Dear Editor,

We hereby submit a revised version of the manuscript gmd-2016-142 entitled "A new and inexpensive non-bit-for-bit solution reproducibility test based on time step convergence (TSC1.0)" for consideration of publication in GMD. We appreciate the careful and insightful reviews from the anonymous referees and from Dr. W. Sacks. In response to the comments and suggestions, we have made the following changes in the manuscript:

1. The purpose of the proposed testing method is clarified. We have realized that it is more accurate to state that the TSC test is designed for regression testing, i.e., for verifying that results from a model stay the same despite changes in the code or in the computing environment (Sect. 1). The TSC method considers the outcome of a simulation unchanged if the numerical solution is found to have the same time stepping error relative to a reference solution obtained with a previously verified code and computing environment (Sect. 2.1). Our understanding of the linkages and distinctions between TSC and other testing methods is explained in Sect. 5.3.

2. The scope of the proposed method is clarified. We point out in the Abstract and explain in Sect. 2.1 that the TSC method is designed for identifying numerically significant changes in solutions of evolution equations. It does not detect issues associated with diagnostic calculations that do not feedback to the model state variables.

3. More information is provided in Sect. 2 and Sect. 3 on the reasoning behind the specific choices we made for the version 1.0 implementation, for example the list of monitored variables, the splitting of model domain into land and ocean, the pass/fail criterion, and the initialization strategy. We also clarify that many of those choices are practical and empirical, and can be further evaluated and improved in the future (Sect. 3.2 and Sect. 5.3).

4. The overall pass/fail criterion is revised (Sect. 3.2). The use of multiple time steps instead of a single time instance reflects our perspective of viewing the model integration as a time evolution problem. We also point out in the manuscript that the revised pass/fail criterion is still empirical and preliminary, and can be further evaluated in the future (Sect. 3.2 and Sect. 5.1).

5. Two test cases with code modifications following Milroy et al. (2016) are added. Three cases with perturbed parameters and two cases with change of computing environment are removed. The purpose is to focus the discussion of the result on comparison with CAM-ECT.

6. All simulations presented in the discussion paper have been repeated with the radiation parameterization calculated every other time step instead of only at the beginning of the simulations. We found this change to have only very small impact

on the outcome of the TSC test. Nevertheless, when evaluating the TSC methodology using different test cases (Sect. 4), we present the new results so that the time step ratios between different model components remain the same as in the default model despite the change in time step sizes. In Sect. 2.3 where the concept of time step convergence is introduced, we present the old results for consistency with the earlier work of Wan et al. (2015), but add a note that the calling frequency of radiation does not change the convergence property of the full CAM5 model.

7. A brief discussion (Sect. 5.2) is added on the impact of noisy parameterization on the results of the TSC test.

8. Reasons for the rapid growth of initial perturbation in the CAM5 model are summarized in Sect. 1.

9. Typographical and grammatical errors are corrected at miscellaneous places.

Our detailed responses to the reviewers' comments and the corresponding changes in the manuscript are attached in the next pages.

Sincerely,

Hui Wan

Reply to Dr. W. Sacks

We thank Dr. Sacks for his insightful questions. Our responses are detailed below.

Comment: *This is a clever idea, and the paper is very well written. I'd like to be convinced that this technique truly has more power than seemingly simpler techniques. For example, can some of the same experiments be redone with this set of runs?:*
(1) control: unmodified model with 1s time step
(2) baseline for comparison: unmodified model with 1s time step, with a roundoff-level perturbation in the temperature field
(3) test code: some change in the code with 1s time step
Basically, I'd like to be convinced that the "time step convergence" is truly needed here, and that it truly provides more power than just comparing two versions of the model with a short time step. Does the above, conceptually simpler test give false positives or false negatives in cases where the TSC test gives the correct answer?

Response: This is an excellent question that touches upon some aspects of the old and new test methods that we did not elaborate on in the discussion paper. Essentially, Dr. Sacks asked whether the old PerGro test would become useful again if the model time step was set to 1 s instead of 1800 s. Our answer is "yes, but that revised test could still give false negatives in some circumstances where the TSC method gives the correct answer".

The original PerGro test is no longer useful for the default CAM5 model because even in a trusted computing environment, initial perturbations of $\mathcal{O}(10^{-14})$ K grow so rapidly that the resulting solution differences are often undistinguishable from solution differences caused by unintended code changes or incorrect porting. In our response to referee #2's comments and in the introduction section of the revised manuscript, three reasons are listed as reasons for the rapid growth: (a) long time step, (b) state-dependent randomness in the radiation code, and (c) particular code pieces. Reducing model time step addresses issue (a) (see Fig. 2b below in our response to referee #2 and Fig. 1b in the revised manuscript), thus helps to alleviate the perturbation growth; but problems (b) and (c) still exist, and lead to divergence of trusted solutions that can masks subtle but systematic solution changes. Below is an example.

We conducted PerGro test runs using 1 s time step and with radiation called every other time step (so that the time step ratio between radiation and the other parameterizations stay the same as in the default model). We then conducted simulations with the dust emission parameter changed from 0.55 to 0.45 as in the DUST case presented in the discussion paper, also with 1 s time step and with radiation called every other time step. The exercise was repeated using 11 additional sets of initial conditions. As can be seen in Fig. 1 below, the temperature RMS differences induced by the parameter change (solid orange lines) stayed substantially below the reference curves (dashed black lines) in the first $\sim$10 time steps, then quickly approached the reference curves but did not

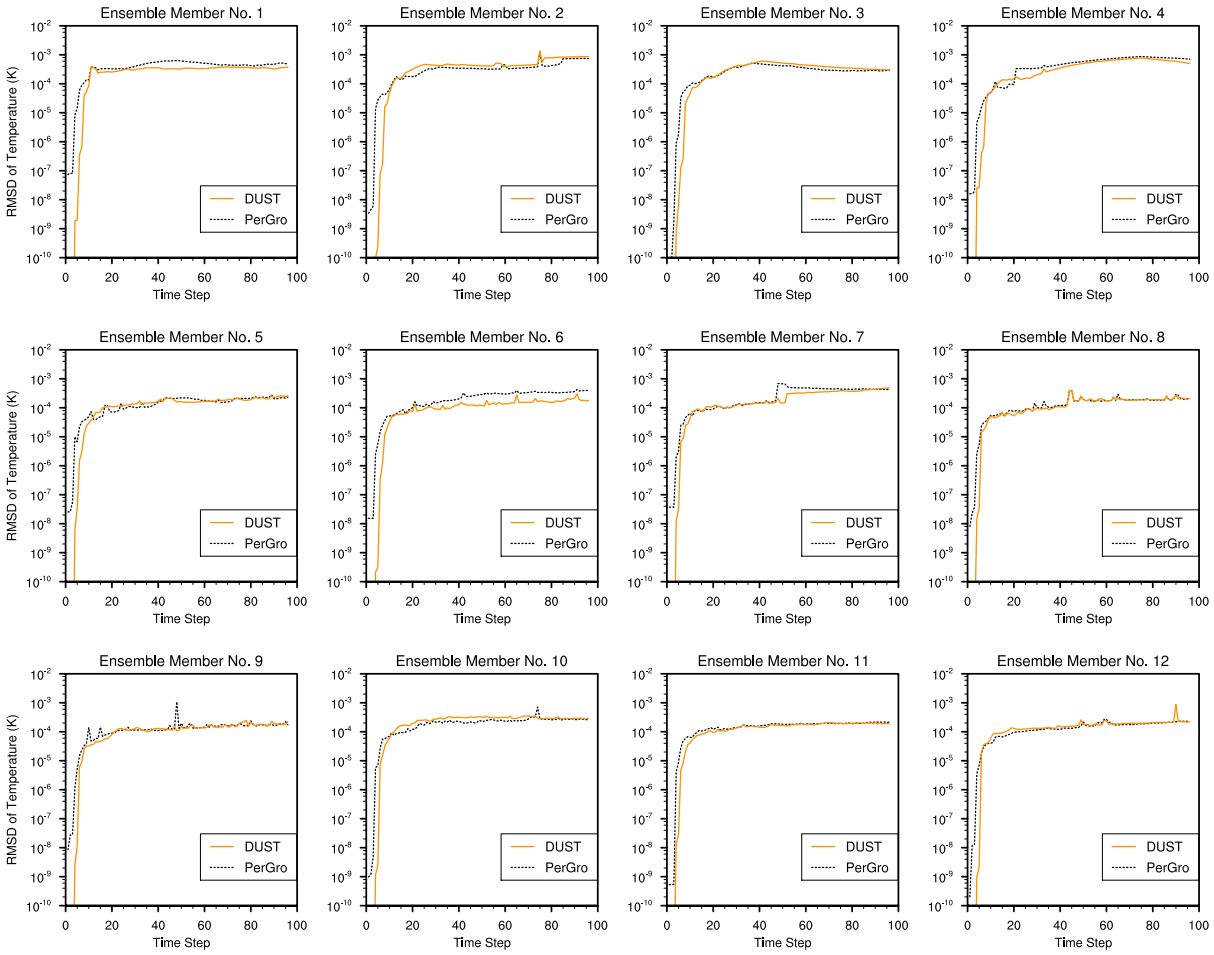

Figure 1: Comparison between the temperature RMS differences caused by initial perturbation of $\mathcal{O}(10^{-14})$ K (dashed black lines, "PerGro") and the differences induced by changing the dust emission parameter from 0.55 to 0.45 (solid colored lines). All physics parameterizations used 1 s time step except for radiation which was calculated every other step. The simulations were conducted on Titan at the Oak Ridge Leadership Computing Facility using the default compiler setups. The 12 ensemble members used initial conditions sampled from different months of a previously conducted multi-year climate simulation with the default CAM5.3 model and the FC5 component set.

exceed them in any of the ensemble members. We extended the simulations to 300 steps and the results remained the same. Based on the description of the PerGro test at http://www.cesm.ucar.edu/models/cesm1.0/cam/docs/port/pergro-test.html, one would consider the DUST case as a clear "pass", while both our TSC method and the CAM-ECT assigned the case a "fail".

It is worth noting that the PerGro method perturbs and monitors only the temperature field. Since the impact of dust emission is limited to a rather small number of grid points in very short simulations, it is not surprising that the emission change cannot be detected by PerGro even with 1 s time step. The TSC method makes use of the fact that a change in model time step directly affects all prognostic equations. We monitor multiple state variables, and also calculate the test diagnostics for land and ocean separately, thus achieved higher sensitivity with the TSC method.

In the revised manuscript, a brief summary is added to the introduction section on the reasons for rapid perturbation growth in the CAM5 model. The motivation for monitoring multiple prognostic variables and model subdomains in the TSC test is explained in Sect. 2.1 ("Purpose and scope") and Sect. 3.2 ("Test procedure"), and further discussed in Sect. 4.2 ("Results at 5 min" under "Numerical experiments") and Sect. 5.1 ("Test setup" under "Discussion").

Comment: *I'd also like clarification on the following point: On a continuum from non-answer-changing to answer-changing, I see mention of the following types of changes: (1) bit-for-bit identical, (2) answer-changing only at the round-off level, (3) answer-changing only within the limits of numerical accuracy due to the discrete time step size, and (4) climate changing, according to criteria like SIEVE or CAM-ECT. The TSC test distinguishes changes at level 3 or lower from those at level 4. But is there actually a level in between (3) and (4): changes that affect the model evolution in an appreciable way, but are not large enough to cause statistically detectable changes in climate? It seems that many bugs might fall into this intermediate regime e.g., accidentally flipping the sign on a minor term in an equation. Do the authors feel that there is a set of changes that falls between (3) and (4), and if so, how do they expect these changes to be categorized by the TSC test?*

Response: This additional level between (3) and (4) might exist in principle, in which case the TSC test would assign a "fail" to the results and would not be able to distinguish them from level-(4) differences.

We also would like to point out that level-(3) and level-(4) changes are not strictly defined in a quantitative sense. For example, two simulations representing indistinguishable climate according to SIEVE based on the AMWG diagnostics package might be distinguishable using additional metrics or using CAM-ECT. Similarly, two simulations determined to be consistent using CAM-ECT based on the global and annual averages might turn out distinguishable using grid-point-wise model output and monthly time series. As for level (3), the relatively strong time step sensitivity in CAM5 implies that

the numerical accuracies are substantially different when time step is changed, so level (3) is not a fixed criterion either. As can be seen in Fig. 2 of the discussion paper, if we had chosen to conduct a TSC test using a 1800 s time step instead of 2 s, the results from the RH-MIN-HIGH case (which was determined by CAM-ECT as climate-changing) would have been assigned a "pass" by TSC. While answering the "climate-changing or non-climate-changing" question using a specific set of metrics provides *one* assessment of the solution similarity/difference, the TSC method provides a different assessment of the magnitude of solution changes. From a theoretical point of view, the relationship between those two kinds of tests is not entirely clear; practically, because there are flexibilities in the design of the TSC test (e.g., time step size and pass/fail criterion), it should be possible to set up the test so that the outcome closely matches the results from a predefined climate reproducibility test. Evidence is provided in the current manuscript, and future work is planned to further evaluate the strengths and limitations of the TSC method.

In the revised manuscript, we have added a new section (Sect. 5.3) to discuss the linkage and distinction between TSC and other testing methods.

Reply to Referee #1

We thank the referee for the careful review. Our responses are detailed below.

Comment: *General comments:*
Overall the paper was well-written and clear. The TSC test idea is a clever application of the time step convergence work from Wan et al. (JAMES, 2015) and appears useful. Certainly this approach is promising and inexpensive, and the manuscript is a good start. More details on the manuscript are provided below, but my main concerns to address are as follows:
(1) The paper would have been stronger if the test parameters had been fleshed out more thoroughly, particularly the ensemble size, the false positive rate, and number of variables. For example, because this test returns a "fail" if a single variable fails, then a larger subset of variables will increase the possibility of failure by chance, so making the reader aware of this relationship would be useful.

Response: The intention of this manuscript is to describe a first implementation of the TSC test procedure in the CAM5 model and to provide initial evidence that it is a practical and useful method for model testing. We revised the wording in the abstract and in the "Conclusions" section to clarify this. A new section ("5. Discussion") is added to the manuscript to point out that the test setup can be hardened, and that future work is planned to further evaluate the specific choices (e.g., ensemble size and the pass/fail criterion), and to evaluate the strengths and limitations of TSC by comparing it with other methods.

In Sect. 3.2 ("Test procedure"), we have added the comment that the typical values of $\mathcal{P}_{\min,t}$ depends on the number of monitored variables (i.e., larger $N_{\text{var}} \times N_{\text{dom}}$ can result in smaller $\mathcal{P}_{\min,t}$ in a statistical sense), hence $\mathcal{P}_0$ (the threshold $\mathcal{P}_{\min}$ for failing the test) needs to be determined empirically for a given $N_{\text{var}} \times N_{\text{dom}}$. Ideally $\mathcal{P}_0$ should be small enough to reduce the chance of false positive (i.e., insignificant solution differences being assigned a "fail"), and large enough to reduce the chance of false negative (i.e., subtle but systematic solution differences being assigned a "pass"). In the present paper we have made an empirical choice. Further evaluation of this choice and possible improvement of the overall pass/fail criterion are topics of future work.

Comment: *(2) More details on the scope of the test would be helpful. There are bits in section 2.1 and later in 4, but it would help to better quantify the scope beyond the "equation-solving" part. In particular, the selection of variables would seem to impact the scope. Because of the limited (?) scope, an example of a bug/issue that is not caught would be helpful. And ideally this counter-example would be discussed within a larger discussion of scope as relating to the choice of variables.*

Response: We have rewritten Sect. 2.1 ("Purpose and scope") to clarify that TSC was designed from the point of view that CAM is a general circulation model that solves a large set of differential, integral, and algebraic equations. The model variables (i.e.,

arrays in the code) can be categorized into the following types:

I. Prognostic and diagnostic variables whose equations are coupled to one another, so that any change in variable A will, within one time step or after multiple time steps, affect variable B in the same category. Examples in this category include basic model state variables like temperature, winds, and humidity, as well as quantities calculated as intermediate products in a parameterization, for instance the aerosol water content (which affects radiation and eventually temperature), and the convective available potential energy (which affects the strength of convection hence temperature and humidity).

II. Prognostic variables that are influenced by type-I variables but do not feedback to type I. An example could be passive tracers carried by the model to investigate atmospheric transport characteristics (e.g., Kristiansen et al., 2016).

III. Diagnostic quantities that are calculated to facilitate the evaluation of a simulation, but do not feedback to type I. Examples include the daily maximum 2-m temperature, the ice-to-liquid conversion rate in the cloud microphysics parameterization (which is a quantity calculated merely for output in CAM5.3), and any variable specific to the COSP simulator package (Bodas-Salcedo et al., 2011).

Code pieces in the model can be categorized accordingly.

Our standpoint is that the essential characteristics of the simulated climate are determined and represented by type-I variables, and the TSC test is designed for code pieces in this category. Since all variables in this type are coupled, and since our test method monitors instantaneous and grid-point-wise values before chaos sets in, any significant bug or compiler error (that affects the solution of the *coupled* equation set) should be detectable through the monitoring of a single variable, as long as there is sufficient integration time for the impact to evolve to a discernable magnitude and propagate to that variable. When the simulations are short (for instance on the order of minutes of model time as in TSC), tracking multiple variables can help increase the sensitivity of the test (decrease the chance of false negative) since discernable solution differences might show up earlier in some variables than in others.

The list of variables monitored by TSC can be extended to type-II variables defined above, if the user wishes to cover the related code pieces in the testing. Diagnostic variables of type I or type III should not be included in the list because the concept of time step convergence does not apply. Consequently, bugs in the implementation of "diagnostic-only" calculations, e.g., a satellite simulator, would not be detected by TSC. Also, issues with code pieces that are not exercised, for instance the restart capability, would not be caught by the test either. In Sect. 2.1 and in the revised abstract, we acknowledge that our test method is not exhaustive in the sense that it does not provide a full coverage of all code pieces in the model.

Comment: *(3) The experimental results section would be stronger if the experiments more closely represented the stated scope (see previous comment). Then the reader would gain a better understanding of the tools utility. The chosen experiments are essentially the same as those in Baker, et al. (GMD, 2015). While it is important to include those, it is not clear that the 10 variables chosen would be sufficient to catch errors in all parts of the code (as stated in section 2.1), so it would be helpful to have an example of an error that is not caught. Also, several times (e.g., section 2.1) "code modifications" are mentioned as an application for this test, but there is not an example supporting this statement (and including such an example seems important).*

Response: Please see our response to the previous comment for a clarification on the scope of our test, and for examples of bugs/issues that would not be caught by TSC.

As for "code modifications", two test cases from Milroy et al. (2016) that represent code optimization strategies are included in the revised manuscript: "division-to-multiplication" (DM) and "precision" (P).

Comment: *(4) Regarding the TSC's use of the t-test, please clarify the reason for the directional t-test. In particular, why does the test only check if the mean is larger than zero? (i.e., mu_j > 0) - as opposed to the non-directed alternative hypothesis: mu_j / = 0. Certainly mu_j can be negative, so is this scenario just not of concern? For example, in figure 3, if delta_RMSD for variable was negative for all 12 members, then TSC would issue a pass. I am not necessarily questioning the efficacy of the test procedure, but I have to wonder if systematically negative results can be problematic as well or even indicative of an issue with the simulation being tested.*

Response: The test metric of the TSC method is the model's time stepping error in simulations conducted with 2 s time step compared to trusted reference solutions conducted with 1 s time step. If both the model equations and the discretization methods stay the same, the time stepping error is expected to stay the same. If bugs are introduced, or if the code is not compiled or executed correctly, the resulting numerical integration will not be solving the originally intended equations, thus not converging to the original reference solutions, resulting in larger apparent time stepping errors. This is now explained also in Sect. 3.2 ("Test procedure") when describing step 3 of the TSC test.

In a non-answer-changing case, while ΔRMSD can be negative by chance for an ensemble member, it is very unlikely that it will be negative for all members. The only situation we could imagine systematically negative ΔRMSD to occur would be the implementation of a new and more accurate set of time stepping algorithms that featured smaller sensitivity to the step size change of 1 s to 2 s, but yet produced very similar solutions at 1 s time step when compared to the original code. Such a case of algorithm update would be considered a substantial code change, so methods like TSC and PerGro would not be the most natural tests to perform since they are designed to assure that the solutions are unchanged. Once the merits of new algorithms have been confirmed and a

new default model is established, a new set of reference solutions (with 1 s time step) and trusted solutions (with 2 s time step) should be generated and used for future testing.

Comment: *Specific comments:*
(1) Section 1: line 20: Check the use of "reproducibility" in this context.
(2) Section 1: The first couple paragraphs are quite similar in parts to the text in Baker, et al. (GMD, 2015), including the same references and some of the same phrases, which is a bit awkward.

Response: The first two paragraphs of the manuscript have been rewritten.

Comment: *(3) page 3, line 24: The tool's application to "code modifications" is mentioned here and in section 5, but I dont believe this is being tested in the experiments. It may be of interest to look at CESM code modification experiments in the followup to Baker, et al. (GMD, 2015), which is:*
Daniel J. Milroy, Allison H. Baker, Dorit M. Hammerling, John M. Dennis, Sheri A. Mickelson, and Elizabeth R. Jessup, Towards characterizing the variability of statistically consistent Community Earth System Model simulations. Procedia Computer Science (ICCS 2016), Vol. 80, 2016, pp. 1589-1600.
(http://www.sciencedirect.com/science/article/pii/S1877050916309759)

Response: Thanks for the reference. Two test cases of code modification from Milroy et al. (2016) that represent code optimization strategies are included in the revised manuscript: "division-to-multiplication" (DM) and "precision" (P).

Comment: *(4) Section 2.1: I would really like to better understand how the selection of the 10 variables affects (or does not affect) the scope.*

Response: Please see our response to general comment #(2) for a categorization of the model variables. The TSC method described in the manuscript is designed to test all code pieces that affect type-I variables, and the 10 variables we chose all belong to that type. Monitoring more (fewer) variables of the same type would not affect the scope of the test but could affect the test's sensitivity for a chosen integration length, i.e., it could decrease (increase) the chance of false negative, since bugs or issues associated with a specific piece of code might take longer time to cause discernable solution differences in one variable than in another. Adding type-II variables, on the other hand, would extend the scope of the TSC test.

Sect. 2.1 ("Purpose and scope") of the manuscript has been rewritten. We also point out in the abstract and in Sect. 3.2 ("Test procedure") that the TSC test targets at the evolution equations in a model, and does not provide a full coverage of the entire code.

Comment: *(5) page 2, line 25: This is not exactly true as CAM-ECT has been used to pinpoint errors in specific code modules (e.g. FMA error on Mira detailed in Milroy et al. 2016).*

Response: The respective sentences in the discussion paper read:

"The CAM-ECT of Baker et al. (2015) fulfills criteria 1–4 and 7... Moreover, since each ensemble member is a one-year simulation, it is unlikely that the method can be used to test a small subset of the model components, or a code that is still in debugging stage thus numerically unstable for long simulations (criterion 5)."

The statements are revised as follows:

"The CAM-ECT of Baker et al. (2015) fulfills criteria 1–4 and 7, and partly 5. For criterion 5, we expect CAM-ECT to be capable of isolating issues associated with variables of type II or III (cf. Sect. 2.1) through systematic elimination of model output variables from the test diagnostics (Milroy et al., 2016). Bugs associated with type-I variables would be more difficult to pinpoint: since all variables in this type are inherently coupled, we expect that any substantial change in one equation would have affected all the type-I variables after a year of model integration. One-year simulations might also be challenging for a code that is still in debugging stage thus numerically unstable for long simulations."

The FMA error on Mira as described in Milroy et al. (2016) is an interesting case worth further investigation. To keep the manuscript focused, we do not include any detailed discussions on that topic, but some of our thoughts are included here:

In the Milroy et al. (2016) paper, it was reported that six output variables from the CAM model were identified as suspects for further inspection. We contacted the authors and obtained the actual list of those variables. Five out of those were in fact type-III ("diagnostic-only") variables as we suspected, but it was curious that the sixth variable was CLDLIQ, the mass concentration of liquid-phase condensate in stratiform clouds. Given the important role of this prognostic variable in the model, it is counterintuitive to us that values of this variable obtained on Mira were inconsistent with the control ensemble while values of other closely related variables like temperature, humidity, and cloud properties were consistent. Could it be that the inconsistency in CLDLIQ was very minor thus the impacts on other variables were negligible? Would we see more substantial inconsistencies and in more variables if spatial patterns were included in CAME-ECT? The answers to these questions are unknown at this point. We also learned from Mr. Milroy and Dr. Baker that a number of code lines and local variables in the cloud microphysics parameterization were identified as being affected by FMA. It was again counterintuitive to us that those local variables included the microphysical tendencies of cloud droplet and ice crystal number concentrations, but the corresponding state variables were deemed consistent between the Mira results and those from the trusted computers. To us, this again indicates that the case is worth further investigation in the future.

Comment: *(6) page 3, lines 27-28: Regarding "...when the accuracy limits related to the algorithmic implementation are taken into account." This doesnt appear to be considered in the rest of the paper.*

Response: The subsection on test scope has been rewritten. What we meant by the sentence cited above has been rephrased: From the point of view that CAM is a general circulation model that solves a large set of differential, integral, and algebraic equations,

we consider the outcome of a simulation unchanged if the numerical solution is found to have the same time stepping error relative to a reference solution obtained with a previously verified code and computing environment.

Comment: *(7) page 4, line 14: I agree with #5 as a desirable feature, but I don't believe that evidence was given in this manuscript that TSC fulfills #5. Certainly no evidence was given in Baker, et al. (GMD, 2015) that CAM-ECT satisfies #5, though one can imagine the framework could possibly apply. So if the claim is that TSC fulfills this while CAM-ECT does not, it would be stronger to provide specific evidence of such a case for TSC (i.e., an experiment to validate the claim).*

Response: In the earlier study of Wan et al. (JAMES, 2015), in addition to assessing time step convergence in the the full CAM5 model, convergence analysis was also done for configurations that exercised the dynamical core plus only one parameterization or parameterizations group at a time, e.g., deep convection, shallow convection, large-scale condensation, or the stratiform cloud microphysics. This was an attempt to find out which of those parameterizations led to the convergence rate of 0.4 (instead of 1) in the full model. Simulations were also conducted using the dynamical core plus a very simple saturation adjustment scheme, or with the cloud microphysics parameterization of CAM5 but with the formation and sedimentation of rain and snow turned off (see Figure 3 in Wan et al., 2015, JAMES). Those simulations conducted with a small portion of the CAM5 code were likely to blow up if the integration had proceeded longer than a few hours or days, and certainly would not produce any realistic climate, but they clearly revealed different convergence rates and time step sensitivities associated with different components of the model code. We imagine the same strategy of breaking down the code into small exercisable units and evaluating convergence could be used to pinpoint bugs when, e.g., a code refactoring leads to unexpected failing results from the TSC test. This is why we believe the TSC method fulfills feature #5. A paragraph is added to the end of Sect. 2.3 ("Time step convergence (TSC)") for clarification.

Comment: *(8) Section 2.3: Since the starting conditions for the TSC ensemble are samples from "a previously conducted long-term simulation", does one need to update this simulation with answer-changing CESM tags, for example? Also does "long-term" mean 1-year or ????? Please give more details on how this part of the process works.*

Response: We mention in Sect. 2.3 of the revised manuscript that experience so far indicates the diagnosed convergence rate is rather insensitive to the choice of initial conditions. We also clarify in Section 3.2 ("Test procedure") that the initial conditions used in this manuscript were sampled from the first year (after 6 months of spin-up) of a previously conducted 5-year simulation. The decision of using the first year was arbitrary. In our experience, climate simulations of 1–5 years are frequently carried out during model development or evaluation, making such initial conditions easy to obtain. The two features we had in mind when choosing the initial conditions were that they (i) contain reasonably spun-up values for the model state variables (e.g., not all zeros or spatially

constant values for the hydrometeors or aerosol concentrations), and (ii) represent synoptic weather patterns in different seasons. Those initial conditions do *not* need to represent well-balanced states in the quasi-equilibrium phase of a multi-year climate simulation. In fact, the default model time step of 1800 s was used when creating the initial conditions for this study, while the control and test simulations in TSC used 1 s or 2 s time step, so the model state was certainly not well-balanced during those TSC simulations. Also notice that while model states from different seasons were used, all ensemble members were initialized on January 1, 00Z for simplicity of the simulation and postprocessing workflow, which also led to initial imbalances. Such imbalances are considered harmless since the purpose of the numerical integration is regression testing rather than faithfully simulating the atmospheric motions in the real world. We expect that the same set of initial conditions can be used after answer-changing code tags are established – until a point when the list of prognostic variables in the model becomes substantially different. Then it would be useful to regenerate the initial conditions, and rethink which variables should be included in the test diagnostics.

Comment: *(9) Would the TSC test results be affected if the ensemble was created instead by perturbing initial conditions (since this does not require a previous simulation)?*

Response: We have not tried this idea yet, but suspect that the answer would depend on the magnitude of the initial perturbations. Since our intended simulation length is on the order of minutes to an hour, small perturbations like those used in PerGro and CAM-ECT might not have time to trigger sufficient spread (variability) among the ensemble members. The need for ensemble is demonstrated by Figure 3 in the manuscript.

Comment: *(10) page 5, last paragraph: Should point out that this test (RH-MIN-HIGH from .8 to .9) is from Baker, et al. (GMD, 2015) for comparison.*

Response: Done.

Comment: *(11) page 5, line 34-page 6, line 1: "[...] concept of self-convergence since no structural changes [...] have been introduced into the model." More generally (and relevant to the discussion in Sect 3.2), what if the modified model's 2s timestep behavior is closer to the 1s timestep reference model than to itself for 1s timestep? In other words, what if its convergence behavior to the reference model is different than its self-convergence?*

Response: Given the complexity of the model and its time stepping algorithms, we would argue it is very unlikely that a modified model's behavior at 2 s time step will be closer to the reference solution at 1 s of an old model than to the reference solution at 1 s time step of the new model. As mentioned earlier in our response to general comment #(4), the only situation we could imagine to see that kind of results would be the implementation of a new and more accurate set of time stepping algorithms that featured smaller sensitivity to the step size change of 1 s to 2 s, but yet produced very similar solutions at 1 s time step when compared to the original code. Such a case of algorithm update would be considered a substantial code change, so methods like TSC

and PerGro would not be the most natural tests to perform since they are designed to assure that the solutions are unchanged.

Comment: *(12) page 6, line 13: The "more substantially" comment is a bit vague. The change is already labeled "climate-changing", which itself seems substantial. Certainly this change is more substantial than, for example, changing the order of operations in the code or something similarly "minor". Clarify?*

Response: Two simulations that are both "climate-changing" can differ from the control simulation by different magnitudes. We revised the wording of the respective sentences as follows:

"If we had introduced larger changes in the model, e.g., by changing cldfrc_rhminh to 0.999 instead of 0.9 from the default value of 0.8, or by replacing a certain parameterization by a different scheme, the impact might be more visible at the default step size. In contrast, if the parameter change were smaller, e.g., from 0.8 to 0.82 instead of 0.9, the red and blue convergence pathways in Fig. 2 might not diverge until a step size on the order of a few seconds."

Comment: *(13) page 7, first paragraph: Did the authors use the SIEVE method to verify all of the results presented? It is not clear. Also wondering if the example (NU) in Baker, et al. (GMD, 2015) that passed (but that Baker et al. claim should have failed) was independently verified by the authors with SIEVE?*

Response: The collection of experiments presented in the revised manuscript is slightly different from that in the discussion paper: results from Cori at NERSC and Constance at PNNL are removed; two cases with code modifications following Milroy et al. (2016) are added (DM and P); three cases (RH-MIN-LOW-2, RH-MIN-LOW-3, and QSMALL) are removed. We point out at the beginning of Sect. 4 ("Numerical experiments") that our strategy is to repeat representative test cases from Baker et al. (2015) and Milroy et al. (2016), and expect the TSC method to give the same "pass" or "fail" results as those from CAM-ECT, with 2 exceptions:

- the NU case is expected to fail TSC, and
- for the P ("Precision") case from Milroy et al. (2016) which has been determined by CAM-ECT to produce consistent climate, we note down an "unknown" for the expected outcome of TSC, due to the deterministic nature of the TSC method and the use of double-precision output in the calculation of the test diagnostics.

In Sect. 4, when introducing the parameter perturbation experiments from Baker et al. (2015), we note that according to that paper, this list of parameters were provided by climate scientists; the parameter changes were thought to affect the model climate in a non-trivial manner, and were intended to be used in different model configurations (e.g. with different resolutions). Therefore we did not apply the SIEVE method to independently verify those test cases. Our understanding of the linkages and distinctions

among the different test methods are explained in a new section ("5.3 Comparison with other test methods") of the revised manuscript.

Comment: *(14) Followup to (12): Recommend that the authors come up with another example of a small scale change that CAM-ECT would not catch because of its use of the global and annual mean (but that TSC would) - other than the NU test from Baker et al. (GMD, 2015). This would probably have to be more subtle than the experiments in Baker, et al. (GMD, 2015). I think this recommendation is particularly pertinent given the list of desired features on page 2 (and that TSC should achieve #6 while CAM-ECT will not).*

Response: Since the test diagnostics of TSC are calculated from instantaneous grid-point-wise model output while CAM-ECT uses global and annual averages, we believe it is reasonable to expect that the former has a larger chance to catch regional differences in the solutions. The NU case has provided evidence to support this reasoning. It is worth noting we also stated in the manuscript that

"On the other hand, since a large number (120) of model output variables are used in CAM-ECT and the simulations are relatively long (1 year), the chance of missing a climate-changing modification (i.e. getting a false 'pass') is relatively small."

We agree that further examples of small-scale solution changes would be informative, but they would not affect the key messages we are trying to deliver in this manuscript. In a more generally sense, it would be useful to compare TSC with CAM-ECT using additional (more subtle and challenging) test cases so as to further understand the strengths and limitations of either method. The motivation for carrying out future work on this topic is explained in Sect. 5.3 ("Comparison with other test methods") of the revised manuscript. We would be delighted to collaborate with the CAM-ECT developers in that effort.

Comment: *(15) page 7, line 16: A false negative example would be a great addition and improve the readers understanding of the tools scope.*

Response: As mentioned earlier in the response to general comment #(2), bugs in "diagnostic-only" parts of the model code, e.g., the calculation of daily maximum 2-m temperature, or the implementation of a satellite simulator, would not be caught by TSC. We point this out in Sect. 2.1 ("Purpose and scope") of the revised manuscript. Another type of false negative is discussed in our response to the next comment.

Comment: *(16) Section 3.2: The splitting into the two domains could be explained more (it is discussed a bit again later in 4.1). It seems a bit arbitrary and suggests that the DUST and CONV-LND failures cannot be detected otherwise. One issue is that by splitting into domains (effectively doubling the number of variables), the false positive rate is being increased. Would be helpful to have more guidance on variable selection and limitations.*

Response: The following paragraph is added to Sect. 3.2 ("Test procedure"):

"Time step size affects the numerical solution at every time step and every grid point,

while certain atmospheric processes might occur in isolated regions thus impacting only a limited number of grid points during very short simulations. Consequently, subtle but systematic solution changes can be masked by the model's time stepping error and can be difficult to detect. To help address this challenge, we calculate RMSDs for $N_{\mathrm{dom}} = 2$ domains, i.e., land and ocean, separately. This is a practical and somewhat arbitrary choice aiming at increasing the sensitivity of the TSC test."

The following two paragraphs are added to Sect. 5.1 ("Test setup" under "Discussion"):

"The TSC test procedure described in this paper has multiple parameters that can be modified: (1) ensemble size, (2) initialization strategy (e.g., simulation start time), (3) time step sizes, (4) integration length, (5) prognostic variables and model sub-domains included in the calculation of test diagnostics, and (6) the pass/fail criterion. Results presented in the previous section indicate that given (1)-(3), the choices for (4)-(6) can have strong impacts on the outcome of the TSC test.

In the DUST case, for example, systematically positive ΔRMSD was detected only in one prognostic variable and only over land (cf. Fig. 5c for results at $t = 5\,min$; results at later time are similar thus not shown). If we had not included aerosol concentrations in the list of monitored variables, or had not chosen to calculate the test diagnostics over land and ocean separately, the TSC test would have given a false "pass" (i.e., a false negative result). While the limited number of test scenarios included in this study have been categorized as expected by the current test setup, there might be more subtle cases, e.g., minor bugs in the code, that require further adjustment of aspects (4)-(6). As a next step, we plan to include a number of bug fixes and additional parameter modifications from the recent model development activities to further evaluate the TSC test setup. "

As for the impact of the number of variables on the test results, we point out in Sect. 3.2 ("Test procedure") that the typical values of $\mathcal{P}_{\mathrm{min},t}$ depend on the number of monitored variables (i.e., larger $N_{\mathrm{var}} \times N_{\mathrm{dom}}$ can result in smaller $\mathcal{P}_{\mathrm{min},t}$ in a statistical sense), hence $\mathcal{P}_0$ needs to be determined empirically for any given $N_{\mathrm{var}} \times N_{\mathrm{dom}}$. Ideally $\mathcal{P}_0$ should be small enough to reduce the chance of false positive (i.e., insignificant solution differences being assigned a "fail"), and large enough to reduce the chance of false negative (i.e., subtle but systematic solution differences being assigned a "pass"). In the present paper we have made an empirical and somewhat arbitrary choice. Further evaluation of the choice and possible improvement of the overall pass/fail criterion are topics of future work. Furthermore, in order to help reduce the false positive rate, we have modified the overall pass/fail criterion in the revised manuscript and propose to fail a test ensemble if $\mathcal{P}_{\mathrm{min},t} < \mathcal{P}_0$ for all output steps in a time window $[X_0, X]$, where X is the total simulation length and X_0 is the spin-up time. The use of multiple time steps in the overall pass/fail criterion reflects our perspective of viewing the model integration as a time evolution problem, and our attempt to distinguish significant and insignificant solution differences based on the characteristics of the $\mathcal{P}_{\mathrm{min},t}$ time series. We point out in the manuscript that the proposed pass/fail criterion was empirically chosen, and the choice can be further

evaluated in future work (cf. Sect. 3.2 and Sect. 5.1).

Comment: *(17) page 8, lines 16-28: I'm still struggling a bit with understanding the scope, which is discussed again here in terms of what will and won't be caught (e.g., aerosol concentrations). Please clarify earlier and consider including supporting experimental results.*

Response: We have rewritten Sect. 2.1 to clarify the purpose and scope of the test. We also added a sentence in the abstract to point out that the test is not exhaustive since it does not detect issues associated with diagnostic calculations that do not feedback to the model state variables. In this section (Sect. 3.2) we have added the comment that additional variables of type I defined in Sect. 2.1 can be added to the list of monitored variables, and a longer variable list might help increase the sensitivity of the test. Type-II variables can also be added if the user wishes to cover the respective code pieces. The TSC method is flexible in this regard, although we emphasize again that only prognostic variables of type I and type II can be included in the list. The concept of time step convergence does not apply to variables that are not calculated using an evolution equation.

Comment: *(18) page 9, line 10: If the implicit assumption that the random variables (mu-sub-j) are Gaussian distributed is violated, will the TSC test results be affected? (And has this been explored? An example could be something like truncation...)*

Response: We have not explored this. The manuscript only describes the first implementation of TSC and provides evidence that it is a useful method. The test setup can be further evaluated and improved in the future.

Comment: *(19) page 9, Step 3 (line 30 -> page 6): More clarification is needed here. For the t-test, the choice of .05% is conservative (as acknowledged in text), and it is clear that the specified t-statistic (4.437) is dependent on both the .05% cutoff *and* the sample (ensemble) size (M=12). However, there is a less intuitive dependence on the number of variables that should be pointed out (and discussed). Because the t-test is performed on each variable *individually*, then the number of variables examined certainly affects the overall test failure rates. The conservative choice of .05% may make sense for the 20 variable subset (meaning that a single variable has to fail quite badly to cause a failure of the overall test). However, if one were to use 2 variables (or 100 variables), the .05% may no longer be the best choice. I think this should be addressed given the discussion on page 8 (line 25) that one could choose to include more (and presumably fewer) fields.*

Response: We agree with the referee's comment. We point out in Sect. 3.2 ("Test procedure") of the revised manuscript that the typical values of $\mathcal{P}_{\mathrm{min},t}$ depend on the number of monitored variables (i.e., larger $N_{\mathrm{var}} \times N_{\mathrm{dom}}$ can result in smaller $\mathcal{P}_{\mathrm{min},t}$ in a statistical sense), hence $\mathcal{P}_0$ needs to be determined empirically for any given $N_{\mathrm{var}} \times N_{\mathrm{dom}}$. Ideally $\mathcal{P}_0$ should be small enough to reduce the chance of false positive (i.e., insignificant solution differences being assigned a "fail"), and large enough to reduce the chance of false negative (i.e., subtle but systematic solution differences being assigned a "pass"). In the present paper we have made an empirical and somewhat arbitrary choice. Further

evaluation of the choice and possible improvement of the overall pass/fail criterion are topics of future work. Furthermore, in order to help reduce the false positive rate, we have modified the overall pass/fail criterion in the revised manuscript and propose to fail a test ensemble if $\mathcal{P}_{\min,t} < \mathcal{P}_0$ for all output steps in a time window $[X_0, X]$, where X is the total simulation length and X_0 is the spin-up time. The use of multiple time steps in the overall pass/fail criterion reflects our perspective of viewing the model integration as a time evolution problem, and our attempt to distinguish significant and insignificant solution differences based on the characteristics of the $\mathcal{P}_{\min,t}$ time series. We point out in the manuscript that the proposed pass/fail criterion was empirically chosen, and the choice can be further evaluated in future work (cf. Sect. 3.2 and Sect. 5.1).

Comment: *(20) page 9, line 9): Please clarify the reason for the directional t-test and consider updating/clarifying the accompanying discussion on page 9,line 25 -> page 10, line 4.*

Response: We point out in Sect. 3.2 that "One-sided test is used here because of the concept of self-convergence explained in Sect. 2.3: When bugs are introduced, or when the code is not compiled or executed correctly, the simulation will not solve the originally intended equations, thus not converging to the reference solutions produced by the original code or environment, resulting in larger RMSDs."

Comment: *(21) page 9, line 10: A minor point, but technically one cannot "accept" the null hypothesis. (One can fail to reject the null hypothesis or reject it.)*

Response: The revised sentence reads "The students t-test is performed on the null hypothesis that $\mu_{j,t}$ is statistically zero".

Comment: *(22) page 11, line 1: Was the .89 vs. .897 detectable by SIEVE? Also how long of a simulation was run for SIEVE in this case?*

Response: 10-year simulations were conducted and compared with a control simulation using the AMWG diagnostics. The case of 0.897 was indistinguishable from the control by SIEVE using the standard plots, but given the rather direct impact of this parameter on the cloud formation in the model, we thought the difference might be detectable by additional metrics. The expected "fail" was rather an educated guess that was later confirmed by TSC.

In order to focus the manuscript on the essential message, in the revised version we no longer show results from the three parameter perturbation cases RH-MIN-LOW-2, RH-MIN-LOW-3, and QSMALL, but add two cases with code modifications following Milroy et al. (2016). We point out at the beginning of Sect. 4 ("Numerical experiments") that our strategy is to repeat representative test cases from Baker et al. (2015) and Milroy et al. (2016), and compare the results from TSC and CAM-ECT.

Comment: *(23) page 10: Given the FMA issues found for Mira in Milroy et al. 2016 (and also for BlueWaters), I am questioning the Cori results a bit - also because the results in Table 1 for Cori are not as definitive as for the other machines. Cori uses*

FMA by default, or was it disabled for these experiments? How long were the simulations examined by SIEVE for Cori?

Response: As explained earlier in the response to specific comment #(5), we think the FMA issue is an interesting one worth further investigation. There is the possibility that the impact of FMA is far below the magnitude of the time stepping error in very short simulations thus not detectable by the TSC setup described in the manuscript. In that case, the use of multiple test methods might help better understand the impact of the FMA issue from different angles. Since the case is not yet well understood, and the Cori example is not essential for demonstrating the basic idea and utility of the TSC method, we do not show the Cori results in the revised manuscript.

A separate comment on the Cori result: in Table 1 of the discussion paper, the $\mathcal{P}_{\min}$ values were shown only at 5 minutes and 30 minutes after model initialization. While the two numbers from Cori were indeed less definitive than those from the other machines, from the complete time series shown in Figure 6 of the discussion paper, the Cori results seem less suspicious. This made us realize that "pass/fail" criteria based on results at a single time instance are more likely to lead to false positives and negatives. In the revised manuscript, we have modified the overall pass/fail criterion and propose to fail a test ensemble if $\mathcal{P}_{\min} < \mathcal{P}_0$ for all output steps in a time window $[X_0, X]$, where X is the total simulation length and X_0 is the spin-up time. The use of multiple time steps in the overall pass/fail criterion reflects our perspective of viewing the model integration as a time evolution problem, and our attempt to distinguish significant and insignificant solution differences based on the characteristics of the $\mathcal{P}_{\min}$ time series. We point out in the manuscript that the proposed pass/fail criterion is an empirical, simple, and preliminary choice. It can be further evaluated in the future (cf. Sect. 3.2 and Sect. 5.1).

Comment: *(24) page 13, line 25: "...failing the test will very likely mean the climate will be different. Passing the convergence test should hence be considered a necessary condition..." I don't quite agree with this. Many of the parameterizations could be quite different in the short term (because of sensitivity), but the longer term behavior is basically the same. In other words, the weather after 150s may look different (e.g., raining or not), but the annual climate is the same. (This assertion is also made on page 7, lines 15-16)*

Response: It sounds like the referee was thinking about chaos and predictability. Since the TSC test only looks at a time window of a few minutes to an hour, We believe the problem should be sufficiently deterministic. The respective sentence has been removed, and our understanding of the linkages and distinctions between TSC and CAM-ECT is explained in Sect. 5.3 ("Comparison with other test methods" under "Discussion").

Comment: *Technical Corrections*
(1) page 2, line 3: Remove the final word "did" from the sentence.
(2) page 2, line 29: The second occurence of "simulation" should be plural.
(3) page 3, line 9: Spell out the number 3 (three).
(4) page 3, lines 23-25: Consider breaking this sentence into smaller parts.

(5) page 5, line 8: "thus saves" should be "thus saving".
(6) page 5, line 18: "dependences" should be "dependencies".

Response: Thanks for pointing out the errors. The respective sentences have been revised. In some cases the entire paragraph has been rewritten.

Comment: *Final thoughts. I like the idea of this work, and I hope that the comments and suggestions provided will be helpful for the revision of the paper. I believe that more flushed out algorithm details, a clarification of scope, and better alignment of the experimental results with the stated features of the test will strengthen the paper and its impact and utility.*

We thank the referee for the detailed and very helpful review. The questions and suggestions, together with the comments from the other referee and from Dr. Sacks, prompted us to think deeper about our method. We have made a substantial revision of the manuscript to clarify the purpose and scope of our method (Sections 1 and 2.1), and to explain our understanding of the relationship between TSC and other methods (Sections 1 and 5.3). We also added comments and discussion on the details of the test design (e.g., test diagnostics, method of statistical testing, and pass/fail criterion), and acknowledge that they can be further evaluated and improved (Sections 3.2 and 5.1). We intend to continue this work and obtain more comprehensive understanding of the strengths and limitations of the TSC method.

Reply to Referee #2

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. In the revised manuscript, we present results from the new simulations in Sect. 4 ("Numerical experiments") for the evaluation of the TSC method. In Sect. 2.3, when explaining the concept of time step convergence, we still present the old results but add a paragraph to point out that the calling frequency of radiation does not change the convergence property of the CAM5 model.

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). A new figure (Figure 6) is added to Sect. 5.2 of the revised manuscript together with a discussion on the impact of noisy parameterization 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 (Sect. 1) 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 conducted 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 "Introduction" section of 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. 2a 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. 2 below and in Fig. 1 in the revised manuscript), 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 sizable 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. 2b 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 particular 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

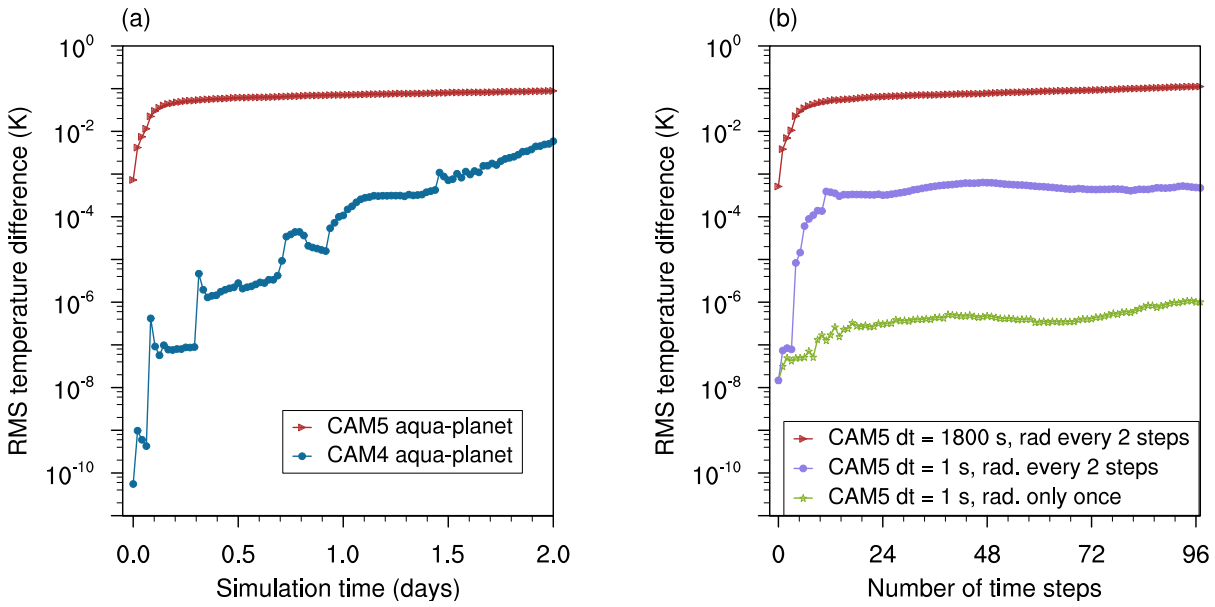

Figure 2: Examples of the evolution of RMS temperature difference (unit: K) caused by random perturbations of order 10^{-14} K imposed on the temperature initial conditions. (a) Aqua-planet simulations conducted with the CAM4 (blue) and CAM5.3 (red) physics parameterization suites using the default 1800 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.

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 similar situations to occur.

The explanations above are included in Sect. 1 of 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 $\log_{10}\mathrm{RMSD}$ 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 Sect. 3.2 ("Test procedure"):

"Time step size affects the numerical solution at every time step and every grid point, while certain atmospheric processes might occur in isolated regions thus impacting only a limited number of grid points during very short simulations. Consequently, subtle but systematic solution changes can be masked by the model's time stepping error and can be difficult to detect. To help address this challenge, we calculate RMSDs for $N_{\mathrm{dom}} = 2$ domains, i.e., land and ocean, separately. This is a practical and somewhat arbitrary choice aiming 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 port 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.

In Sect. 1, Sect. 2.1, and Sect. 5.3, we clarify that PerGro, TSC, and CAM-ECT are all regression testing methods for verifying that results from the CAM model stay the same despite changes in the code or in the computing environment. Since the different methods assess the magnitude of solution change with different criteria and at different time scales, we expect there will be situations when they give different answers. Our understanding of the linkages and distinctions among the different methods is explained in Sect. 5.3.

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.

A new and inexpensive non-bit-for-bit solution reproducibility test based on time step convergence (TSC1.0)

Hui Wan[1], Kai Zhang[1], Philip J. Rasch[1], Balwinder Singh[1], Xingyuan Chen[1], and Jim Edwards[2]

[1]Pacific Northwest National Laboratory, Richland, WA, USA
[2]National Center for Atmospheric Research, Boulder, CO, USA

Correspondence to: Hui Wan (Hui.Wan@pnnl.gov)

Abstract. A test procedure is proposed for identifying numerically significant solution changes in ~~atmospheric models that solve the partial differential equations of fluid dynamics~~evolution equations used in atmospheric models. The test issues a "fail" signal when any code modifications or computing environment changes lead to solution differences that exceed the known time step sensitivity of the reference model. ~~It is demonstrated~~ Initial evidence is provided using the Community Atmosphere Model (CAM) version ~~5.3 (CAM5.3)~~ 5.3 that the proposed procedure can ~~correctly~~ be used to distinguish rounding-level ~~changes in the solutions~~ solution changes from impacts of compiler optimization or parameter ~~perturbations~~ perturbation that are known to cause ~~non-negligible~~ substantial differences in the simulated climate. The test is not exhaustive since it does not detect issues associated with diagnostic calculations that do not feedback to the model state variables. Nevertheless it provides a practical and objective way to assess the significance of solution changes. The short simulation length implies low computational cost~~,~~ ~~and makes the test useful for debugging~~. The independence between ensemble members allows for parallel execution of all simulations thus facilitating fast turnaround. ~~The version 1.0 implementation described in the present paper uses 12-member 5-minute simulations. The computational cost of producing the reference results is close to a 4-month simulation conducted using the default model time step, and the cost of testing a new code or computing environment is close to a 1-month simulation conducted using the default model time step.~~ The new method is simple to implement since it does not require any code modifications. We expect that the same methodology can be used for any geophysical model to which the concept of time step convergence is applicable.

1 Introduction

~~Verification and validation are indispensable steps in the development of a numerical model. According to the widely accepted definitions in IEEE and related communities (e.g. **??**), verification is the process to substantiate that a numerical model represents the intended conceptual model, while validation is the process to determine whether the numerical model is a sufficiently accurate representation of the targeted real-world system. The task of verification can be further divided into numerical algorithm verification and software quality assurance (**?**). The present paper addresses the latter topic, and proposes a new method for testing the reproducibility of the numerical solution.~~

~~Numerical models for weather and climate research and prediction, like the Community Atmosphere Model (CAM, Neale et al., 2010, 2012), undergo constant changes and improvements both in their source codes and in the computing environments. In a small part of the model developers' daily work, it is possible to assure a model has been compiled and executed correctly by demonstrating that a newly conducted simulation produces results that are bit-for-bit (BFB) identical to those of a previously verified simulation. More often, however, software or hardware updates as well as code optimization, extension, or refactoring inevitably lead to the loss of BFB reproducibility. In such cases, a necessary step of code verification is to assess whether the new solutions still represent the same characteristics of the atmospheric motions as the old solutions did. The complexity of the state-of-the-art weather and climate models makes it a nontrivial task to perform such verification in an efficient and objective manner (Baker et al., 2015).~~

~~For the Community Atmosphere Model (CAM), a porting verification procedure called the perturbation growth test (hereafter the PerGro test, cf. Rosinski and Williamson, 1997) had been used to evaluate non-BFB changes till version 4 of the model (Neale et al., 2010; **?**). The method involves comparing one test simulation and two trusted simulations over the course of 2 model days. The differences between the two trusted simulations are caused by random temperature perturbations of order 10^{-14} K introduced to the initial conditions of one of the runs, and the solution differences are quantified by the spatial root-mean-square differences (RMSD) in the temperature field at each time step. When the evolution of the RMSD between a test simulation and either of the trusted simulations deviates substantially from the evolution of RMSD between the trusted simulations, the verification is considered a failure; if the two RMSD time series appear to be quantitatively similar, the presumption is that the simulations are equivalent, and the new simulation is regarded as "verified".~~

The Community Atmosphere Model (CAM, Neale et al., 2010, 2012), like all other general circulation models (GCMs) used for weather and climate prediction and research, is a large body of computer code that solves a system of differential, integral, and algebraic equations. Testing the code to ensure it behaves as expected involves a wide range of efforts that touch upon the formulation of the equations, the solution algorithms, and the software design and implementation. This paper addresses the issue of regression testing, i.e., verifying that results from the model stay the same despite changes in the code or the computing environment. In certain cases, it is possible to achieve this goal by demonstrating that a newly conducted simulation produces bit-for-bit (BFB) identical output compared to a simulation previously certified to be valid. More often, however, software or hardware updates as well as code optimization or refactoring inevitably lead to the loss of BFB reproducibility, in which case a different criterion is needed to declare two simulations as "the same". The large number of equations in an atmospheric GCM and the nonlinearities of the equation set make it a challenging task to define such a criterion.

Since CAM is a climate model, one possibility could be to require that the long-term statistics of the atmospheric motions be representative of the climate simulated by the old code in the old environment (see, e.g., "Condition 3" in Rosinski and Williamson, 1997). One procedure to make such an assessment could be a "Subjective Independent Examination and Verification by Experts", or SIEVE, that consists of experienced climate modelers performing multi-year simulations and examining many fields of the model output to determine whether the simulated climate has changed or not. This procedure is unsatisfactory due to its subjectivity and the high computational cost, but we speculate this is the most widely used method in many modeling groups. Recently, Baker et al. (2015) developed an Ensemble-based Consistency Test (ECT) as a replacement of SIEVE, which we

refer to as CAM-ECT following Baker et al. (2016). CAM-ECT involves first generating a reference ensemble of one-year simulations on a trusted computer with an accepted version and configuration of CAM, and creating a statistical distribution that characterizes the ensemble using principal component analysis (PCA) of the globally averaged annual mean fields. To test a new code or computing environment, a small ensemble of one-year simulations is conducted, and the CAM-ECT tool determines whether the new simulations are statistically distinguishable from the reference ensemble. Compared to SIEVE, CAM-ECT is a major step forward in regression testing since it clearly defines an objective criterion for “pass" or “fail". The use of PCA allows the test diagnostics to include all variables written out by the model, resulting in rather complete code coverage. As demonstrated by Baker et al. (2015) and Milroy et al. (2016), the method is able to detect the impact of parameter changes in the model source code as well as issues in the computing environment. The main limitation of CAM-ECT lies in its computational cost. In the original implementation described by Baker et al. (2015), the reference ensemble consisted of 151 members and the test ensemble included 3 simulations. A follow-up study by Milroy et al. (2016) proposed using 453 simulations from multiple compilers to provide sufficient variability in the reference ensemble. Since the reference ensemble needs to be updated every time a new code version with different climate characteristics is selected as the baseline for further model development (e.g., after a climate-changing bug fix), the large ensemble size can be a substantial burden in computational cost, especially during very active model development phases.

Given that the purpose of the regression testing is to assure the model results stay the same, rather than to provide a descriptive characterization of the simulated physical phenomena, it would be useful to have additional test methods that can give early warnings of unexpected model behavior using computationally inexpensive simulations. The perturbation growth test (hereafter PERGRO) based on the work of Rosinski and Williamson (1997) is an example that assesses the short-term behavior of the model results. PERGRO was originally designed to verify the simulations after a predecessor of CAM was ported to different computers. More generally, the method has been used to verify that code modifications only produced roundoff-level changes in the model results.

The PERGRO test involved comparing one test simulation and two trusted simulations over the course of two model days. Solution differences were quantified by the spatial root-mean-square differences (RMSD) in the temperature field at each time step. The differences between the two trusted simulations were triggered by random temperature perturbations of order 10^{-14} K introduced to the initial conditions in one of the simulations. Rosinski and Williamson (1997) established two conditions for the verification 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 the initial perturbation.

It is worth noting 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, 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 port 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. ~~The method gradually became less useful as~~ As the model became more comprehensive and complex, ~~and compromises~~ more rapid growth of rounding-level initial perturbation was observed. 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. Unfortunately, even those compromises are no longer adequate for CAM5. 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. An example of the current situation is shown in Fig. 1 by the red curve which depicts a typical evolution of the temperature RMSD triggered by $\mathcal{O}(10^{-14})$ K initial perturbation in CAM5.3. For comparison, the characteristic perturbation growth of the CAM4 physics parameterization suite (Neale et al., 2010) is shown in blue. All the simulations were conducted using the spectral element dynamical core (Taylor and Fournier, 2010; Dennis et al., 2012) at approximately 1° horizontal resolution. Fig. 1 indicates that after the first time step (30 min), the RMSD in CAM5.3 is already 7 orders of magnitude larger than that produced by the CAM4 physics package. The RMSD in CAM5.4 after 4 time steps (2 hours) is larger than the RMSD in CAM4 after 2 days. It is now very difficult to distinguish differences between a test and trusted simulation from differences between two trusted simulation even after a single time step.~~ If those pieces of codes were not switched off or revised, perturbations on the trusted machine would grow so rapidly that the RMSD would reach $\mathcal{O}(0.1)$ K 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 RMSD grew at a rate well below one order of magnitude per timestep in a trusted environment. An example is shown in Fig. 1a with the blue curve. 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 verification effort. But with CAM5, initial perturbations grow too rapidly even in an aqua-planet simulation (Fig. 1a, red curve), making the original PERGRO method no longer useful for port verification.

~~The very fast evolution of initial perturbation is caused by multiple factors.~~ Rosinski and Williamson (1997) noted that dynamical-core-only simulations typically showed much slower growth of initial perturbation, and this characteristics remains true in newer model versions. For example, using the default configuration of CAM5's spectral element dynamical core

(Taylor and Fournier, 2010; Dennis et al., 2012) at 1 ° spatial resolution, the temperature RMSD only reaches $\mathcal{O}(10^{-12})$ K by day 2, suggesting that the rapid growth shown in Fig. 1a is due to the physics parameterizations. Efforts have been made to understand the cause of the rapid growth, and those findings will be detailed in a separate manuscript. [1] Here we provide only a brief description of three causes:

First, the default time step of 1800 s in CAM5 is sizable compared to the characteristic time scales of many physical processes represented by the model, so the increments in the model state during one time step (i.e., 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 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. Longer model time steps lead to larger increments from the simulated physical processes, but not necessarily so for software or hardware issues. Therefore the growth of perturbation in a model with long time step can make it difficult to expose solution differences caused by a new computing environment.

The second reason for rapid perturbation growth is related to the fact that the radiation parameterization in CAM5 uses a pseudo random number generator, and the seeds for the generator are chosen from the less significant digits of the pressure field. This effectively introduces state-dependent noise into the numerical solution. The green curve in Fig. 1b 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 more generally, the radiation example also implies that models with state-dependent stochastic parameterizations might feature more rapid perturbation growth than those using deterministic schemes.

The third reason for rapid perturbation growth has to do with particular pieces of code. Two types of examples were discussed in Rosinski and Williamson (1997): (i) an upshift in digit of solution difference 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 by 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 similar situations to occur.

~~It would be desirable to revise the physics parameterizations in the CAM5 model to obtain a code that behaves more like CAM4 or their predecessors in terms of rounding-error growth. Recent work by Singh et al. has addressed some of those issues, but it also has shown that the process of identifying the culprits, revising the code, and assessing the impact on the simulated model climate can be rather time-consuming.~~ The examples shown in Fig. 1b indicate that it is possible to identify reasons for perturbation growth, with the potential to make PERGRO a useful testing method again, although experience shows that such efforts can be rather substantial and time-consuming. We will document that path elsewhere. ~~Therefore we believe it will~~

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

~~also be useful to use methods that can test~~ In the present paper, we describes a strategy that tests the code "as is" so that new parameterizations and code updates can be assessed as soon as they enter the model.

~~Recently, Baker et al. (2015) developed an Ensemble-based Consistency Test (ECT) as a replacement for the PerGro test. Their new test, hereafter referred to as CAM-ECT following Baker et al. (2016), abandons the idea of monitoring the gradual growth of a small perturbation. Instead, their method quantifies the consequenceof such growth as manifested in the globally averaged annual meanof a large number of model output variables in climate simulations. The test procedure involves first generating a reference ensemble of 151 one-year simulations on a trusted machine with an accepted version and configuration of CAM, and creating a statistical distribution that characterizes the ensemble using principal component analysis of the globally averaged annual meanfields. To test a new code or computing environment, 3 one-year simulations are conducted, and the CAM-ECT tool determines whether the new simulations are statistically distinguishable from the reference ensemble. Baker et al. (2015) showed that CAM-ECT is capable of detecting impacts of model parameter changes as well as errors in the software and hardware environments.~~

~~In this paperwe propose a complementary test procedure that builds~~ The new test procedure is based on the work of Wan et al. (2015) on ~~the~~ time step convergence in CAM5. ~~The new test is also ensemble-based, but takes a deterministic perspective and focuses on short-term behavior of the numerical solution. The independent ensemble members are obtained differently than done in CAM-ECT, and the computational cost is substantially lower. The remainder of the paper introduces the test philosophy in~~ The underlying concept and design considerations are explained in Sect. 2. A first implementation of the test in CAM5 is described in Sect. ~~2, and describes the implementation~~ 3 and evaluated in Sect. ~~3. Evaluation of the test procedure is~~ 4. Further discussions on the test design and its relationship to other methods are presented in Sect. ~~4. The conclusions~~ 5. Conclusions are drawn in Sect. 6.

2 Test philosophy

In this section, we start with a ~~further~~ clarification of the purpose ~~of the code verification procedure~~ and scope of the new test method (Sect. 2.1), then proceed to a discussion of the desirable features that guided the design of our ~~new method~~ test (Sect. 2.2). The underlying concept of the new method is explained in Sect. 2.3~~, with additional discussions presented in Sect. 2.4.~~ .

2.1 ~~Scope~~Purpose and scope

~~As mentioned earlier, the purpose of the code verification task discussed here – from a perspective of climate model development – is to substantiate whether the climate characteristics simulated by a model remain the same when code modifications or computing environment updates lead to the loss of BFB reproducibility. From a mathematical perspective, the essence of the task is to determine whether numerical solutions to the model equations remain the same when the accuracy limits related to the algorithmic implementation are taken into account. Hence the scope of the present paper is restricted to the equation-solving part of a climate model, i.e., the discrete formulation of the model equations and how they are coded to carry~~

~~out time integration. While the CAM code also includes additional functionalities such as various diagnostics and flexible I/O options, those pieces of code do not directly affect the solution procedure, thus are not targets of this study.~~

As stated earlier, the topic of this paper is regression testing under circumstances when results from an atmospheric GCM are no longer BFB reproducible. In other words, the testing discussed here aims at substantiating whether results from an atmospheric GCM stay the same after supposedly minor code modifications or computing environment changes. By "minor code modifications" we mean code refactoring, optimization of the computational efficiency, or any other code changes that might alter the sequence of computation but still solve the same set of equations using the same mathematical algorithms. Computing environment changes refer to any changes in the hardware or software configuration in which the model code is compiled and executed. Two factors need to be considered when designing a method for regression testing: (i) the variables that represent the outcome of a simulation, and (ii) a criterion for declaring two simulations as "the same". In the present paper, we consider the outcome of a simulation unchanged if the numerical solution is found to have the same time stepping error relative to a reference solution obtained with a previously verified code and computing environment. The details are explained later in Sect. 2.3. The reasoning behind our choice for element (ii) is explained below.

From the perspective that a GCM is a suite of algorithms solving a large set of differential, integral, and algebraic equations, the physical quantities (model variables) calculated by the code can be sorted into 3 categories:

I. Prognostic and diagnostic variables whose equations are coupled to one another such that any change in variable A will, within one time step or after multiple time steps, affect variable B in this same category. Examples in this category include basic model state variables like temperature, winds, and humidity, as well as quantities calculated as intermediate products in a parameterization, for example the aerosol water content (which affects radiation and eventually temperature), and the convective available potential energy (which affects the strength of convection hence temperature and humidity).

II. Prognostic variables that are influenced by type-I variables but do not feedback to them. An example could be passive tracers carried by the model to investigate atmospheric transport characteristics (e.g., Zhang et al., 2008; Kristiansen et al., 2016)

III. Diagnostic quantities calculated to facilitate evaluation of a simulation, but do not feedback to type I or type II. Examples include the daily maximum 2-m temperature, the total ice-to-liquid conversion rate in the cloud microphysics parameterization (which is calculated merely for output in CAM5), and any variable specific to the COSP simulator package (Bodas-Salcedo et al., 2011).

We take the standpoint that the essential characteristics of the simulated atmospheric phenomena are determined and represented by type-I variables. If instantaneous and grid-point values are monitored, any significant solution change should be detectable through the monitoring of a single variable in type I, per definition of that variable type, as long as the simulations are long enough for the impact to propagate and evolve to a discernable signal in that monitored variable. On the other hand, since we are taking a deterministic perspective here, the simulations need to be sufficiently short to avoid chaos.

Based on the reasoning above, the test diagnostics of our new method are calculated from a small set of prognostic variables of type I. The use of multiple variables is meant to help increase the sensitivity of the test (decrease the chance of failing to

detect a significant solution change), since bugs or issues associated with a specific piece of code might take longer time to cause discernable solution differences in one variable than in another. In Sects. 3 and 4 where we describe and evaluate the first implementation of our method in CAM5, the monitored variables include a few basic atmospheric state variables plus aerosol and hydrometeor concentrations. We note that this choice of variables can be further evaluated or tailored to meet the user's needs. The test method can also be extended to include variables of type II, but cannot be used on type-III variables or diagnostic variables in type I, because the concept of time step convergence does not apply. This means our test does not provide a full coverage of all code pieces in the model. For example, bugs in the implementation of a satellite simulator or other "diagnostic-only" calculations would not be detected by our test. Issues in software functionalities that are not exercised during the simulations, e.g., the reading and writing of restart files, would not be caught, either. We acknowledge that the proposed test method is not exhaustive; but given its simplicity, low computational cost, and the effectiveness demonstrated in Sect. 4, we believe it is a practical and promising method for assessing the magnitude of solution differences in complex models.

2.2 Desirable features

~~One way to accomplish the above-mentioned code verification task could be a "Subjective Independent Examination and Verification by Experts", or SIEVE, which consists of experienced climate modelers performing multi-year simulations and examining many fields of the model output to determine whether the simulated climate has changed or not. This procedure is unsatisfactory due to its subjectivity and the high computational cost, but we speculate this is the most widely used method in many modeling groups.~~ Given the continuously growing complexity of the modern ~~climate models~~ atmospheric GCMs and the need by large groups of model developers ~~/~~and users to perform ~~code verification~~ regression testing routinely (e.g. on a daily basis), it is desirable to have test procedures that have the following features:

1. Objective;
2. Easy to perform and automate;
3. Requiring no or minimum code modifications;
4. Exercising the entire model in its "operational" configuration;
5. Also applicable to a subset of the code thus useful for debugging;
6. Capable of detecting changes in both global and/or regional features of the simulations;
7. Insensitive to roundoff differences associated with changes in the order of accumulations or ~~commutative~~ associative operations, etc;
8. Computationally efficient.

The CAM-ECT of Baker et al. (2015) fulfills criteria 1–4 and ~~7.~~ 7, and partly 5. For criterion 5, we expect CAM-ECT to be capable of isolating issues associated with variables of type II or III (cf. Sect. 2.1) through systematic elimination of model

output variables from the test diagnostics (Milroy et al., 2016). Bugs associated with type-I variables would be more difficult to pinpoint: since all variables in this type are inherently coupled, we expect that any substantial change in one equation would have affected all the type-I variables after a year of model integration. One-year simulations might also be challenging for a code that is still in debugging stage thus numerically unstable for long simulations. The use of global annual averages ~~in the results assessment can~~ by CAM-ECT might lead to difficulty in detecting changes in small-scale features (criterion 6)~~, as~~ . For example, Baker et al. (2015) noted that CAM-ECT did not identify the impact of a ~~change in a~~ perturbed horizontal diffusion parameter ~~in the dynamical core~~ as "climate-changing" (see case NU discussed in Sect. 4.3 therein). On the other hand, since a large number (120) of model output variables are used in CAM-ECT and the simulations are relatively long ~~(1 year)~~thus allowing ample time for the impact of a bug or system issue to evolve and propagate, the chance of missing a climate-changing ~~modification~~ feature (i.e. getting a false "pass") is relatively small. The main limitation of CAM-ECT lies in its computational cost (criterion 8)~~. Moreover, since each ensemble member is a one-year simulation, it is unlikely that the method can be used to test a small subset of the model components, or a code that is still in debugging stage thus numerically unstable for long simulations (criterion 5).~~, as already mentioned in Sect. 1.

The ~~PerGro~~ PERGRO test of Rosinski and Williamson (1997) fulfills criterion 7 per design~~; it is very efficient in terms of~~ . The use of 2-day simulations translates to very low computational cost thus fulfilling criterion ~~8; it~~ 8. the method also satisfies criteria 2, 3, 5, and 6. The aqua-planet setup with a few test-specific code changes leads to a configuration that is very close to the full version of the atmosphere model (criterion 4). The interpretation of the perturbation growth test has some subjectivity (criterion 1), since there is not a quantitative criterion regarding how close the new RMSD curve should resemble the reference curve. However, the ~~modeler~~ developers' experience ~~with CAM4 is~~ was that when a simulation fails the test, "it generally fails spectacularly, i.e., the difference curve will exceed the perturbation curve by many orders of magnitude within a few model timesteps" (http://www.cesm.ucar.edu/models/cesm1.0/cam/docs/port/pergro-test.html). Therefore objectivity is also not a major weakness of the ~~PerGro~~ PERGRO test. The main – and also critical – difficulty with the method is that it is ~~now~~ ill-suited for CAM5 because the ~~initial perturbations amplify so rapidly even in a trusted environment that they cannot be distinguished from model differences caused by compiler or machine problems, making the reference curve (i. e. the red curve in Fig. 1) too relaxed to be useful for code verification.~~ "Condition 0" needed by the test strategy has now been violated.

The new test proposed in this paper aims at ~~fulfilling~~ satisfying all the 8 features listed above. It keeps the deterministic spirit of ~~the PerGro test~~ PERGRO to achieve an early detection of solution differences thus ~~saves computational time, but uses a different method to capture the solution uncertainty related to the non-linear and discrete nature of the model equation set~~saving computational time. Ensemble simulations are conducted to take into account the internal variability of the atmospheric motions. The test design was inspired by the results of Wan et al. (2015), as explained below. In the remainder of the paper, we will refer to the new test method as the Time Step Convergence (TSC) test.

2.3 Time step convergence (TSC)

Wan et al. (2015) evaluated the short-term time step convergence in CAM5 for the purpose of quantifying and attributing numerical artifacts caused by time integration. Starting from the same initial conditions, a series of 1 h simulations were conducted

using time step sizes ranging from 1 s to 1800 s. The numerical solution with $\Delta t = 1$ s was viewed as the proxy "truth", and the time stepping error associated with a longer step size was defined as the RMSD between instantaneous 3D temperature fields after 1 h of model integration. To take into account possible flow-dependencies of the numerical error, the exercise was repeated using initial conditions sampled from different months of a previously conducted ~~long-term simulation~~multi-year simulation, following the idea of Wan et al. (2014). A linear regression was then applied between the ensemble mean $\log_{10}(\mathrm{RMSD})$ and $\log_{10}(\Delta t)$~~to obtain the~~ . The regression coefficient gives the time step convergence rate. Experience so far indicates that the diagnosed convergence rate is rather insensitive to the choice of initial conditions (cf. Sect. 3.2 for further discussion).

In Fig. 2, the 12-member ensemble mean temperature RMSD in the default CAM5.3 model ("CTRL") is shown with blue circles, and the $\pm\sigma$ ranges are shown by vertical bars. Here σ denotes the ensemble standard deviation. The blue regression line indicates a convergence rate close to 0.4. It is important to emphasize that this regression line corresponds to the *self*-convergence, i.e., the convergence towards a solution produced with the same code and a very small step size. When the code is not exercised correctly, or when the model equations have changed because of parameterization update or parameter tuning, convergence towards the original reference solution should no longer be expected. This is the key hypothesis on which our new ~~verification test~~ test method is based.

To demonstrate this point, Fig. 2 also shows results from simulations conducted with a modified parameter in the physics package. Specifically, the grid-box mean relative humidity threshold for the formation of high-level clouds, a parameter called cldfrc_rhminh in the large-scale condensation scheme of Park et al. (2014), was changed from 0.8 to 0.9. This ~~set of simulations are labeled~~ parameter change was used in Baker et al. (2015) in the evaluation of CAM-ECT, and we label it "RH-MIN-HIGH" ~~hereafter~~following that study. The RMSD calculated against a new reference solution using cldfrc_rhminh = 0.9 and $\Delta t = 1$ s is shown in green in Fig. 2. The self-convergence of the modified model turns out to be very similar to the self-convergence in the original model. This is expected, and also consistent with the concept of self-convergence since no structural changes (e.g. parameterization or numerical algorithm modifications) have been introduced into the model. However, when the RMSD of the RH-MIN-HIGH simulations are calculated against the 1 s simulations of CTRL, the RMSD values appear to be considerably larger at smaller step sizes. The discrepancies – caused by the parameter change – far exceed the ensemble spread of the reference solutions. The divergence of the red and blue convergence pathways in Fig. 2 provides a proof of concept that the model's time step convergence behavior can be used as a metric to detect significant changes in the numerical solution. In Fig. 2, the RMSD is shown for a range of step sizes for a better illustration of the concept. In practice, anomalous RMSD at one step size will be sufficient to flag a code or computing environment as failing the expectation that they provide the same numerical solution as the reference code or environment does, although the identification of a "true anomaly" requires an ensemble of independent simulations, which we will demonstrate in Sect. 3.2.

Fig. 2 also indicates that the RMSDs calculated both ways are hardly distinguishable at the default step size, suggesting that the impact of the parameter change is smaller than or similar to the time integration error, at least for this prognostic variable and at the chosen time scale (1 h). If we had introduced larger changes in the model, e.g., by changing cldfrc_rhminh ~~more substantially~~to 0.999 instead of 0.9 from the default value of 0.8, or by replacing a certain parameterization by a different scheme, the impact might be more visible at the default step size. In contrast, if the ~~model change were less substantial,~~

parameter change were smaller, e.g., from 0.8 to 0.82 instead of 0.9, the red and blue convergence pathways in Fig. 2 might not diverge until a step size on the order of a few seconds. In order to establish a highly sensitive ~~code verification procedure~~ regression test that can detect very small solution changes, it would be desirable to find a time step size that corresponds to very small numerical error. The shortest possible step size for CAM5.3 simulations is 1 s~~which corresponds to~~ ; this is the shortest possible interval at which the dynamical core and the various parameterized physical processes interact with each other~~; 1 is~~ , and also the shortest step size the coupler can handle for the coupling between different model components (atmosphere, land, ocean, sea ice, etc.). Hence the new TSC test uses the RMSD between a pair of simulations with 2 s and 1 s time steps as the metric for assessing the magnitude of solution changes.

In the study of Wan et al. (2015), simulations with shortened time step sizes were conducted with all physics parameterizations calculated every time step except for radiation which was called only once (i.e., with a 1 h step size, cf. Table 1 in Wan et al., 2015). The simulations shown in Fig. 2 followed the same design, but we also repeated the simulations with radiation calculated every other time step (as in the default model). The results were hardly distinguishable from Fig. 2 (not shown), suggesting that the calling frequency of radiation does not change the convergence property of the CAM5 model. When describing the TSC implementation in the next section, we propose to calculate radiation every other time step so that the time step ratio is kept the same among all model components. In Sect. 5 we also present results from simulations with radiation calculated only at the first time step, and discuss the impact of noisy parameterization on the TSC results.

We also note that in the earlier study of Wan et al. (2015), convergence analysis was done not only with the full CAM5 model, but also using configurations that exercised the dynamical core plus one parameterization or parameterizations group at a time, e.g., deep convection, shallow convection, large-scale condensation, or the stratiform cloud microphysics, as an attempt to find out which of those parameterizations led to the convergence rate of 0.4 instead of 1 in the full model. Additional simulations were conducted using the dynamical core plus a simple saturation adjustment scheme or with the cloud microphysics parameterization of CAM5 but with the formation and sedimentation of rain and snow turned off (cf. Fig. 3 in Wan et al., 2015). Those simulations revealed different convergence rates and time step sensitivities associated with different components of the model code. We expect that this strategy of breaking down the code into small exercisable units could be used to pinpoint bugs when, e.g., a code refactoring effort leads to solution differences that are unexpectedly large according to the TSC test. In other words, we expect the TSC method to fulfill feature 5 listed in Sect. 2.2. Future work is planned to evaluate TSC's utility for that purpose.

2.4 ~~Simulation length~~

~~The 1 h simulation length used by Wan et al. (2015) and in Fig. 2 allowed the CAM5 model to integrate for 2 time steps when the default step size of 1800 s was used. For the TSC test which uses 1 s and 2 s step sizes, it can be beneficial to further reduce the simulation length and hence the computational cost. Results and further discussions are presented in Sect. 4.~~

~~More generally, it is worth pointing out a major distinction between the test strategies of TSC/PerGro and that of CAM-ECT. As stated earlier, for a climate model like CAM, the purpose of the verification discussed in this paper is to determine whether a loss of BFB reproducibility is accompanied by changes in the simulated climate characteristics. CAM-ECT addresses the~~

verification question in a direct way by conducting climate simulations and comparing statistical distributions of annual averages. In contrast, PerGro and TSC view CAM as a deterministic model; one-to-one solution comparisons are conducted using instantaneous gridpoint values, and the solution differences are evaluated well within the deterministic limit of the flow evolution. A key assumption behind PerGro and TSC is that, since climate is essentially the statistical characterization of deterministic-scale atmospheric conditions, and the same set of differential-integral equations control the short-term and long-term behaviors of the atmospheric motion in a numerical model, climate-changing solution differences should be detectable at very early stages of the model integration. Past experiences with PerGro in older versions of the CAM model as well as the results shown in Sect. 4 provide evidences that support this assumption.

For the purpose of evaluating the effectiveness of a method like PerGro or TSC that indirectly addresses the "has the model climate changed" question, it is necessary to use various test cases to determine whether (1) the indirect method gives a "fail" signal when certain code modifications or computing environment changes are deemed climate-changing according to the SIEVE procedure defined earlier in Sect. 2.2, and (2) whether any solution differences that trigger a "fail" signal in the indirect method are indeed climate-changing, again according to SIEVE. Ideally the role of an expert should be fulfilled by objective means, and the CAM-ECT was designed for that purpose; but the current CAM-ECT is limited in its sensitivity due to the use of global and annual mean values in constructing the test metric. In Sect. 4 we compare results from the new TSC test with those from CAM-ECT using the "correct" answers provided by the modeler developers using the subjective method.

Wan et al. (2015) reported that within the step size range of 1 s to 1800 s, the time step convergence in CAM5.3 is slow (the rate is about 0.4) and the integration errors are relatively large. In other words, in the few-second time step range, the solutions are converging but have not yet converged. For this reason, we speculate that passing the TSC test does not necessarily guarantee that the model will produce the same climate characteristics in multi-year simulations, while failing the TSC test very likely means that the model climate will be different. In other words, passing the TSC test should be considered a necessary condition for a code modification to be non-climate-changing. So far we have not seen examples of false negative in the TSC test results, but future studies are planned to extend the evaluation.

3 Implementation

In this section we first give a brief overview of the CAM5 model ~~in~~ (Sect. 3.1), emphasizing only on the aspects that are directly relevant for the technical implementation of the TSC test. The test procedure is then described in detail in Sect. 3.2

3.1 CAM5.3 overview

The global climate model used in this paper is CAM5.3 (Neale et al., 2012) with the spectral element dynamical core (Taylor and Fournier, 2010; Dennis et al., 2012). The dynamical core solves a hydrostatic version of the fluid dynamics equation, with surface pressure (PS), temperature (T), and horizontal winds (U, V) being the prognostic variables. In addition, the model includes budget equations for specific humidity (Q), as well as the mass and number concentrations of the stratiform cloud droplets (CLDLIQ, NUMLIQ) and ice crystals (CLDICE, NUMICE). The time evolution and spatial distribution of water

vapor and hydrometeors are affected by resolved-scale transport and by subgrid-scale moist processes such as turbulence, convection, and cloud microphysics. Those subgrid-scale processes provide feedback to the thermodynamical state of the atmosphere through latent heat release. CAM5.3 also has a Modal Aerosol Module (MAM, Liu et al., 2012; Ghan et al., 2012) that represents the life cycle of 6 aerosol species: sulfate, black carbon, primary organic aerosols, secondary organic aerosols, sea salt, and mineral dust. The size distribution of the aerosol population is mathematically approximated by a few log-normal modes. In this study we used the 3-mode version of MAM, thus the model's prognostic variable set also includes the particle number concentrations of the 3 modes (num_a1, num_a2, and num_a3, for the accumulation mode, Aitken mode, and coarse mode, respectively), and the mass concentrations of each aerosol species in each mode.

In the present paper we use the FC5 component set of the model, meaning that the model is configured to run with interactive atmosphere and land, prescribed climatological sea surface temperature and sea ice cover, and with the anthropogenic aerosol and precursor emissions specified using values representative of the year 2000.

3.2 Test procedure

The basic idea of the TSC ~~code verification~~ test is to perform control and test simulations with a 2 s time step, calculate their RMSDs with respect to reference simulations conducted with the control model with a 1 s time step, then determine whether the RMSDs of the control and test simulations are substantially different.

For a generic prognostic variable ψ, we define

$$\mathrm{RMSD}(\psi) = \left\{ \frac{\sum_i \sum_k w_i \left[\Delta\psi(i,k)\right]^2 \Delta\bar{p}(i,k)}{\sum_i \sum_k w_i \, \Delta\bar{p}(i,k)} \right\}^{1/2}, \tag{1}$$

$$\Delta\psi(i,k) = \psi(i,k) - \psi_r(i,k)\,, \tag{2}$$

$$\Delta\bar{p}(i,k) = \left[\Delta p(i,k) + \Delta p_r(i,k)\right]/2\,. \tag{3}$$

Here $\Delta p(i,k)$ denotes the pressure layer thickness at vertical level k and cell i, and w_i is the area of cell i. Subscript r indicates the reference solution. This formulation of RMSD follows the work of Rosinski and Williamson (1997).

~~Since the simulations are short (on the order of minutes to an hour, cf. Sect. 4), certain changes in the model, e.g. those related to dust emission or convection over land , might have limited impact on the global circulation; therefore we divide the globe into $N_{\mathrm{dom}} = 2$ domains in the analysis.~~ Time step size affects the numerical solution at every time step and every grid point, while certain atmospheric processes might occur in isolated regions thus impacting only a limited number of grid points during very short simulations. Consequently, subtle but systematic solution changes can be masked by the model's time stepping error and can be difficult to detect. To help address this challenge, we calculate RMSDs for $N_{\mathrm{dom}} = 2$ domains, i.e., land and ocean, separately. This is a practical and somewhat arbitrary choice aiming at increasing the sensitivity of the TSC test.

As for the physical quantities, the results shown in the present paper include ~~RMSDs for~~ RMSD of $N_{\mathrm{var}} = 10$ prognostic variables: V, T, Q, CLDLIQ, CLDICE, NUMLIQ, NUMICE, num_a1, num_a2, and num_a3 (i.e. the meridional wind field, temperature, specific humidity, ~~gridbox~~ grid-box mean mass and number concentrations of the stratiform cloud droplets and

ice crystals, and the particle number concentrations of the three log-normal modes that describe the aerosol size distribution, respectively). This selection of prognostic variables is motivated by an emphasis on atmospheric circulation, thermodynamics, clouds, and aerosols. The mass concentrations of aerosol species are not included, because it is unlikely that a perturbation will change the aerosol mass concentrations without affecting the number concentrations after multiple steps of integration. ~~But we note that the test analysis is easily extendable if a model developer or~~ Additional variables of type I defined in Sect. 2.1 can be added to the list, and a longer variable list might help increase the sensitivity of the test. Type-II variables can also be added if the user wishes to ~~monitor more fields.~~ cover the respective code pieces. The TSC method is flexible in this regard, although we emphasize again that only prognostic variables of type I and type II can be included in the list. The concept of time step convergence does not apply to variables that are not calculated using an evolution equation.

The test procedure includes three steps~~as described below~~. Steps 1 and 2 are needed every time a new baseline model with ~~modified climate~~ different solution characteristics is established. Between such baseline releases, only step 3 is needed for the testing of a new code version or computing environment.

Step 1: Create an M-member simulation ensemble with a control version of the model in a trusted computing environment, using 1 s time step for a simulation length of X minutes. These are considered the *reference solutions*. The independent members are initialized on January 1, 00Z using model states sampled from different months of a previously performed climate simulation, with non-zero concentrations for water vapor, hydrometeors, aerosols, and all other tracers that the model carries. ~~At the end of the X-min simulations, save the~~ Save the 3D instantaneous values of the N_{var} prognostic variables listed above, plus the values of surface pressure and land fraction, all in double precision, after a model time of t.

Step 2: Obtain an M-member ensemble using the same initial conditions as in step 1, again with the control model in a trusted computing environment, but using a 2-s time step. Compute the RMSD using Eq. (1) for each pair of simulations that started from the same initial conditions. The resulting ~~set of $N_{\mathrm{var}} \times N_{\mathrm{dom}} = 20$ RMSDs~~ RMSDs at time t are denoted as $\mathrm{RMSD}_{\sout{\mathrm{trusted}}\mathrm{trusted},t}$.

Step 3: Repeat Step 2 with a modified code or in a different computing environment. Compute the RMSDs with respect to the reference solutions created in Step 1, and denote the results ~~as $\mathrm{RMSD}_{\mathrm{test}}$~~ at model time t as $\mathrm{RMSD}_{\mathrm{test},t}$. Now define

$$\Delta\mathrm{RMSD}_{j,m\,t,j,m} = \mathrm{RMSD}_{\mathrm{test},j,m\,\mathrm{test},t,j,m} - \mathrm{RMSD}_{\mathrm{trusted},j,m\,\mathrm{trusted},t,j,m} \quad (m = 1, \cdots, M;\ j = 1, \cdots, N_{\mathrm{var}} \times N_{\mathrm{dom}}), \quad (4)$$

and denote the M-member ~~average by $\overline{\Delta\mathrm{RMSD}_j}$~~ ensemble mean by $\overline{\Delta\mathrm{RMSD}_{t,j}}$. For each prognostic variable and domain (i.e. each j), we assume the ensemble mean of ~~$\Delta\mathrm{RMSD}_j$~~ $\Delta\mathrm{RMSD}_{t,j,m}$ is a random variable ~~μ_j~~ $\mu_{t,j}$. The students t-test is performed ~~to accept or reject~~ on the null hypothesis that ~~μ_j~~ $\mu_{j,t}$ is statistically zero~~. The alternative hypothesis is $\mu_j > 0$. The null hypothesis is rejected, i.e. the~~ , with the alternative hypothesis of $\mu_{j,t} > 0$. One-sided test is used here because of the concept of self-convergence explained in Sect. 2.3: When bugs are introduced, or when the code is not compiled or executed correctly, the simulation will not solve the originally intended equations, thus not converging to the reference solutions produced by the original code or environment, resulting in larger RMSDs.

The jth variable at time t fails the TSC test ~~, if~~ if the null hypothesis is rejected, i.e., if

$$\mathcal{P}\left(\mu_{j\,t,j} > \overline{\Delta\mathrm{RMSD}}_{j\,t,j}\right) < \mathcal{P}_0\,, \quad (5)$$

where $\mathcal{P}$ stands for probability and $\mathcal{P}_0$ is an empirically chosen threshold. If Eq. (5) ~~is fulfilled~~ turns out to be true for any j, or in other words,

$$\mathcal{P}_{\mathrm{min}\,\mathrm{min},t} = \min_{j=1,N_{\mathrm{var}}\times N_{\mathrm{dom}}} \left[\mathcal{P}\left(\mu_{j\,t,j} > \overline{\Delta\mathrm{RMSD}}_{j\,t,j} \right) \right] < \mathcal{P}_0\,, \tag{6}$$

then the ensemble fails the TSC test at time t. ~~, and the code or software/hardware change is considered climate-changing.~~

In case the test and control simulations only contain insignificant differences, $\mathcal{P}_{\mathrm{min},t}$ is expected to be relatively large during the X minutes of integration, but can still get small values by chance, thus appearing like a random variable. In case a bug software/hardware issue causes substantial solution differences, it is expected that $\mathcal{P}_{\mathrm{min},t}$ will show very small values after a certain time of spin-up. We use this distinction to determine an overall pass or fail for a test ensemble. In order to fully automate the test procedure, a quantitative criterion is needed to describe this distinction. For simplicity and as a preliminary choice, we propose to fail a test ensemble if $\mathcal{P}_{\mathrm{min},t} < \mathcal{P}_0$ for all output steps in a time window $[X_0, X]$, where X is the total simulation length and X_0 is the spin-up time. The use of multiple time steps in the overall pass/fail criterion reflects our perspective of viewing the model integration as a time evolution problem. We note that the typical values of $\mathcal{P}_{\mathrm{min},t}$ depend on the number of monitored variables (i.e., larger $N_{\mathrm{var}} \times N_{\mathrm{dom}}$ can result in smaller $\mathcal{P}_{\mathrm{min},t}$ in a statistical sense), hence $\mathcal{P}_0$ needs to be determined empirically for a given $N_{\mathrm{var}} \times N_{\mathrm{dom}}$. Ideally $\mathcal{P}_0$ should be small enough to reduce the chance of false positive (i.e., insignificant solution differences being assigned a "fail"), and large enough to reduce the chance of false negative (i.e., subtle but systematic solution differences being assigned a "pass"). In the present paper we have made an empirical and somewhat arbitrary choice of

$$\mathcal{P}_0 = 0.5\%, \quad X_0 = 5\ \mathrm{min}, \quad X = 10\ \mathrm{min}\,. \tag{7}$$

Further evaluation of this choice and possible improvement of the overall pass/fail criterion are topics of future work. In the next section, we present results from 30 min simulations with the test diagnostics calculated every minute to reveal the time evolution of $\mathcal{P}_{\mathrm{min},t}$.

~~$M = 12$ ensemblemembers are included in the TSC test version 1.0 which we evaluate in the next section.~~ $M = 12$ ensemble members are used in this study. One set of initial conditions is sampled from each month of the year to obtain a reasonable coverage of the seasonal variations in the atmospheric circulation, clouds, and aerosol life cycle. The purpose is to account for possible flow-dependencies of the numerical error. The need for an ensemble is demonstrated in Fig. 3 where the normalized ΔRMSD of selected variables is shown for individual ensemble members after 5 min of integration in an experiment with a modified parameter in the deep convection parameterization over land ("CONV-LND", following Baker et al., 2015; cf. Table 1 and Sect. 4.2 for further details). Passing and failing variables are indicated by dashed and solid lines, respectively. Ocean and land are shown in separate panels using different scales for the y-axes. The values of ~~$\Delta\mathrm{RMSD}_{j,m}$~~ $\Delta\mathrm{RMSD}_{t,j,m}$ have been normalized by the mean RMSD of the trusted ensemble, i.e., by ~~$\overline{\mathrm{RMSD}}_{\mathrm{trusted},j}$~~ $\overline{\mathrm{RMSD}}_{\mathrm{trusted},t,j}$. Our exploration has indicated that, due to the complexity and nonlinearity of the model equations, the values of ΔRMSD of a passing variable from individual ensemble members often are distributed around zero (Fig. 3a). Therefore a single positive ~~$\Delta\mathrm{RMSD}_{j,m}$~~ $\Delta\mathrm{RMSD}_{t,j,m}$ cannot be viewed as sufficient evidence of non-convergence towards the reference solution. The magnitude of a positive ~~$\Delta\mathrm{RMSD}_{j,m}$~~

$\Delta\mathrm{RMSD}_{t,j,m}$ is not a good indicator, either, as Fig. 3b shows that even after normalization, a failing variable (e.g. NUMICE in Fig. 3b) can still have small albeit consistently positive ΔRMSD, while a passing variable (e.g. Q in Fig. 3b) may occasionally show large deviations from zero. We have not yet explored the dependence of the test results on the ensemble size, but plan to do so in the future. ~~The cut-off probability $\mathcal{P}_0$ determines the false positive rate of the TSC test. Our exploration showed that it was not uncommon to get $\mathcal{P}_{\min}$ below 1 from non-climate-changing solutions (cf. Fig. 4 in Sect. 4) . Therefore a rather conservative threshold of 0.05 % is used in this paper which corresponds to a t-statistic of 4.437 for 12-member ensembles. In the future, it might be useful to further evaluate this choice.~~ Furthermore, while we currently apply a t-test to determine whether the ensemble mean ΔRMSD is equal to or larger than zero, more advanced methods might help to better characterize the ensemble distribution. ~~As for the integration length, Fig. 2 provides a clear hint that an hour is sufficient for simulations with 2 s time stepto diverge. In the next section, we present results from 30 simulationswith the test diagnostics calculated every minute to reveal the initial evolution of ΔRMSD.~~

For all the simulations presented in this paper, the initial conditions were sampled from the first year (after 6 months of spin-up) of a previously conducted 5-year simulation. The decision of using the first year was arbitrary. In our experience, climate simulations of 1–5 years are frequently carried out during model development or evaluation, making such initial conditions easy to obtain. The two features we had in mind when choosing the initial conditions were that (i) they contain reasonably spun-up values for the model state variables (e.g., not all zeros or spatially constant values for the hydrometeors or aerosol concentrations), and (ii) they represent synoptic weather patterns in different seasons. The initial conditions do not need to represent well-balanced states in the quasi-equilibrium phase of a multi-year climate simulation. In fact, the default model time step of 1800 s was used when creating the initial conditions for this study, while the control and test simulations in TSC used a 1 s or 2 s time step, so the model state was certainly not well-balanced during those TSC simulations. Also notice that while model states from different seasons were used for initialization, all ensemble members started on January 1, 00Z for simplicity of the simulation and postprocessing workflow, which also led to initial imbalances. Such imbalances are considered harmless since the purpose of the numerical integration is regression testing rather than faithfully simulating the atmospheric motions in the real world. We expect that the same set of initial conditions can be used after answer-changing code baselines are established – until a point when the list of prognostic variables in the model becomes substantially different. Then it would be useful to regenerate the initial conditions, and rethink which variables should be included in the test diagnostics.

4 ~~Evaluation of the new method~~Numerical experiments

~~We challenged the TSC test with~~ Numerical simulations were carried out under a number of scenarios ~~to verify whether it issued the expected pass/fail signal~~(test cases) to help characterize $\mathcal{P}_{\min,t}$ and evaluate the TSC method. A reference ensemble with a 1 s time step and a trusted ensemble with a 2 s time step were obtained on the supercomputer Titan at the Oak Ridge Leadership Computing Facility using the Intel compiler version 15.0.2 with optimization level -O2. Various test simulations were then conducted ~~under~~ in three groups (Table 1)~~:~~. Our strategy here is to repeat representative test cases from Baker et al. (2015) and

Milroy et al. (2016), and expect the TSC method to give the same "pass" or "fail" results as CAM-ECT did, with a few exceptions explained below.

~~Group E ("computing Environment") simulations~~ Group ENV used the same code ~~but~~ as in the reference ensemble but with different computers, ~~compiler versions~~compilers, or optimization levels~~. Four configurations in this group had been previously verified by the SIEVE procedure as non-climate-changing~~:

- PGI compiler version 15.3.0 with -O2 on Titan ("Titan-PGI");
- ~~Intel compiler version 15.0.1 with -O2 on the Linux cluster Constance at the Pacific Northwest National Laboratory's Institutional Computing ("Constance-Intel");~~
- ~~Intel compiler version 16.0.0 with -O2 on Cori at the National Energy Research Scientific Computing Center ("Cori-Intel");~~
- Intel compiler version 15.0.0 with -O2 on Yellowstone (ark:/85065/d7wd3xhc) at the Computational and Information Systems Laboratory of the National Center for Atmospheric Research ("YS-Intel15-O2");
- Intel compiler version 15.0.0 with -O3 on Yellowstone ("YS-Intel15-O3").

~~The fifth case ("YS-Iintel15-O3")used a higher optimization level on Yellowstone, and had~~ Titan-PGI and YS-Intel15-O2 are supported environments for CAM5.3, in which the simulations are expected to pass the TSC test. The YS-Intel15-O3 case has been found by Baker et al. (2015) to produce incorrect answers, and is expected to fail TSC. (We note that such incorrect answers are produced only when the model is compiled without the "-fp-model" flag. ~~If~~ In contrast, if the "-fp-model source" flag is applied to the Fortran code~~and~~ , and the "-fp-model precise" is applied to the C code, the -O2 and -O3 optimization options will produce BFB identical results when CAM5.3 is compiled on Yellowstone with Intel 15.0.0.) We do not include here results from computers that produced borderline pass/fail results in CAM-ECT (e.g., Mira at the Argonne National Laboratory and Bluewaters at the University of Illinois). Valuable investigations have been made by Milroy et al. (2016), but those cases still need further investigation and characterization.

Group MOD consists of two code modification cases from Milroy et al. (2016) that were motivated by optimization of the computational performance:

- In the Division-to-multiplication ("DM") case, division by a time-invariant array was replace by multiplication of the inverse at one place in the dynamical core (cf. Sect. 3.2 in Milroy et al., 2016). This case has been found by CAM-ECT to produce a model climate that is statistically consistent with the reference ensemble. We expect the TSC test to produce a "pass" result;
- In the Precision ("P") case, a subroutine in the physics suite for calculating the saturation vapor pressure over water using the Goff-Gratch formula was changed from double-precision to single-precision. This modification has also been found by CAM-ECT to produce consistent climate, but we put "unknown" in Table 1 for the expected outcome of TSC due to the deterministic nature of the TSC method and the use of double-precision output in the calculation of the test diagnostics.

~~In group P1 ("Parameter perturbation set 1")~~ In group PAR, we repeated all the parameter perturbation experiments presented by Baker et al. (2015) ~~(cf. Section 4.3 therein). One~~ where one parameter in CAM5's physics package was modified ~~in each experiment,~~ at a time. ~~and the perturbations were expected to cause physically significant changes in the simulated climate.~~ According to Baker et al. (2015), this list of parameters were provided by climate scientists; the parameter changes were thought to affect the model climate in a non-trivial manner, and were intended to be used in different model configurations (e.g. with different resolutions, cf. Sect. 4.3 in Baker et al., 2015). All but one cases failed CAM-ECT, the exception being the NU case in which the numerical diffusion in the dynamical core was changed by about 10%. Baker et al. (2015) pointed out that CAM-ECT gave an unexpected but understandable "pass" flag in this case, because CAM-ECT monitored the global mean values that were not directly affected by the numerical horizontal diffusion. We expect the TSC test to assign "fail" to all cases in this group, including NU, since TSC compares the instantaneous grid-point values of the prognostic variables, thus is expected to be capable of detecting solution changes at all spatial scales resolved by the model. ~~Group P2 ("Parameter perturbation set 2") includes two additional scenarios that were similar to RH-MIN-LOW in group P1 but with smaller perturbations: the values of 0.89 and 0.897 for cldfrc_rhminl correspond to relative changes of 0.8 % and 0.06 %, respectively, compared to the default value of 0.8975. Furthermore, we tested the QSMALL configuration in which the smallest non-zero condensate concentration in the stratiform cloud microphysics parameterization was increased from 10^{-18} kg kg^{-1} to 10^{-8} kg kg^{-1}. It has been found that this increase of concentration threshold can help avoid undesirably rapid growth of initial perturbations, but produces a climate change detectable by SIEVE.~~ All simulations in groups ~~P1 and P2~~ MOD and PAR were conducted on Titan using the default Intel compiler version and optimization level (15.0.2-O2). ~~In the following, we will refer to each row in Table 1 as a "case".~~

4.1 ~~Results at 5 min~~Evolution of $\mathcal{P}_{\min,t}$

To understand the initial evolution of $\mathcal{P}_{\min,t}$, we conducted 30 min simulations and calculated the test diagnostics after every minute. Fig. 4 shows the time series of $\mathcal{P}_{\min,t}$ using a linear scale in panel (a) and a logarithmic scale in panel (b). Two distinct types of behavior can be seen in the figure. In test scenarios where solution differences were thought to be insignificant, $\mathcal{P}_{\min,t}$ resembles random perturbations around mean values of a few percent. The value at a particular time instance can fall below 1 %, but returns to larger values at later time steps (Fig. 4a). In all test scenarios with modified model parameters, the values of $\mathcal{P}_{\min,t}$ are distinctly closer to zero (Fig. 4a). The time series either show a clear decrease in the first 10 min and considerably slower changes afterwards (e.g., CONV-LND and NU in Fig. 4b), or start with very low probabilities already and show relatively small changes during the integration (e.g., DUST and FACTIC in Fig. 4b).

The dashed gray lines in Fig. 4 indicate the threshold we chose for assigning an overall "pass" or "fail" to a test ensemble (Eq. 7). The test scenarios that were expected to produce insignificant (significant) solution differences indeed pass (fail) the TSC test. The Precision ("P") case of unknown outcome also passes the TSC test, giving a result consistent with that from CAM-ECT. The two rightmost columns of Table 1 show the values of $\mathcal{P}_{\min,t}$ at $t = 5$ min or averaged between 5 min and 10 min. Both the instantaneous and averged probabilities are orders of magnitude smaller in the failing cases than in the passing cases.

4.2 Results at 5 min

~~Summaries of the test results after 5 min of model integration are presented in Table 1 and in Fig. ??. According to the criterion that $\mu_j > 0$ for any j results in an overall fail, all the simulations with software/hardware change, except the YS-Intel15-O3 case, passed the TSC test, while all the simulations with modified parameters failed the test. The outcome agrees with our original expectation. The results shown in Table 1 and Fig. ??b indicate that $\mathcal{P}_{\min}$ ranges between 0.6 % and 15 % in the passing cases. In contrast, the probabilities that the trusted and test simulations are behaving similarly are substantially smaller in the failing cases, ranging between 10^{-16} % and 0.011 %.~~

We now take a closer look at the test diagnostics at a single time instance. In Fig. 5, the statistical distributions of ~~μ_j~~ $\mu_{t,j}$ (the mean ΔRMSD) estimated from the 12-member ensembles are shown at $t = 5$ min for the individual prognostic variables and domains for four test cases. The values are normalized using the corresponding mean RMSD of the trusted ensemble, i.e., ~~$\overline{\mathrm{RMSD}}_{\mathrm{trusted},j}$~~ $\overline{\mathrm{RMSD}}_{\mathrm{trusted},t,j}$. The dots indicate the observed ensemble mean (i.e. ~~$\overline{\Delta\mathrm{RMSD}}_j$~~ $\overline{\Delta\mathrm{RMSD}}_{t,j}$), and the filled boxes indicate the $\pm 2\sigma$ range of the mean. The left end of an unfilled box shows the threshold value corresponding to ~~$\mathcal{P}_0 = 0.05$~~ $\mathcal{P}_0 = 0.5$ % in the one-sided t-test. Red and blue indicate fail and pass, respectively, according to the criterion defined by Eq. (5). Notice that the x-axes in the subpanels of Fig. 5 are shown in different scales. The normalized mean RMSD differences between the ~~Cori~~ P ensemble and the trusted ensemble are ~~very~~ small, on the order of ~~10^{-4}~~ 0.1 or smaller, and the value of 0 lies within the $\pm 2\sigma$ range of the observed ~~$\overline{\Delta\mathrm{RMSD}}_j$ for most of the~~ $\overline{\Delta\mathrm{RMSD}}_{t,j}$ for all the $N_{\mathrm{var}} \times N_{\mathrm{dom}}$ variables (Fig. 5a). In contrast, the YS-Intel15-O3 case (which is known to produce incorrect solutions) is associated with typical RMSD differences ~~of order 10^0, and 14 out of the~~ around 1. The large number of failing variables (16 out of 20~~variables failed the TSC test with a $\mathcal{P}_{\min}$ of 7~~) and the very small $\mathcal{P}_{\min,t}$ (1×10^{-14}–$^{-11}$ %~~, indicating~~) indicate a clearly failing case.

The test case with a modified dust emission factor (DUST) was expected to be challenging for the TSC method. In any model day, the emission only occurs at a very small fraction of the dust source areas. Dust particles emitted from the surface can only be transported over a short distance during the few-minute simulation time, and the impact on meteorological conditions through the absorption and/or scattering of radiation is also limited. Hence it is unlikely that the solution differences can be seen in the global temperature RMSD. This was the reason that motivated us to use multiple prognostic variables and to separate land and ocean when defining the test diagnostics. The results shown in Fig. 5c confirm our expectation, as only 1 out of the ~~20 $\overline{\Delta\mathrm{RMSD}}_j$ values~~ twenty $\overline{\Delta\mathrm{RMSD}}_{j,t}$ is significantly larger than zero. The DUST experiment should nevertheless be considered a clearly failing case since the failing variable (num_a3 over land) is indeed the physical quantity that is most directly affected by dust emission, and the large ~~$\overline{\Delta\mathrm{RMSD}}_j$~~ $\overline{\Delta\mathrm{RMSD}}_{j,t}$ corresponds to a very small ~~$\mathcal{P}\left(\mu_j > \overline{\Delta\mathrm{RMSD}}_j\right)$ of 0.0015~~$\mathcal{P}\left(\mu_{j,t} > \overline{\Delta\mathrm{RMSD}}_{j,t}\right)$ of 0.0019 % (cf. Table 1).

The CONV-LND case is challenging for similar reasons. Here the coefficient that controls the conversion of cloud condensate to precipitation was modified for deep convection over land. With a smaller value for zmconv_c0_lnd, we expect to have more cloud condensate detrained by deep convection, which can lead to changes in the mass and number concentrations of ice crystals in stratiform clouds. Failing results are indeed seen in these two variables (Fig. 5d)~~with a $\mathcal{P}_{\min}$ of 0.0026~~. The anomalous result in num_a2 is likely related to the removal of aerosol particles by convective precipitation. Since deep convection over

land happens in limited areas, and the natural variability is very strong, it is not surprising that ~~$\overline{\Delta\mathrm{RMSD}_j}$~~ $\overline{\Delta\mathrm{RMSD}_{j,t}}$ of the other variables are not yet significantly larger than zero after 5 min of integration.

~~Another test case worth noting is the NU configuration in which the numerical diffusion in the dynamical core was changed by about 10 %, and the resulting model climate was expected to be different. Baker et al. (2015) pointed out that CAM-ECT gave an unexpected but understandable "pass" flag in this case, because CAM-ECT monitored the global mean values that were not directly affected by the numerical horizontal diffusion. Our TSC test compares the instantaneous grid-point values of the prognostic variables, thus can detect solution changes at all spatial scales resolved by the model. In the NU test case, we saw 5 failing variables after 5 min (not shown) with a very low $\mathcal{P}_{\min}$ of 8.4×10^{-6} %.~~

As mentioned earlier, CAM-ECT assigned a "pass" to the NU case but we expect the TSC result to be a "fail". The respective time series in Fig. 4b reveals $\mathcal{P}_{\min,t}$ values below 10^{-4} % after 3 min of integration. At 5 min, there are a total of 6 variables with $\mathcal{P}_{t,j} < 0.5\,\%$; the four variables with lowest probabilities are ocean-mean meridional wind, land-mean meridional wind, ocean-mean temperature, and ocean-mean specific humidity. The small minimum probability and the combination of the failing variables provide confidence in the "fail" result of the NU case.

4.3 ~~30 simulations~~

~~To understand the initial evolution of $\Delta\mathrm{RMSD}$, we conducted 30 min simulations and calculated the test diagnostics after every minute. Fig. 4 shows the time series of $\mathcal{P}_{\min}$ using a linear scale in panel (a) and a logarithmic scale in panel (b). $\mathcal{P}_{\min}$ in the passing cases resembles random perturbations around mean values of a few percent; the value at any time instance can fall below 1 % or exceed 20 % (Fig. 4a). Values of $\mathcal{P}_{\min}$ in the failing cases are distinctly closer to zero (Fig. 4a), often showing a clear decrease in the first 15 min and considerably slower changes afterwards (Fig. 4b). Since the fastest changes of $\mathcal{P}_{\min}$ typically occur in the first few minutes, we chose 5 min as the simulation length for the version 1.0 implementation of the TSC test.~~

~~The rightmost columns of Table 1 show that the test diagnostics calculated after 30 min generally feature smaller $\mathcal{P}_{\min}$; further review of the results also indicated a typical increase in the number of failing variables when the integration time is increased. However, the overall passes and fails turn out to be the same at 5 min and at 30 min. If we had chosen a simulation length of 3 min or shorter and still used 0.05 % for the cut-off probability $\mathcal{P}_0$, the CONV-LND case would have passed the TSC test. To avoid such a false negative, it might be possible to increase $\mathcal{P}_0$ but require in addition that $\mathcal{P}_{\min}$ show a clear trend of decrease since the beginning of the simulations. We did not carry out further exploration in that direction because 5 min simulations (150 time steps) are already inexpensive to carry out (see below).~~

4.3 Computational cost

Based on the results shown above, we propose a version 1.0 implementation of the TSC test that uses 12-member ~~5~~10 min simulations. As such, the computational cost of obtaining an ensemble of reference solutions (using 1 s time step) plus an ensemble of trusted solutions (using 2 s time step) is similar to conducting a single ~~4-month~~ 7.5-month simulation using the default model time step (30 min). For the testing of a new code or computing environment, the cost of conducting 12

simulations using a 2 s time step is similar to that of a ~~40-day simulations~~ 75-day simulation performed using the default time step. Compared to the CAM-ECT which ~~includes~~ uses 151 to 453 one-year simulations in the reference ensemble and 3 one-year simulations in the test ensemble, the TSC test is a factor of ~~450~~ several hundred cheaper to obtain the reference simulations, and a factor of ~~30~~ 15 cheaper to test a new code or environment.

The TSC method also allows for very fast test turnaround since the ensemble simulations can be conducted in parallel. On Titan we used 512 MPI processes for each simulation and often submitted 12 simulations to the Portable Batch System (PBS) in three 128-node batch jobs. The wall clock time for finishing a single 10 min simulation with 2 s time step was about ~~5~~10 min; the entire set of 12 simulations was ~~typically completed in 10~~often completed in 30 min ~~to 20 after submission, and the~~ after submission. The time between first job submission and last job completion rarely exceeded a few hours.

5 Discussion

In this paper we have presented evidence to demonstrate that the concept of time step convergence can be used to assess the magnitude of solution difference in the CAM model. Future work will be useful to explore the following topics:

5.1 Test setup

The TSC test procedure described in this paper has multiple parameters that can be modified: (1) ensemble size, (2) initialization strategy (e.g., simulation start time), (3) time step sizes, (4) integration length, (5) prognostic variables and model sub-domains included in the calculation of test diagnostics, and (6) the pass/fail criterion. Results presented in the previous section indicate that given (1)-(3), the choices for (4)-(6) can have strong impacts on the outcome of the TSC test.

In the DUST case, for example, systematically positive ΔRMSD was detected only in one prognostic variable and only over land (cf. Fig. 5c for results at $t = 5\,\mathrm{min}$; results at later time are similar thus not shown). If we had not included aerosol concentrations in the list of monitored variables, or had not chosen to calculate the test diagnostics over land and ocean separately, the TSC test would have given a false "pass" (i.e., a false negative result). While the limited number of test scenarios included in this study have been categorized as expected by the current test setup, there might be more subtle cases, e.g., minor bugs in the code, that require further adjustment of aspects (4)-(6). As a next step, we plan to include a number of bug fixes and additional parameter modifications from the recent model development activities to further evaluate the TSC test setup.

Results in Fig. 4 revealed that $\mathcal{P}_{\mathrm{min},t}$ in passing and failing cases evolve differently. Considering the inherent nonlinearities in the model equations and the resulting variability in the numerical solutions, a pass/fail criterion that characterizes the time series of $\mathcal{P}_{\mathrm{min},t}$ using multiple time steps is expected to provide more accurate test results than a criterion based on one time step. In this paper we made a simple and preliminary choice, requiring all $\mathcal{P}_{\mathrm{min},t}$ diagnosed between $t = 5\,\mathrm{min}$ and $t = 10\,\mathrm{min}$ to fall below a threshold of 0.5 ~~.~~% in order for a case to fail the test. Adopting a more refined criterion, e.g., one that takes into account not only the magnitude of $\mathcal{P}_{\mathrm{min},t}$ but also its trend, might allow us to further shorten the integration time. The impacts of ensemble size and initialization strategy were not explored in this study, but are worth investigating in future work.

5.2 Impact of noisy parameterization

As mentioned in the introduction, the radiation parameterization in CAM5 uses a random number generator that leads to state-dependent noise in the model results. All the simulations presented in Sect. 4 were conducted with a fixed time step size ratio between radiation and the other physics parameterizations, with radiation calculated every other time step. We also conducted TSC simulations with radiation calculated only at the first time step. The impact is illustrated by Fig. 6 where one failing case, CONV-LND, is shown together with two passing cases, Titan-PGI and YS-Intel15-O2. The time series of $\mathcal{P}_{\min,t}$ in the CONV-LND case is not distinguishable from the passing cases in the first 3 min of model integration when radiation was called frequently, but already distinguishable after the first minute when radiation was called only once. Substantial decrease of initial $\mathcal{P}_{\min,t}$ in the "radiation-once-only" configuration was also seen in several other test scenarios. Our interpretation of this observation is that noise in the model makes it harder to detect signal associated with parameter perturbation, thus requiring longer spin-up in the TSC test. This implies that for models that have very noisy physics, e.g., those with stochastic parameterizations, the TSC simulations might need to be longer than proposed here. Hodyss et al. (2013) demonstrated that noise in a discrete model can result in reduced convergence rate or even loss of convergence. We speculate that the TSC method can still be useful as long as the model has an appreciably positive convergence rate (recall that the time step convergence in CAM5 features a slow rate of 0.4). It will be interesting to explore the utility of our method in models with stochastic parameterizations.

5.3 Comparison with other test methods

The development of the TSC test was motivated by the loss of utility of the PERGRO method and the relatively high computational cost of CAM-ECT. Since all three are regression testing methods, it is worth clarifying some linkages and distinctions among them.

CAM-ECT compares the model climate, and considers two sets of results "the same" when ensembles of one-year simulations show consistent statistical distributions of global annual averages. PERGRO and TSC view CAM as a deterministic model, and considers two sets of model results "the same" when the observed solution differences with respected to trusted solutions appear to be consistent with the expected evolution of initial perturbation or time stepping error. In PERGRO and TSC, one-to-one solution comparisons are conducted using instantaneous grid-point values, and the solution differences are evaluated well within the deterministic limit of the flow evolution.

From the perspective that climate is essentially the statistical characterization of deterministic-scale atmospheric conditions, and the fact that the same set of differential-integral equations control the short-term and long-term behaviors of the atmospheric motion in a numerical model, one can expect the different regression testing methods to provide the same "pass" or "fail" results when the solution differences are either very small (e.g., at round-off level) or very different (e.g., due to a major bug in the code). The general consistency between the TSC results shown in this paper and the corresponding test results from Baker et al. (2015) provides evidence to support this reasoning. On the other hand, since the different methods assess the magnitude of solution change with different criteria and at different time scales, we expect there will be cases when they give

different answers. The NU case (cf. Table 1 and Sect. 4) that passed CAM-ECT but failed TSC is one such example. As a possible opposite example, we note that within the step size range of 1 s to 1800 s, the time step convergence in CAM5.3 is slow (the rate is about 0.4) and the time step sensitivity is strong (Wan et al., 2015). In other words, in the few-second time step range, the solutions are converging but have not yet converged. For this reason, we speculate that some subtle solution changes might pass the TSC but fail CAM-ECT.

For practical model testing, it is highly desirable to find methods capable of detecting early signs of climate-changing results at low computational cost and with fast test turnaround. However, it is worth noting that the word "climate-changing" is ambiguous until a quantitative criterion is specified. For example, two simulations representing indistinguishable climate characteristics according to SIEVE (cf. Sect. 1) based on the AMWG diagnostics package (https://www2.cesm.ucar.edu/working-groups/amwg/amwg-diagnostics-package) might be distinguishable using additional metrics or using CAM-ECT. Similarly, two simulations determined to be consistent using CAM-ECT based on the global and annual averages might turn out distinguishable using grid-point-wise model output and monthly time series. As for the TSC method, the relatively strong time step sensitivity in CAM5 implies that the numerical accuracies are substantially different when time step size is changed, hence a test procedure based on time step convergence also includes some level of ambiguity. As can be seen in Fig. 2, if we had chosen to conduct a TSC test using a 1800 s time step instead of 2 s, the results from the RH-MIN-HIGH case (which was determined by CAM-ECT as climate-changing) would have been assigned a "pass" by TSC. In the future, if CAM's convergence rate is improved and the accuracy of time stepping increased, one can expect TSC test conducted with 2 s step size to be capable of detecting more subtle solution differences. Since there are flexibilities in the TSC test (cf. Sect. 5.1), we expect it will be possible to adjust the test setup so that the outcome closely matches the results from CAM-ECT or other methods that compare the model climate with a clearly defined criterion for "climate-changing" results. Future work is planned to further compare TSC with other regression testing methods.

6 Conclusions

In this study, we designed and evaluated a test procedure for determining whether the solutions of a numerical model remain the same within the limit of the time integration accuracy when the bit-for-bit reproducibility is lost due to code modifications or computing environment changes. A "fail" signal is issued when the numerical solutions no longer converge to the reference solutions of the original model. The test method is deterministic by nature, but involves an ensemble of simulations to account for ~~the~~ possible flow dependencies of the numerical error.

Using the CAM5 model, we ~~demonstrated~~ provided initial evidence that the test procedure based on ~~5~~10 min simulations with 2 s step size (i.e., a total of ~~150~~ 300 time steps per simulation) can be used to distinguish situations where ~~experts' judgements based on multi-year simulations leads to the conclusion that the model results represent the same or different~~ solution differences were deemed insignificant or substantial by a different testing method based on assessment of the simulated climate statistics. The ~~test hence provides an~~ new test is not exhaustive since it does not detect issues associated with diagnostic calculations that do not feedback to the model state variables. Nevertheless it provides a practical, objective and computa-

tionally inexpensive way to assess the significance of solution changes. Our experience showed that, using supercomputing facilities, the wall clock time for conducting an ensemble of 12-member simulations ~~can be as short as~~ typically ranges from a few minutes to a few hours. Such fast turnaround makes the new test a ~~very~~ convenient tool for model testing. ~~Furthermore, the earlier work of Wan et al. (2015) has shown that with a very short integration time it is possible~~ Future studies are planned to further evaluate the new method using more test scenarios, compare it with other methods of regression testing, and optimize the implementation of the strategy. We also plan to assess the ~~time step convergence of individual parameterizations in isolation. This implies the new test procedure can be applied~~ feasibility of applying the test to subcomponents of the model code ~~thus facilitate debugging.~~

~~Because the test design uses the time stepping error associated with 2 step size as the key metric for determining a pass or fail, we speculate that, in principle, passing this test does not guarantee that the model will produce the same climate characteristics, while failing the test will very likely mean that the model climate will be different. Passing the convergence test should hence be considered as a necessary condition for a code modification to be non-climate-changing. We did not see any examples of false negative (i.e. climate-changing modifications passing the convergence test) in the test cases presented in this paper, but future studies are planned to further evaluate the method. In addition, we plan to conduct an empirical study to quantify the false positive rate (i.e., the chance of a non-climate-changing code modification passing the convergence test by coincidence) associated with different ensemble sizes, and further optimize the implementation of the methodology~~for the purpose of unit testing and debugging.

The new test is ~~based~~ built on the generic concept of time step convergence, and the implementation does not require any code modifications. We plan to explore the utility of the method in other components of our Earth system model (e.g., ocean, sea ice, and land ice), and expect that the same concept is applicable to a wide range of geophysical models such as global and regional weather and climate models, cloud resolving models, large eddy simulations, and even direct numerical simulations.

~~It is worth noting that the CAM5 model used in this study is a deterministic model. Although the radiation code uses the Monte-Carlo Independent Column Approximation (?) to represent the subgrid-scale cloud variability, the resulting randomness is avoided in our test design by fixing the radiation time step at 1 as in the default model. We have not yet evaluated any alternate test implementation that involves more frequent radiation calculation. More generally, it will be interesting to evaluate the usefulness of the new test in models with truly stochastic parameterizations. Hodyss et al. (2013) have demonstrated that random noise in a discrete model can result in reduced convergence rate or even loss of convergence. We speculate that our convergence-based test method can still be useful as long as the model has an appreciably positive convergence rate (recall that the time step convergence in CAM5 features a slow rate of 0.4), but the speculation needs to be verified by future work.~~

7 Code and data availability

The source code of CAM5 can be obtained as part of the Community Earth System Model (CESM) from the public release website https://www2.cesm.ucar.edu/models/current. The scripts for conducting and analyzing the ensemble simulations, and the simulation data discussed in the paper, are available from the corresponding author upon request.

Acknowledgements. The authors thank Dr. W. Sacks (NCAR) and the two anonymous reviewers for their valuable comments and suggestions. ~~The research described in this paper was funded by~~ This research was supported as part of the Accelerated Climate Modeling for Energy (ACME) ~~project through the Office of Biological and Environmental Research (BER) in the~~ program, funded by the U.S. Department of Energy~~(DOE)~~, Office of Science, Office of Biological and Environmental Research (BER). The basis of the work, the time step convergence study, was previously supported by ~~the DOE Office of Science~~ BER as part of the Scientific Discovery through Advanced Computing (SciDAC) Program, and by the Linus Pauling Distinguished Postdoctoral Fellowship of the Pacific Northwest National Laboratory (PNNL). This research used high-performance computing resources from the ~~following institutions: the~~ Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, supported by the Office of Science of DOE under Contract No. DE-AC05-00OR22725~~; the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of DOE under Contract No. DE-AC02-05CH11231. the~~, and the National Center for Atmospheric Research (NCAR) Computational and Information Systems Laboratory, sponsored by the National Science Foundation~~; PNNL Institutional Computing (PIC).~~, PNNL is operated by Battelle Memorial Institute for DOE under contract DE-AC05-76RL01830. NCAR is sponsored by the National Science Foundation.

References

Baker, A. H., Hammerling, D. M., Levy, M. N., Xu, H., Dennis, J. M., Eaton, B. E., Edwards, J., Hannay, C., Mickelson, S. A., Neale, R. B., Nychka, D., Shollenberger, J., Tribbia, J., Vertenstein, M., and Williamson, D.: A new ensemble-based consistency test for the Community Earth System Model (pyCECT v1.0), Geoscientific Model Development, 8, 2829–2840, doi:10.5194/gmd-8-2829-2015, http://www.geosci-model-dev.net/8/2829/2015/, 2015.

Baker, A. H., Hu, Y., Hammerling, D. M., Tseng, Y., Xu, H., Huang, X., Bryan, F. O., and Yang, G.: Evaluating Statistical Consistency in the Ocean Model Component of the Community Earth System Model (pyCECT v2.0), Geoscientific Model Development Discussions, 2016, 1–28, doi:10.5194/gmd-2016-3, http://www.geosci-model-dev-discuss.net/gmd-2016-3/, 2016.

~~Carson, J. S.: Model verification and validation, in: Simulation Conference, 2002. Proceedings of the Winter, vol. 1, pp. 52–58 vol.1, , 2002.~~

Bodas-Salcedo, A., Webb, M. J., Bony, S., Chepfer, H., Dufresne, J.-L., Klein, S. A., Zhang, Y., Marchand, R., Haynes, J. M., Pincus, R., and John, V. O.: COSP: Satellite simulation software for model assessment, Bulletin of the American Meteorological Society, 92, 1023–1043, doi:10.1175/2011BAMS2856.1, http://dx.doi.org/10.1175/2011BAMS2856.1, 2011.

Dennis, J. M., Edwards, J., Evans, K. J., Guba, O., Lauritzen, P. H., Mirin, A. A., St-Cyr, A., Taylor, M. A., and Worley, P. H.: CAM-SE: A scalable spectral element dynamical core for the Community Atmosphere Model, Int. J. High Perform. Comput. Appl., 26, 74–89, doi:10.1177/1094342011428142, 2012.

Ghan, S. J., Liu, X., Easter, R. C., Zaveri, R., Rasch, P. J., Yoon, J.-H., and Eaton, B.: Toward a Minimal Representation of Aerosols in Climate Models: Comparative Decomposition of Aerosol Direct, Semidirect, and Indirect Radiative Forcing, J. Climate, 25, 6461–6476, doi:10.1175/JCLI-D-11-00650.1, 2012.

Hodyss, D., Viner, K. C., Reinecke, A., and Hansen, J. A.: The Impact of Noisy Physics on the Stability and Accuracy of Physics–Dynamics Coupling, Monthly Weather Review, 141, 4470–4486, 2013.

Kristiansen, N. I., Stohl, A., Olivié, D. J. L., Croft, B., Søvde, O. A., Klein, H., Christoudias, T., Kunkel, D., Leadbetter, S. J., Lee, Y. H., Zhang, K., Tsigaridis, K., Bergman, T., Evangeliou, N., Wang, H., Ma, P.-L., Easter, R. C., Rasch, P. J., Liu, X., Pitari, G., Di Genova, G., Zhao, S. Y., Balkanski, Y., Bauer, S. E., Faluvegi, G. S., Kokkola, H., Martin, R. V., Pierce, J. R., Schulz, M., Shindell, D., Tost, H., and Zhang, H.: Evaluation of observed and modelled aerosol lifetimes using radioactive tracers of opportunity and an ensemble of 19 global models, Atmospheric Chemistry and Physics, 16, 3525–3561, doi:10.5194/acp-16-3525-2016, http://www.atmos-chem-phys.net/16/3525/2016/, 2016.

Liu, X., Easter, R. C., Ghan, S. J., Zaveri, R., Rasch, P., Shi, X., Lamarque, J.-F., Gettelman, A., Morrison, H., Vitt, F., Conley, A., Park, S., Neale, R., Hannay, C., Ekman, A. M. L., Hess, P., Mahowald, N., Collins, W., Iacono, M. J., Bretherton, C. S., Flanner, M. G., and Mitchell, D.: Toward a minimal representation of aerosols in climate models: description and evaluation in the Community Atmosphere Model CAM5, Geosci. Model Dev., 5, 709–739, doi:10.5194/gmd-5-709-2012, 2012.

Milroy, D. J., Baker, A. H., Hammerling, D. M., Dennis, J. M., Mickelson, S. A., and Jessup, E. R.: Towards Characterizing the Variability of Statistically Consistent Community Earth System Model Simulations, Procedia Computer Science, 80, 1589–1600, doi:10.1016/j.procs.2016.05.489, http://www.sciencedirect.com/science/article/pii/S1877050916309759, 2016.

Neale, R. B., Richter, J. H., Conley, A. J., Park, S., Gettelman, A., Williamson, D. L., Rasch, P. J., Vavrus, S. J., Taylor, M. A., Collins, W. D., Zhang, M., and Lin, S. J.: Description of the NCAR Community Atmosphere Model (CAM4.0), NCAR Technical Note NCAR/TN-485+STR, National Center for Atmospheric Research, Boulder, Colorado, USA, 2010.

Neale, R. B., Chen, C. C., Gettelman, A., Lauritzen, P. H., Park, S., Williamson, D. L., Conley, A. J., Garcia, R., Kinnison, D., Lamarque, J. F., Marsh, D., Mills, M., Smith, A. K., Tilmes, S., Vitt, F., Morrison, H., Cameron-Smith, P., Collins, W. D., Iacono, M. J., Easter, R. C., Ghan, S. J., Liu, X. H., Rasch, P. J., and Taylor, M. A.: Description of the NCAR Community Atmosphere Model (CAM5.0), NCAR Technical Note NCAR/TN-486+STR, National Center for Atmospheric Research, Boulder, Colorado, USA, 2012.

Neale, R. B., Richter, J., Park, S., Lauritzen, P. H., Vavrus, S. J., Rasch, P. J., and Zhang, M.: The Mean Climate of the Community Atmosphere Model (CAM4)in Forced SSTand Fully Coupled Experiments, J. Clim., 26, 5150–5168, , 2013.

Oberkampf, W. and Roy, C.: Verification and Validation in Scientific Computing, Cambridge University Press, , 2010.

Park, S., Bretherton, C. S., and Rasch, P. J.: Integrating Cloud Processes in the Community Atmosphere Model, Version 5., J. Clim., 27, 6821–6855, doi:10.1175/JCLI-D-14-00087.1, 2014.

Pincus, R., Barker, H. W., and Morcrette, J.-J.: A fast, flexible, approximate technique for computing radiative transfer in inhomogeneous cloud fields, Journal of Geophysical Research: Atmospheres, 108, n/a–n/a, , , 4376, 2003.

Rosinski, J. M. and Williamson, D. L.: The Accumulation of Rounding Errors and Port Validation for Global Atmospheric Models, SIAM J. Sci. Comput., 18, 552–564, doi:10.1137/S1064827594275534, 1997.

Taylor, M. A. and Fournier, A.: A compatible and conservative spectral element method on unstructured grids, J. Comput. Phys., 229, 5879–5895, doi:10.1016/j.jcp.2010.04.008, 2010.

Wan, H., Rasch, P. J., Zhang, K., Qian, Y., Yan, H., and Zhao, C.: Short ensembles: an efficient method for discerning climate-relevant sensitivities in atmospheric general circulation models, Geosci. Model Dev., 7, 1961–1977, doi:10.5194/gmd-7-1961-2014, 2014.

Wan, H., Rasch, P. J., Taylor, M. A., and Jablonowski, C.: Short-term time step convergence in a climate model, J. Adv. Model. Earth Syst., 7, 215–225, doi:10.1002/2014MS000368, 2015.

Zhang, K., Wan, H., Zhang, M., and Wang, B.: Evaluation of the atmospheric transport in a GCM using radon measurements: sensitivity to cumulus convection parameterization, Atmospheric Chemistry and Physics, 8, 2811–2832, doi:10.5194/acp-8-2811-2008, http://www.atmos-chem-phys.net/8/2811/2008/, 2008.

Table 1. CAM5 simulations conducted to evaluate the effectiveness of the TSC method. Simulations in group ~~E ("computing Environment")~~ ENV used the same code but different computers, compiler versions, or optimization levels. Group ~~P1 ("Parameter perturbation set 1")~~ MOD includes code modifications following Milroy et al. (2016). Group PAR includes parameter perturbation simulations ~~conducted~~ following ~~the design of~~ Baker et al. (2015). ~~Group P2 ("Parameter perturbation set 2") contains additional simulations that were designed to fail the TSC test.~~ The pass/fail criterion and the definition of ~~$\mathcal{P}_{\min}$~~ $\mathcal{P}_{\min,t}$ can be found in Sect. 3.2.

Group	Case name	Computer	Compiler/ optimization	Code change	Model parameters	Pass/fail expected	Pass/fail from TSC	$\mathcal{P}_{\min,t}$ 5–10 min avg.	$\mathcal{P}_{\min,t}$ at $t = 5$min
-	CTRL	Titan	Intel 15.0.2 –O2	No	All default	-	-	-	-
ENV	Titan-PGI	Titan	PGI 15.3.0 –O2	No	All default	Pass	Pass	11 %	6.4 %
ENV	YS-Intel15-O2	Yellowstone	Intel 15.0.0 –O2*	No	All default	Pass	Pass	4.5 %	3.8 %
ENV	YS-Intel15-O3	Yellowstone	Intel 15.0.0 –O3*	No	All default	Fail	Fail	3.8×10^{-12} %	1.0×10^{-11} %
MOD	DM	Titan	Intel 15.0.2 –O2	Yes	All default	Pass	Pass	8.6 %	6.2 %
MOD	P	Titan	Intel 15.0.2 –O2	Yes	All default	Unknown	Pass	7.8 %	4.2 %
PAR	DUST	Titan	Intel 15.0.2 –O2	No	dust_emis_fact = 0.45 (0.55)	Fail	Fail	1.6×10^{-3} %	1.9×10^{-3} %
PAR	FACTB	Titan	Intel 15.0.2 –O2	No	sol_factb_interstitial = 1.0 (0.1)	Fail	Fail	2.5×10^{-6} %	8.6×10^{-6} %
PAR	FACTIC	Titan	Intel 15.0.2 –O2	No	sol_factic_interstitial = 1.0 (0.4)	Fail	Fail	4.8×10^{-7} %	4.6×10^{-7} %
PAR	RH-MIN-LOW	Titan	Intel 15.0.2 –O2	No	cldfrc_rhminl = 0.85 (0.8975)	Fail	Fail	3.6×10^{-15} %	3.5×10^{-15} %
PAR	RH-MIN-HIGH	Titan	Intel 15.0.2 –O2	No	cldfrc_rhminh = 0.9 (0.8)	Fail	Fail	9.2×10^{-14} %	3.3×10^{-14} %
PAR	CLDFRC-DP	Titan	Intel 15.0.2 –O2	No	cldfrc_dp1 = 0.14 (0.10)	Fail	Fail	2.1×10^{-9} %	4.0×10^{-9} %
PAR	UW-SH	Titan	Intel 15.0.2 –O2	No	uwschu_rpen = 10.0 (5.0)	Fail	Fail	2.0×10^{-9} %	3.7×10^{-9} %
PAR	CONV-LND	Titan	Intel 15.0.2 –O2	No	zmconv_c0_lnd = 0.0035 (0.0059)	Fail	Fail	9.0×10^{-4} %	4.7×10^{-3} %
PAR	CONV-OCN	Titan	Intel 15.0.2 –O2	No	zmconv_c0_ocn = 0.0035 (0.045)	Fail	Fail	6.7×10^{-10} %	8.1×10^{-10} %
PAR	NU-P	Titan	Intel 15.0.2 –O2	No	nu_p = 1.0×10^{14} (1.0×10^{15})	Fail	Fail	2.5×10^{-10} %	1.4×10^{-10} %
PAR	NU	Titan	Intel 15.0.2 –O2	No	nu = 9.0×10^{14} (1.0×10^{15})	Fail	Fail	1.4×10^{-5} %	1.5×10^{-5} %

* Model was compiled without the "-fp-model" flag; All the other Intel simulations in the table used "-fp-model source" for Fortran and"-fp-model precise" for the C code.

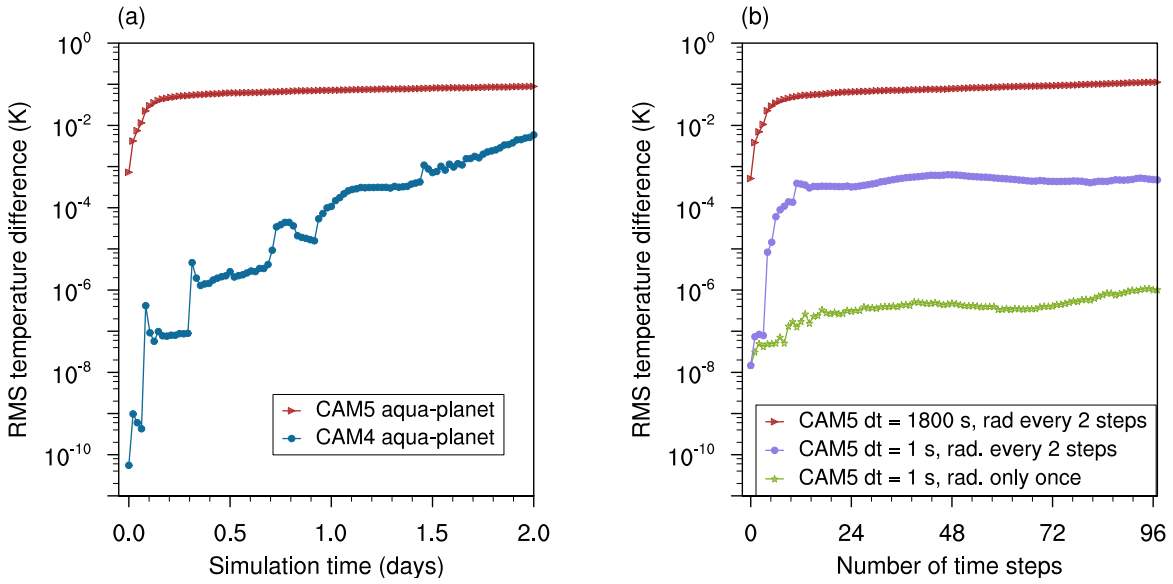

Figure 1. Examples of the evolution of RMS temperature difference (unit: K) caused by random perturbations of order 10^{-14} K imposed on the temperature initial conditions. ~~Blue and red indicate results from~~ (a) Aqua-planet simulations conducted with the CAM4 (blue) and CAM5~~, respectively~~.3 (red) physics parameterization suites using the default 1800 s time step. ~~All simulations were~~ (b) Simulations conducted ~~in~~ with the ~~aqua-planet mode~~ 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. All simulations used the spectral element dynamical core at approximately 1° horizontal resolution~~, with 26 vertical levels for the CAM4 physics and 30 levels for the CAM5 physics~~.

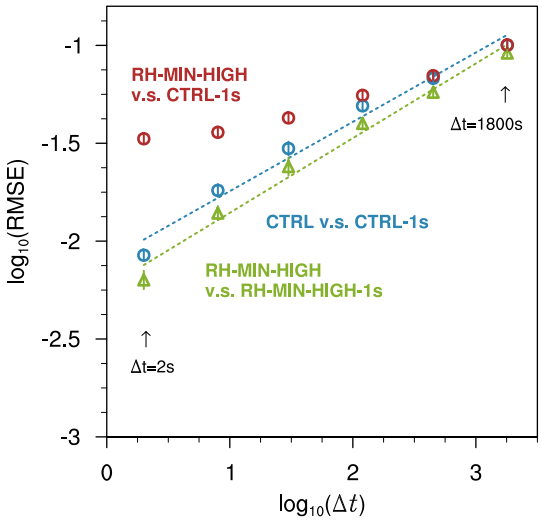

Figure 2. Convergence diagram showing the RMS solution differences calculated using the instantaneous 3D temperature field after 1 h of CAM5 integration. Blue circles and green triangles are the RMS differences relative to reference solutions obtained with the same code but using a 1 s time step. Red circles are the RMS differences between the reference solution of the CTRL model (1 s time step) and the RH-MIN-HIGH simulations with longer step sizes. Each marker shows the average RMS difference of 12 ensemble simulations that used different initial conditions sampled from different months of the year; the bars indicate the $\pm\sigma$ ranges where σ denotes the ensemble standard deviation. The dashed lines are linear fits between $\log_{10}(\mathrm{RMSD})$ and $\log_{10}(\Delta t)$.

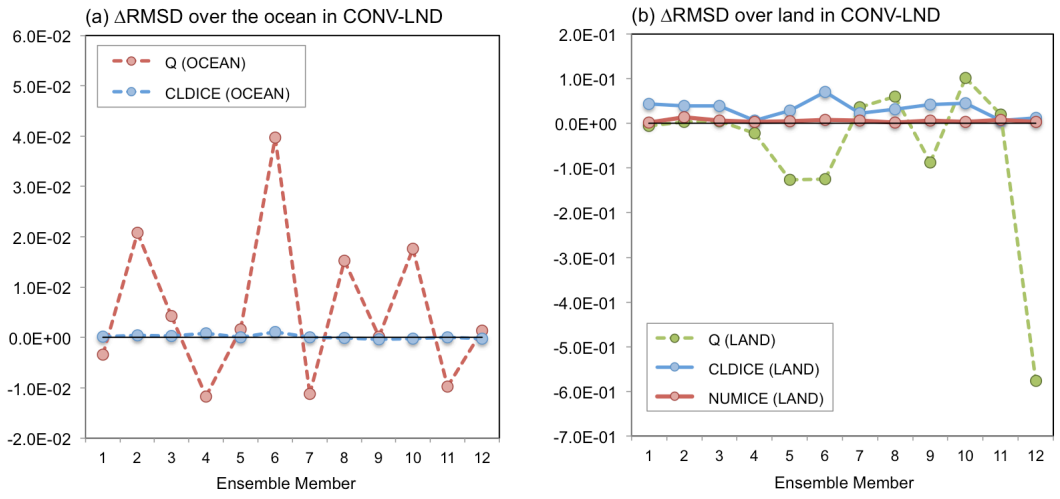

Figure 3. ~~$\Delta\mathrm{RMSD}_{j,m}$~~ $\Delta\mathrm{RMSD}_{t,j,m}$ of individual ensemble members after ~~5~~ $t = 5\,\mathrm{min}$ of model integration in the "CONV-LND" test case that was designed to fail the TSC test when all variables, domains, and ensemble members are considered (cf. Table 1 and Sect. 4.2). The values have been normalized by the mean RMSD of the trusted ensemble, i.e., ~~$\overline{\mathrm{RMSD}}_{\mathrm{trusted},j}$~~ $\overline{\mathrm{RMSD}}_{\mathrm{trusted},t,j}$, of the corresponding prognostic variables and domains. (a) ocean; (b) land. Dashed (solid) lines correspond to variables that passed (failed) the TSC test according to the criterion defined by Eq. (5). The prognostic variables shown in the figure are specific humidity (Q), grid-box mean ice crystal mass concentration in stratiform clouds (CLDICE), and grid-box mean ice crystal number concentration in stratiform clouds (NUMICE).

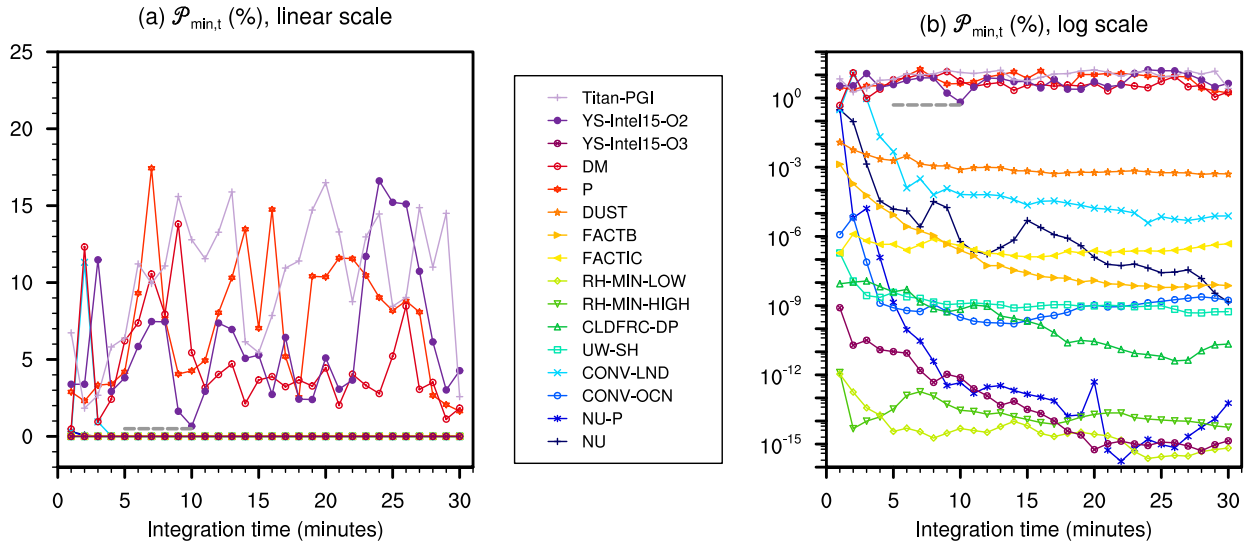

Figure 4. $\mathcal{P}_{\mathrm{min},t}$ as a function of model integration time, plotted in linear scale (a) and in logarithmic scale (b). The dashed gray lines indicate the threshold for assigning an overall "pass" or "fail" to a test ensemble (cf. Eq. 7 and the text above it).

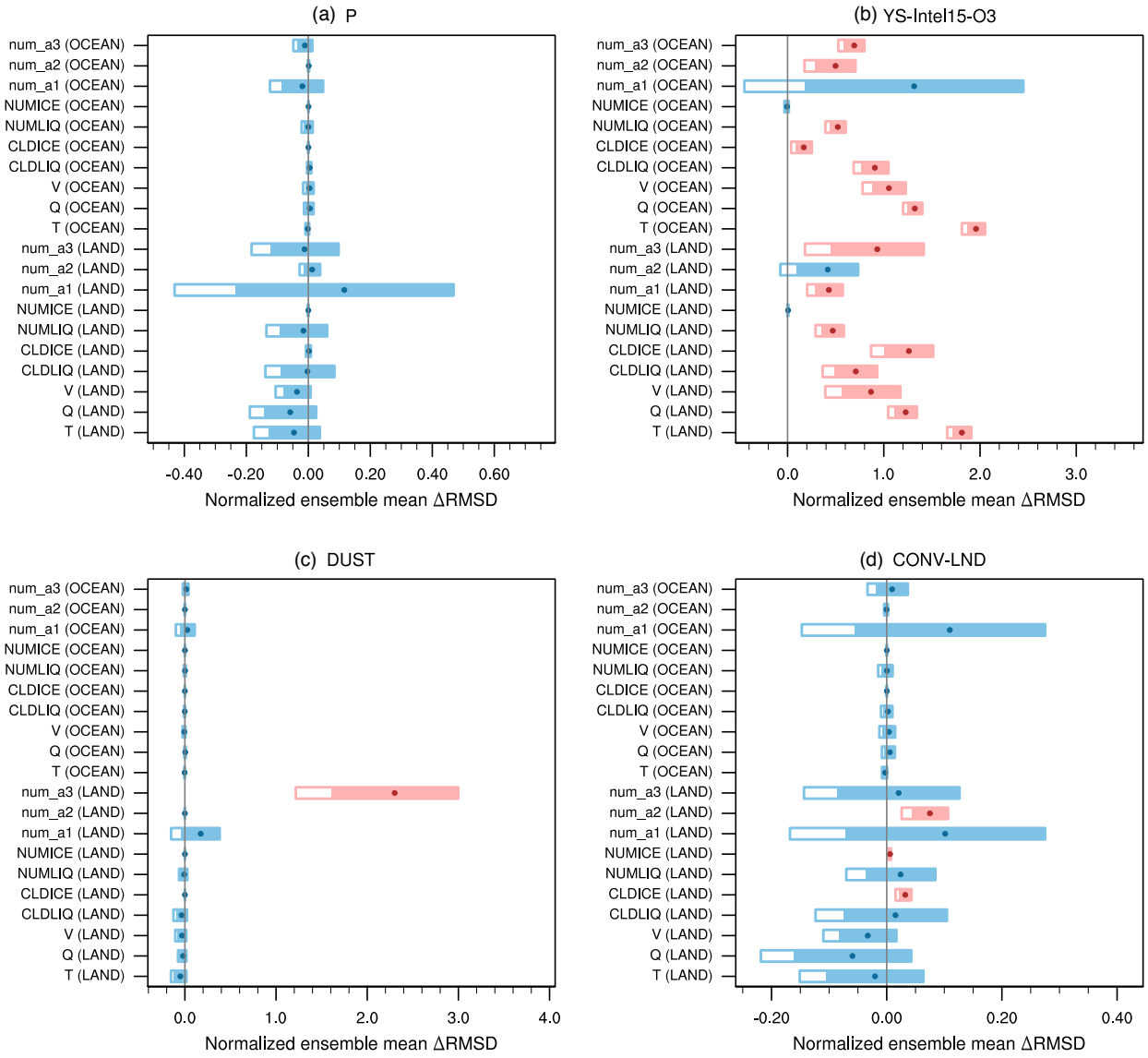

Figure 5. ~~The calculated ensemble~~ Ensemble mean ~~$\overline{\Delta\mathrm{RMSD}_j}$~~ $\overline{\Delta\mathrm{RMSD}}_{t,j}$ (dots) and the $\pm 2\sigma$ range of the mean (filled boxes) where σ denotes the standard deviation. The left end of an unfilled box shows the threshold value corresponding to ~~$\mathcal{P}_0 = 0.05$~~ $\mathcal{P}_0 = 0.5\,\%$ in the one-sided t-test. All values shown here have been normalized by the mean RMSD of the trusted ensemble, i.e., ~~$\overline{\mathrm{RMSD}}_{\mathrm{trusted},j}$~~ $\overline{\mathrm{RMSD}}_{\mathrm{trusted},t,j}$, of the corresponding prognostic variable and domain (cf. y-axis labels). Red and blue indicate fail and pass, respectively, according to the criterion defined by Eq. (5). Results are shown at $t = 5\,\mathrm{min}$ for four test cases: (a) ~~Cori-Intel~~P, (b) YS-Intel15-O3, (c) DUST, and (d) CONV-LND. The test case configurations are explained in Table 1 and Sect. 4.

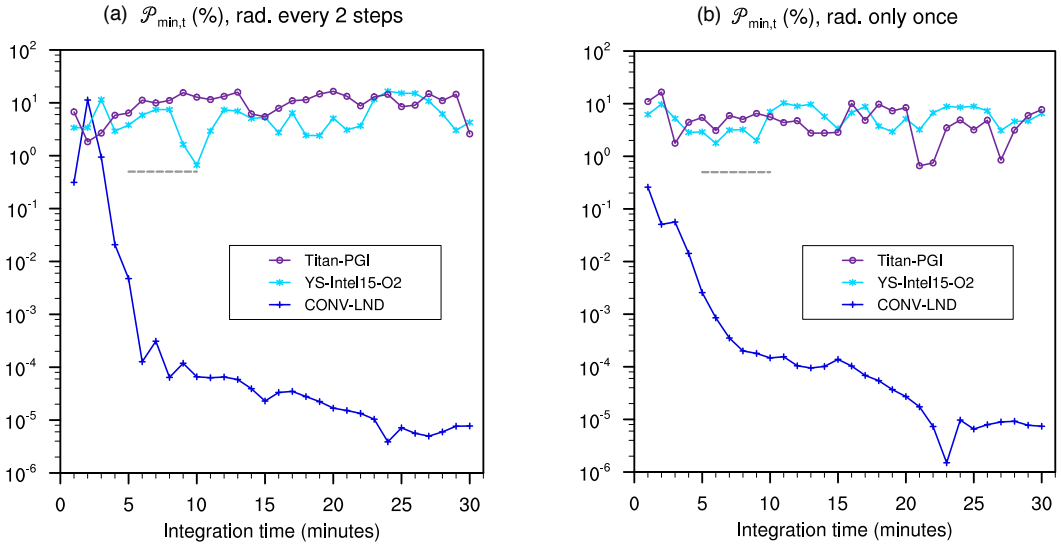

Figure 6. As in Fig. 4b, but showing only a few test scenarios to compare the results obtained from simulations where (a) radiation is calculated every other time step, and (b) radiation is calculated only at the beginning of the integration.