes

Does the abstract provide a concise and complete summary?
Yes

Is the overall presentation well structured and clear?
Yes

Is the language fluent and precise?
Yes

Are mathematical formulae, symbols, abbreviations, and units correctly defined and used?
Yes

Should any parts of the paper (text, formulae, figures, tables) be clarified, reduced, combined, or eliminated?
I have suggested some additional comment would be useful for the algorithms.

Are the number and quality of references appropriate?
Yes.

Is the amount and quality of supplementary material appropriate?
Yes.

General Comments

1. Line 15: 'reduced precision' is referred to throughout the manuscript, but this is specifically 32-bit IEEE (single precision). As GPUs in particular can (and most Fortran compilers in general cannot currently) exploit other floating point models, e.g. 16-bit IEEE or 32-bit bfloat, I think it would be better to be specific about which precision. As a general question I wonder if you are well placed to explore these other floating-point models with GT4Py?

2. Line 96: You discuss a motivation for using the particular CLOUDSC schemes is that they are representative of the computational patterns in physical parametrizations. It would be good to be explicit about what those patterns are. In contrast, on line 398, you talk about not all patterns in parametrizations being natively supported.

I would suggest part of this discussion should clarify that the patterns under consideration with CLOUDSC are based on the spatial gridded structure (and in the column), while other parametrizations may use 'pseudo-dimensions' such as number of spectral bands, land surface types, or even moments or bins for more complex microphysics schemes. These may then require additional computational motifs and looping structures to extract optimal performance on different hardware.

3. Line 139/Algorithm 1: I'm not entirely sure the details of the TaylorTest are necessary for this paper. However, if you wish to keep it in then please could you add some brief inline documentation (as you might with real code) as to what the algorithm is doing at each stage. It is quite a difficult algorithm to read without it.

4. Line 188/Figure 2: The example provided here is for the 2D horizontal Laplacian which uses a stencil accessing horizontal neighbouring columns, but no vertical neighbouring grid points. This is the opposite of the microphysics stencil which does not require access to horizontal neighbours, but does require vertical neighbours (due to sedimentation of hydrometeors). Since this paper is specifically applying the method to the latter, it would be best if the example related to that. At the very least, having some text alongside the discussion of Figure 2 to explain this would be helpful.

5. Line 285: I'm not sure why you use the 'allclose' method to determine if a particular tolerance has been met rather than calculating and reporting the error. That said, my (limited) understanding of the allclose method is that only one of rtol and atol is sufficient to pass the test. As a result, the other can be made arbitrarily small and still pass the test. E.g.

   ```
   >>> x=1.0 + 0.99e-12
   >>> y=1.0
   >>> np.allclose(y, x, atol=1e-12, rtol=0)
   True
   >>> np.allclose(y, x, atol=0, rtol=1e-18)
   False
   >>> np.allclose(y, x, atol=1e-12, rtol=1e-18)
   True
   ```

   So how did you use this function to arrive at the quoted numbers? (Presumably one would want to use the first two methods demonstrated independently for each tolerance.)
   I think it would be more informative to simply provide values of max(abs(x-y)) and max(abs(x-y)/abs(x)) (where x is the reference data and y is the rewrite).

6. There seems to be more to investigate here to ensure correctness. The previous comment would go some way to helping with this, but the fact that the relative tolerance remains the same, but the absolute tolerance is 2 orders of magnitude larger suggests that the evolution of the experiment has changed. This could be a symptom of perturbation error growth where a small change leads to a different branch of the code being followed. To add some insight into this, it would also be useful to add more detail about the scientific set up of the dwarf being tested - what are the initial conditions and

timestepping - or where is this described?

7. Line 291: Was machine epsilon to single precision used in these tests?

8. Line 355: Could you say how performance engineering would be done with GT4Py? Would this be in the 'Optimizations' step in Figure 2?

9. Line 397: As per comment 2 above, it would be good to expand on what other patterns might be needed that aren't natively supported.

Specific Comments

1. Line 17: 'has become' should be 'became'.
2. Line 56: References for the GungHo dynamical core can be found at https://doi.org/10.1002/qj.3501 and https://arxiv.org/abs/2402.13738
3. Line 103: 'slightly polished' could perhaps be a bit more informative. What needed to be done? Was it purely cosmetic?
4. Line 141: I think there is a missing δ in the first inner product.
5. Line 142: 'hearth' should be 'heart'.
6. Line 180: 'scientists is exposed' should be 'scientists are exposed'.
7. Line 237: 'it builds' should be 'does it build'.
8. Line 237: It's not clear to me what 'grid-aware' means in this context. Could you be specific?
9. Line 296: I don't think NPROMA has been defined.