Geosci. Model Dev. Discuss.,
doi:10.5194/gmd-2016-142-AC1, 2016

Geoscientific Model Development Discussions

Interactive comment on "A new and inexpensive non-bit-for-bit solution reproducibility test based on time step convergence (TSC1.0)" *by* Hui Wan et al.

Hui Wan et al.

hui.wan@pnnl.gov

Received and published: 21 October 2016

We thank the referee for the insightful comments and suggestions. Our responses are detailed below.

Comment: *Apologies for being so late with my initial comments. Agree with other reviewers that the paper is overall well-written and clear. I do have some questions and concerns, which are outlined below.*

In the test scenario given the drastically shortened simulation length (5 minutes) with much shorter time steps (1 or 2 seconds), how often are the physical parameterizations (radiation and non-radiation physics) executed? Is it only once for the entire run? If only

once, is this a weakness in the overall test design?

Response: Simulations presented in the discussion paper had all parameterizations calculated every time step except for radiation which was called only once. We have repeated the simulations with radiation calculated every other time step (i.e., using the same time step ratio between radiation and the other parameterization as in the default model). We found that the TSC results were similar to those in the discussion paper in the sense that the simulations that were expected to "pass" showed typical $\mathcal{P}_{\min}$ values between a few percent and $\sim$20% during a model time of 30 minutes, while those expected to "fail" showed $\mathcal{P}_{\min}$ values substantially smaller than 1% after a short (few-minute) spin-up.

It is worth noting that radiation is the only part in the current atmosphere model code that contains intentionally introduced randomness at magnitudes way beyond the level of rounding error. The radiation code uses a pseudo random number generator, and the seeds for the random number generator are chosen from the least significant digits of the pressure field. This effectively introduces state-dependent noise to the numerical solution, and is one of the reasons for the very rapid growth of initial perturbation (see also our response to respective comments below). In the revised manuscript, we present both sets of simulations (i.e., with radiation called at every other time step or only once), and include a discussion on the impact of noise on the utility of the TSC method.

Comment: *Are all of the outputs from the physical parameterizations that are used in the dynamics applied as tendencies rather than adjustments? Presumably yes, since the effects of any parameterization that applies its effects as a hard adjustment will not be mitigated by a much shorter time step.*

Response: Yes, in the version of CAM5 we used in this study, the impacts of the parameterized physics are provided as tendencies to the dynamical core. Within the physics parameterization suite, however, processes are calculated with sequential

splitting meaning that the tendencies from one parameterization are used to update the model state variables before those variables are passed onto the next parameterization. The sequential splitting still causes large time integration error when used in combination with long time steps (as is the case in CAM5 which uses a 30-minute time step for the coupling between different parameterizations and between physics and dynamics), because the splitting allows individual processes to operate in isolation for a long time (i.e., one time step) without considering the possible interactions between different processes.

Comment: *Is it true that the very rapid growth of a perturbation is due entirely to the physical parameterizations rather than the dynamics? If so, it would be good to point this out specifically, meaning that more traditional means of code verification could still be applied for changes to the dynamical core, assuming the ability to run the model adiabatically.*

Response: Yes, we clarify in the revised manuscript that the rapid growth is indeed due to the physics parameterizations. Perturbation growth test performed with the spectral transform dynamical core indicated RMS temperature difference on the order of $\mathcal{O}(10^{-12})$ by the end of the second model day. We have not done many simulations with the dynamical-core-only configuration, but given such small magnitudes of RMS temperature difference and the rather slow growth, we expect that the original test strategy is still applicable to and useful for testing of the dynamical core.

Comment: *Page 2, #50: Regarding the PerGro test using CAM4, presumably the test always fails due to Condition 1 from Rosinski and Williamson (1997): "During the first few time steps, differences between the original and ported code solutions should be within one to two orders of magnitude of machine rounding". If this is correct, it would help to clarify as the primary reason for failure.*

Response: The respective sentences in the discussion paper were: "When the test was originally developed, the physical parameterizations were quite simple, and the

test was robust. The method gradually became less useful as the model became more comprehensive and complex, and compromises were made to preserve some utility for the test.For example, in CAM4, the PerGro test needed to be performed in an aqua-planet configuration, i.e., without the land surface parameterizations, and with a few (small) pieces of code in the atmospheric physics parameterizations switched off or revised, because those codes were known to be very sensitive to small perturbations, and would always lead the test to fail."

We provide the following clarification in the revised manuscript: Rosinski and Williamson (1997) established two conditions for the validation of a ported code:

- Condition 1. During the first few time steps, differences between the original and ported code solutions should be within one to two orders of magnitude of machine rounding.
- Condition 2. During the first few days, growth of the difference between the original and ported code solutions should not exceed the growth of an initial perturbation introduced into the lowest-order bits of the original code solution.

It is important to note that in order for those two conditions to be useful for the intended verification, the model code has to satisfy a "Condition 0":

- Condition 0. During the first few time steps, rounding-level initial perturbations introduced to the original code in the original environment should not trigger solution differences larger than one to two orders of magnitude of machine rounding.

If Condition 0 is violated, it is expected that the ported code will always fail Condition 1 whether there is a porting error or not; in addition, the very rapid growth of perturbations even in a trusted computing environment could make it difficult to distinguish differences between trusted solutions from differences between a trusted solution and

a problematic test solution, causing misleading fulfillment of condition 2. Therefore, if Condition 0 is violated, Conditions 1 and 2 might no longer be useful for porting verification.

When the PerGro test was originally developed, the physical parameterizations were quite simple, the code was able to satisfy Condition 0, and the test method was robust. As the model became more comprehensive and complex, more rapid growth of rounding-level initial perturbation was observed. Compromises were made to preserve some utility for the PerGro test. For example, in CAM4, the test needed to be performed in an aqua-planet configuration, i.e., without the land surface parameterizations, and with a few (small) pieces of code in the atmospheric physics parameterizations switched off or revised, because those codes were known to be very sensitive to small perturbations. If those pieces of codes were not switched off or revised, perturbations on the trusted machine would grow so rapidly that the RMS differences grew to $\mathcal{O}(0.1)$ over a few timesteps. Disabling the land interactions and a few pieces of code returned the bulk of the atmospheric model to a configuration where differences between perturbed and unperturbed initial conditions grew substantially more slowly. Most of the time, the RMS differences grew at a rate well below one order of magnitude per timestep in a trusted environment. An example is shown by the blue curve in Fig. 1 of the discussion paper (see also Fig. 1a in this document). With the revised aqua-planet configuration of CAM4, it was still possible to examine solution differences between original and test solutions to see whether they violated Condition 2 for a port validation effort. But with CAM5, initial perturbations grow too rapidly even in an aqua-planet simulation (see red curve in Fig. 1a below), making the original PerGro method no longer useful for porting test.

Comment: *Page 2, #55: It is stated that "Recent versions of the model have become so complicated that rounding level differences in the initial condition can result in very rapid divergence of the simulations". It is not obvious, and no evidence is presented, that code "complication" is a reason for the faster growth. Is it possible, for example,*

that the initial condition has points which lie on a code branch ("if" test)? Or more generally, perhaps the new physics is driving some quantity such as temperature toward a value which lies on a branch, such as the freezing point of water? If implemented via a tendency equation, the computed value may be one mantissa bit greater than, or one mantissa bit less than, the actual freezing point of water. If a subsequent "if" test applies substantially different algorithms across "true" and "false" branches of a test versus the freezing point, this can be a reason for rapid growth not necessarily related to code complication. This exact scenario was encountered many years ago when testing growth behavior with the relatively simple BATS land model in CAM.

Page 3, #65: It is stated that "The very fast evolution of initial perturbation is caused by multiple factors". What are those factors? Similar to the previous point, a weakness of the paper is that it does not describe any of the reasons for rapid growth. There is only speculation that code complication is to blame.

Response: So far we have found three major contributors to the rapid divergence of solutions in the current model:

First, the default time step of 1800 s in CAM5 is significant compared to the characteristic time scales of many physical processes represented by the model, so the increments in the model state (the process tendencies times the model time step) are significant, and the differences between a pair of simulations with slightly different initial conditions can also be perceptible. The red and purple curves in Fig. 1b below show that when the time step sizes of all model components are changed by a factor of 1800, the solution differences after the same number of time steps also change by a similar ratio.

Second, the solar and terrestrial radiation parameterization in CAM5 uses a pseudo random number generator, and the seeds for the generator are chosen from the least significant digits of the pressure field. This effectively introduces state-dependent noise into the numerical solution. The green curve in Fig. 1b below shows the differences

between a pair of simulations conducted with 1 s time step but with radiation calculated only once at the beginning of the integration. Compared to the purple curve where radiation was calculated every other time step, the solution differences were further reduced by about 3 orders of magnitude. We note that the noisiness from the radiation calculation can be controlled by making the random seeds independent of the model state so that the random series become reproducible from one simulation to another. But the radiation example also implies that models with state-dependent stochastic parameterizations might feature rapid perturbation growth as well.

The third reason for rapid perturbation growth has to do with poorly conditioned pieces of code. Two types of examples where discussed by Rosinski and Williamson (1997): (i) an upshift in digit of solution error resulting from division by a small number, and (ii) if-statements associated with algorithmic discontinuity. We have experienced both types of situations in the CAM5 code, although the specific formulae were different from those given in the paper of Rosinski and Williamson (1997). Compared to its predecessors, CAM5 uses modern parameterizations with substantially more detailed description of the atmospheric phenomena, and the model also carries an expanded list of tracers. The increase in model complexity and the corresponding growth in the size of the code substantially increase the chance for poor conditioning to occur.

The explanations above are included in the revised manuscript. We think a more detailed description of our findings is out of the scope of the present manuscript. A separate paper is in preparation:

Singh B., Rasch, P. J., Wan, H., and Edwards, J.: A verification strategy for atmospheric model codes using initial condition perturbations. To be submitted.

Comment: *Page 5, #125: Generally commutative operations are not answer-changing. Instead perhaps the authors mean "associative operations"?*

Response: Thanks for pointing out this error. We indeed meant “associative”. This is corrected in the revised manuscript.

Comment: *Page 6, #170: How is the convergence rate of 0.4 calculated?*

Response: The convergence rate is the regression coefficient of the linear regression between ensemble mean $\log_{10}$ RMSD and $\log_{10} \Delta t$. This is clarified in the revised manuscript.

Comment: *Page 9, #285: Definition of the two separate domains is presumably land and ocean. It would help readability to state this up front, and also the reasons for the choice.*

Response: We clarify the following in the revised manuscript: The essence of our new test method is to distinguish solution differences caused by code modifications or computing environment changes from solution differences caused by model time step change (2 s versus 1 s). While certain changes in the model code, e.g., those related to dust emission or convection over land, only affect a limited number of grid points during simulations that are just a few minutes to a hour in length, time step size affects the solution from the first step and at all grid points. Consequently, subtle but “real" solution changes might be masked by the model’s time stepping error thus difficult to detect. To help address this challenge, we calculate the test diagnostics for N_{dom} = 2 domains, i.e., land and ocean. This is a practical and somewhat arbitrary choice that aims at increasing the sensitivity of the TSC test.

Comment: *Page 15, #495: If passing the test doesn’t guarantee that the model will produce the same climate characteristics, isn’t this a weakness of the procedure? I thought the main point of the procedure was to provide a mechanism to enable non-experts to confidently commit roundoff-level code changes to the repository.*

Response: Strictly speaking, the TSC test is a method for assessing whether solution differences seen in very short (few-minute) simulations exceed the numerical accuracy of the model’s time stepping algorithms. This neither assesses whether the solution differences are at rounding level, nor determines whether the climate characteristics are the same. We note that when PerGro was considered a useful porting validation

method, passing that test did not guarantee the model would produce the same climate, either. Given the invalidity of the PerGro method in CAM5, and the high computational costs associated with conducting and evaluating climate simulations, the TSC method provides a practical and useful alternative to determine whether the model is behaving as expected in the sense that the numerical solutions feature the same time stepping error when compared to a predefined set of reference solutions. This is clarified in the revised manuscript.

Comment: *The "major revisions" requested involve a much more thorough analysis of the reasons for rapid perturbation growth in CAM4 and CAM5. Speculation about "code complexity" is not adequate. The example cited by this reviewer of rapid growth caused by a simple land scheme (BATS) was really a bug not a feature of the scheme. It would be nice to have some assurance that this possibility (ill-formed or buggy algorithms) has been explored to some extent with the current CAM model.*

Response: We agree with the referee that the reasons for rapid perturbation growth in CAM is an important (and also very interesting) research topic. As mentioned above, we have managed to understand at least some of the causes, and included brief explanations in the revised manuscript. To us, the rapid perturbation growth is a motivation for developing a new test method but not the focus of this manuscript. We will report in detail our findings regarding perturbation growth in a separate paper.

Interactive comment on Geosci. Model Dev. Discuss., doi:10.5194/gmd-2016-142, 2016.

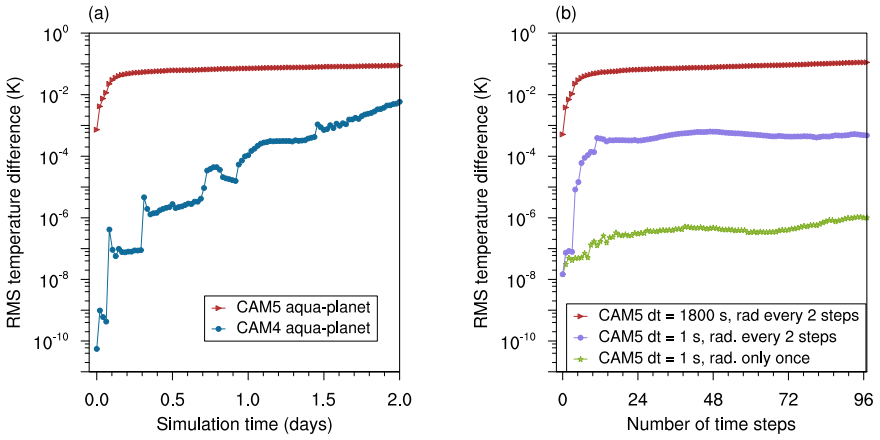

Figure 1: Examples of the evolution of RMS temperature difference (unit: K) caused by random perturbations of order 10^{-14} K imposed on the temperature initial conditions. (a) Aqua-planet simulations conducted with the CAM4 (blue) and CAM5.3 (red) physics parameterization suites using the default 1800 s time step. (b) Simulations conducted with the CAM5.3 physics suite using the default 1800 s time step and with radiation calculated every other step (red), using 1 s time step and with radiation calculated every other step (purple), and using 1 s time step and with radiation calculated only once at the beginning of the integration (green). All simulations used the spectral element dynamical core at approximately 1° horizontal resolution.

Fig. 1. Examples of the evolution of RMS temperature difference caused by initial perturbation.