Geosci. Model Dev. Discuss.,
doi:10.5194/gmd-2016-3-RC2, 2016

Geoscientific Model Development Discussions
Open Access
EGU

Interactive comment on "Evaluating Statistical Consistency in the Ocean Model Component of the Community Earth System Model (pyCECT v2.0)" *by* A. H. Baker et al.

Anonymous Referee #2

Received and published: 26 March 2016

General comments:

Kudos for addressing an important and often overlooked challenge within geophysical model development - how to evaluate software and hardware changes being made to a chaotic system.

We need more work like this that is frank about shortcomings in model software quality.

Keep in mind that testing for bit for bit reproducibility across code changes is an extremely useful technique during model development. A large proportion of code changes should have no effect on model output. Since the test for BFB reproducibility can be done very easily it allows code changes that do not change results to be

Printer-friendly version

Discussion paper

merged quickly.

Specific comments:

In the introduction the exact circumstances in which pyCECT should be used are not made clear. I suggest adding a comment to indicate when pyCECT should _not_ be used. For example all code changes can be divided into commits that should change results and those that shouldn't. Using pyCECT to evaluate a change that shouldn't change results (but does) would be a mistake.

A good example of this is at 10: "or CESM-POP, even selecting a different number of cores on the same architecture results in non-BFB identical output". There are two reasons why this could happen 1) irreproducibility introduced by nondeterminism in MPI communications and 2) bugs in any of domain decomposition/ halo handling and possibly indexing errors. The correct way to handle this is to remove the 1) for example using "An order-invariant real-to-integer conversion sum" Hallberg and Adcroft 2014 and then fix the bugs in 2). So in fact I don't think changing domain decompositions is a situation where pyCECT should be used, this is just masking a problem.

In regard to this it may be worth noting that unexpected changes in model results across identical runs is an indicated of a software bug/problem (caused by, for example, invalid reads of uninitialized memory). The behavior of these bugs is unknown, they may have very little impact most of the time (otherwise they would have been found) but occasionally significantly change results. This can all depend, for example, on the contents of uninitialized memory.

Similar to the above points. pyCECT should not be used as a substitute for proper (unit) testing. You give the example of introducing a new solver to improve performance. It would be tempting to skip the unit testing and use pyCECT with the new solver and declare that it's bug-free because the results are not statistically different.

The value of section 4.2 is questionable given the above points. My conclusion from

this would be the opposite of yours. i.e. it shows that the simulation differences are noticeable when there should be none.

Technical notes:

The plot layout should be improved. For example, in figure 9, can you include a short description or label for the color bar in the plot itself. Also, instead of labels like convect_diff*10, t_advect=tw_lim can you use descriptive names.

Thoughts (but no changes required):

It would be nice to have some mention of the effects of model resolution here. Increasingly climate models are using ocean components higher than 1deg. Wouldn't it make sense to perform pyCECT analyses on runs which are as low as possible resolution (for performance reasons)? if so what are the constraints on using pyCECT in this way?

Interactive comment on Geosci. Model Dev. Discuss., doi:10.5194/gmd-2016-3, 2016.

Printer-friendly version

Discussion paper