Geosci. Model Dev. Discuss., 7, C3469–C3472, 2015
www.geosci-model-dev-discuss.net/7/C3469/2015/
© Author(s) 2015. This work is distributed under
the Creative Commons Attribute 3.0 License.

Geoscientific Model Development Discussions
Open Access

Interactive comment on "NEMOTAM: tangent and adjoint models for the ocean modelling platform NEMO" *by* A. Vidard et al.

A. Vidard et al.

vidard@imag.fr

Received and published: 3 March 2015

Dear Reviewer,

Thank you for your numerous valuable comments, we tried to answer properly to all of them. Some specific answers are listed below.

- *"The development of an adjoint for a complex and nonlinear model, such as NEMO, is notoriously difficult. Algorithmic differentiation (AD) tools exist that can greatly facilitate the derivation. However, since NEMO has not been developed in compatibility with an AD tool":* There is a misunderstanding in the reason why we did not go through the automatic way. This is not due to the way NEMO is coded, we actually did a first attempt using tapenade and obtained a satisfactory version of the tangent and adjoint models in a reasonable amount of time. The main reason is due to the original motivation of building a multi-resolution incremental 4D-Var, as mentioned in our answer to the first reviewer. We modified the relevant paragraph to make this message clearer.

- *“Furthermore, I feel that much of the mathematical description is - although relevant - well-known and could be replaced with suitable references.”* We kind of disagree, the paper is short enough that we can leave this description in place, it makes understanding easier for readers not familiar with the topic without disturbing readability

- *“Section 4 would be greatly improved by placing emphasis in each subsection on computational aspects specific to NEMOTAM”:* Good point, we added some comment about computational aspect in each subsection.

- *“Although detailed discussion of the dynamics implied by the diverse applications is not required here, confirmation that NEMOTAM output is sensible is certainly suitable, but is missing from section 4.2.”:* indeed, this paragraph was lacking details, we added some more comments about what the figure shows and an additional figure about the evolution of the amplifying factors respect to model resolution and time window length.

- *“Since NEMOTAM is hand-coded it would be helpful if the authors would offer some concluding remark on the flexibility of the current release. Although it is hinted that “more flexibility” would be “very beneficial” in the conclusions section, I feel it is appropriate to give a more explicit indication of the potential for NEMOTAM to be applied (to sensitivity, assimilation, stability investigations etc.) under different experimental configurations. I feel that confirming some degree of existing flexibility is important in ensuring that the substantial effort invested in NEMOTAM translates to a useful scientific contribution.”:* indeed, this is a drawback of hand-coded TAM, but it offers some flexibility as it is, we added comment on that in the conclusion.

- *"Is the oscillatory advection term retained in the full nonlinear model but neglected by the gradient computations (i.e in both the TLM and the adjoint)? I am confused by the justification of the choice here, since the exactness of the adjoint with respect to the full nonlinear model is relevant":* This is actually a tricky bit. There is no "right way" to differentiate a non-oscillatory scheme since it is highly non differentiable. Doing it as an algorithmic differentiation tool would (retaining the non linear branching) is clearly not the right way to do as the result proves highly unstable. Both other approaches (ignoring the limiter or adding it in the tam) induce some errors and it is not clear if one is always better than the other. However the latter make the adjoint and tangent tests to fail since it induces some non linearities. My wild guess is that for shorter period of time the former is better and for longer you'd better add the limiter. But so far, my experience is that, even with few month of seabass1/12° configuration integration the absence of the limiter in TAM is not causing trouble. During the revision of this paper we added the limiter (not available in the distributed version) and no significant impact were seen on the computed sensitivities.

- *" Please elaborate a little on what is meant by "well-chosen" (i.e choice is determined by resources? nonlinearities in the code? etc.). "* see below.

- *"I may have misunderstood this part of the discussion, but linearly interpolating between checkpoints to relieve storage demands seems drastic. Please included additional citations if this approach is taken in other model frameworks."* It is indeed a bit drastic, even though it does not seem to a have had a significant impact so far in our applications. My guess is that it is a common practice to all TAM codes that are used with incremental 4D-Var (ROMS, IFS, ARPEGE, . . .).

  ' Although this simplification does not appear to significantly impact the TAM for

the test cases referenced in table 2, it will surely become important in other experimental configurations; for example where checkpoints are spaced further apart during longer integrations. “ Maybe our use of the word “checkpoint’ is a bit misleading, even if we store at each checkpoint, there will be some recomputation involved. Only a limited subsample of direct variables are stored (the pronostic ones and one or two diagnostic that are expensive to compute). For a global configuration that amounts to about 12 3D variables. They are stored on disk, so this is indeed a limitation, depending on the disk space available, but it is not that problematic. The real drawback is the IO overhead implied and it can be significant. However this will be alleviated by the presence of a IO server available now in NEMO (but not yet used for NEMOTAM trajectory).

“Could the authors not employ a higher level checkpointing scheme here?” No, it would require to be able to rerun the full non linear model, which is what we wanted to avoid and that prevented us to use algorithmic differentiation in the first place. We modified this paragraph

- *“I found section 3.3 confusing. I think the aim is to define an error measure for approximating the full model physics in the generation of the TLM but must admit that I can’t see how this is provided by E in Eq 9. Are the I components of " in Eq 7 related to different approximations made to the nonlinear model M ? And then is it necessary to assume linearity to obtain Eq 8?”* Sorry, the equations here were not quite right, I hope this is clearer now.

The other comments, although relevant do not require a detailed answer. We accounted for all of them.

Interactive comment on Geosci. Model Dev. Discuss., 7, 6705, 2014.

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

Discussion Paper