Geosci. Model Dev. Discuss., doi:10.5194/gmd-2016-142, 2016
Manuscript under review for journal Geosci. Model Dev.
Published: 28 June 2016

Geoscientific
Model Development
Discussions
Open Access

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. 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 using the Community Atmosphere Model version 5.3 (CAM5.3) that the proposed procedure can correctly distinguish rounding-level changes in the solutions from impacts of compiler optimization or parameter perturbations that are known to cause non-negligible differences in the simulated climate. 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 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. Oberkampf and Roy, 2010; Carson, 2002), 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 (Oberkampf and Roy, 2010). 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, 2013). 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".

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. 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.

The very fast evolution of initial perturbation is caused by multiple factors. 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. [1] 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.

[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.

Therefore we believe it will also be useful to use methods that can test 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 *consequence* of such growth as manifested in the *globally averaged annual mean* of 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 mean* fields. 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 paper we propose a complementary test procedure that builds 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 Sect. 2, and describes the implementation in Sect. 3. Evaluation of the test procedure is presented in Sect. 4. The conclusions are drawn in Sect. 5.

2 Test philosophy

In this section we start with a further clarification of the purpose of the code verification procedure (Sect. 2.1), then proceed to a discussion of the desirable features that guided the design of our new method (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

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.

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 and the need by large groups of model developers/users to perform code verification 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 operations, etc;
8. Computationally efficient.

The CAM-ECT of Baker et al. (2015) fulfills criteria 1–4 and 7. The use of global annual averages in the results assessment can lead to difficulty in detecting changes in small-scale features (criterion 6), as Baker et al. (2015) noted that CAM-ECT did not identify the impact of a change in a 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), the chance of missing a climate-changing modification (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).

The PerGro test of Rosinski and Williamson (1997) fulfills criterion 7 per design; it is very efficient in terms of computational cost thus fulfilling criterion 8; it 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 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 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.

The new test proposed in this paper aims at fulfilling all the 8 features listed above. It keeps the deterministic spirit of the PerGro test 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. 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 $1\,\mathrm{h}$ simulations were conducted using time step sizes ranging from $1\,\mathrm{s}$ to $1800\,\mathrm{s}$. The numerical solution with $\Delta t = 1\,\mathrm{s}$ 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 $1\,\mathrm{h}$ of model integration. To take into account possible flow-dependences of the numerical error, the exercise was repeated using initial conditions sampled from different months of a previously conducted long-term 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 convergence rate.

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 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 "RH-MIN-HIGH" hereafter. The RMSD calculated against a new reference solution using cldfrc_rhminh = 0.9 and $\Delta t = 1\,\mathrm{s}$ 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, 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, 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 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 the shortest possible interval at which the dynamical core and the various parameterized physical processes interact with each other; 1 s is 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.

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 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 overview

The global climate model used in this paper is CAM5 (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 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.

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. As for the physical quantities, the results shown in the present paper include RMSDs for $N_{\mathrm{var}} = 10$ 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 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 user wishes to monitor more fields.

The test procedure includes three steps as described below. Steps 1 and 2 are needed every time a new baseline model with modified climate 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 3D instantaneous values of the N_{var} prognostic variables listed above, plus the values of surface pressure and land fraction, all in double precision.

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 are denoted as $\mathrm{RMSD}_{\mathrm{trusted}}$.

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 $\mathrm{RMSD}_{\mathrm{test}}$. Now define

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

and denote the M-member average by $\overline{\Delta\mathrm{RMSD}}_j$. For each prognostic variable and domain (i.e. each j), we assume the ensemble mean of $\Delta\mathrm{RMSD}_j$ is a random variable μ_j. The students t-test is performed to accept or reject the null hypothesis that μ_j is statistically zero. The alternative hypothesis is $\mu_j > 0$. The null hypothesis is rejected, i.e. the jth variable fails the TSC test, if

$$\mathcal{P}\left(\mu_j > \overline{\Delta\mathrm{RMSD}}_j\right) < \mathcal{P}_0\,, \tag{5}$$

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

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

then the ensemble fails the TCS test, and the code or software/hardware change is considered climate-changing.

$M = 12$ ensemble members are included in the TSC test version 1.0 which we evaluate in the next section. 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-dependences of the numerical error. The need for an ensemble is demonstrated in Fig. 3 where the normalized $\Delta\mathrm{RMSD}$ 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", cf. Table 1 and Sect. 4.1 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 $\Delta\mathrm{RMSD}_{j,m}$ have been normalized by the mean RMSD of the trusted ensemble, i.e., by $\overline{\mathrm{RMSD}}_{\mathrm{trusted},j}$. Our exploration has indicated that, due to the complexity and nonlinearity of the model equations, the values of $\Delta\mathrm{RMSD}$ of a passing variable from individual ensemble members often are distributed around zero (Fig. 3a). Therefore a single positive $\Delta\mathrm{RMSD}_{j,m}$ cannot be viewed as sufficient evidence of non-convergence towards the reference solution. The magnitude of a positive $\Delta\mathrm{RMSD}_{j,m}$ 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 $\Delta\mathrm{RMSD}$, 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. 6 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 step to diverge. In the next section, we present results from 30 min simulations with the test diagnostics calculated every minute to reveal the initial evolution of ΔRMSD.

4 Evaluation of the new method

We challenged the TSC test with a number of scenarios to verify whether it issued the expected pass/fail signal. A reference ensemble with 1 s time step and a trusted ensemble with 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 simulations were then conducted under three groups (Table 1):

Group E ("computing Environment") simulations used the same code but different computers, compiler versions, 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");

The fifth case ("YS-Iintel15-O3") used a higher optimization level on Yellowstone, and had been found by Baker et al. (2015) to produce incorrect answers. (We note that such incorrect answers are produced only when the model is compiled without the "-fp-model" flag. If the "-fp-model source" flag is applied to the Fortran code and "-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.)

In group P1 ("Parameter perturbation set 1") we repeated all the parameter perturbation experiments presented by Baker et al. (2015) (cf. Section 4.3 therein). One parameter in CAM5's physics package was modified in each experiment, and the perturbations were expected to cause physically significant changes in the simulated climate. 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 $10^{-18}\,\mathrm{kg\,kg^{-1}}$ to $10^{-8}\,\mathrm{kg\,kg^{-1}}$. 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 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

Summaries of the test results after 5 min of model integration are presented in Table 1 and in Fig. 4. 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. 4b 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 %.

In Fig. 5, the statistical distributions of μ_j (the mean ΔRMSD) estimated from the 12-member ensembles are shown 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., $\overline{\mathrm{RMSD}}_{\mathrm{trusted},j}$. The dots indicate the observed ensemble mean (i.e. $\overline{\Delta\mathrm{RMSD}}_j$), 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 $\mathcal{P}_0 = 0.05\,\%$ 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 ensemble and the trusted ensemble are very small, on the order of 10^{-4}, and the value of 0 lies within the $\pm 2\sigma$ range of the observed $\overline{\Delta\mathrm{RMSD}}_j$ for most of the 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 20 variables failed the TSC test with a $\mathcal{P}_{\min}$ of 7×10^{-14} %, indicating 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 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 $\overline{\Delta\mathrm{RMSD}}_j$ corresponds to a very small $\mathcal{P}\left(\mu_j > \overline{\Delta\mathrm{RMSD}}_j\right)$ of 0.0015 % (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 %. Since deep convection over land happens in limited areas, and the natural variability is very strong, it is not surprising that $\overline{\Delta\mathrm{RMSD}}_j$ 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} %.

4.2 30 min simulations

To understand the initial evolution of $\Delta\mathrm{RMSD}$, we conducted 30 min simulations and calculated the test diagnostics after every minute. Fig. 6 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. 6a). Values of $\mathcal{P}_{\min}$ in the failing cases are distinctly closer to zero (Fig. 6a), often showing a clear decrease in the first 15 min and considerably slower changes afterwards (Fig. 6b). 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 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 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 performed using the default time step. Compared to the CAM-ECT which includes 151 one-year simulations in the reference ensemble and 3 one-year simulations in the test ensemble, the TSC test is a factor of 450 cheaper to obtain the reference simulations, and a factor of 30 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 simulation with 2 s time step was about 5 min; the entire set of 12 simulations was typically completed in 10 min to 20 min after submission, and the time between first job submission and last job completion rarely exceeded 1 h.

5 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 dependences of the numerical error.

Using the CAM5 model, we demonstrated that the test procedure based on 5 min simulations with 2 s step size (i.e. a total of 150 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 climate statistics. The test hence provides an objective and computationally 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 a few minutes. 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 to assess the time step convergence of individual parameterizations in isolation. This implies the new test procedure can be applied to subcomponents of the model code thus facilitate debugging.

Because the test design uses the time stepping error associated with 2 s 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.

The new test is based 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 (Pincus et al., 2003) to represent the subgrid-scale cloud variability, the resulting randomness is avoided in our test design by fixing the radiation time step at 1 h 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.

6 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 research described in this paper was funded by the Accelerated Climate Modeling for Energy (ACME) project through the Office of Biological and Environmental Research (BER) in the U.S. Department of Energy (DOE) Office of Science. The basis of the work, the time step convergence study, was previously supported by the DOE Office of Science 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 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, doi:10.1109/WSC.2002.1172868, 2002.

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.

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.

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 SST and Fully Coupled Experiments, J. Clim., 26, 5150–5168, doi:10.1175/JCLI-D-12-00236.1, 2013.

Oberkampf, W. and Roy, C.: Verification and Validation in Scientific Computing, Cambridge University Press, https://books.google.com/books?id=7d26zLEJ1FUC, 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, doi:10.1029/2002JD003322, http://dx.doi.org/10.1029/2002JD003322, 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.

Table 1. CAM5 simulations conducted to evaluate the effectiveness of the TSC method. Simulations in group E ("computing Environment") used the same code but different computers, compiler versions, or optimization levels. Group P1 ("Parameter perturbation set 1") 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 $\mathcal{P}_{\min}$ can be found in Sect. 3.2.

Group	Case name	Computer	Compiler/ optimization	Model parameters	Pass/fail (expected)	Pass/fail (5 min)	$\mathcal{P}_{\min}$ (5 min)	Pass/fail (30 min)	$\mathcal{P}_{\min}$ (30 min)
-	CTRL	Titan	Intel 15.0.2 –O2	All default	-	-	-	-	-
E	Titan-PGI	Titan	PGI 15.3.0 –O2	All default	Pass	Pass	5.5 %	Pass	7.7 %
E	Constance-Intel	Constance	Intel 15.0.1 –O2	All default	Pass	Pass	15 %	Pass	9.3 %
E	Cori-Intel	Cori Phase I	Intel 16.0.0 –O2	All default	Pass	Pass	0.6 %	Pass	1.5 %
E	YS-Intel15-O2	Yellowstone	Intel 15.0.0 –O2 *	All default	Pass	Pass	2.9 %	Pass	6.5 %
E	YS-Intel15-O3	Yellowstone	Intel 15.0.0 –O3 *	All default	Fail	Fail	7.3×10^{-14} %	Fail	2.8×10^{-13} %
P1	DUST	Titan	Intel 15.0.2 –O2	dust_emis_fact = 0.45 (0.55)	Fail	Fail	1.5×10^{-3} %	Fail	3.0×10^{-4} %
P1	FACTB	Titan	Intel 15.0.2 –O2	sol_factb_interstitial = 1.0 (0.1)	Fail	Fail	8.7×10^{-6} %	Fail	4.4×10^{-9} %
P1	FACTIC	Titan	Intel 15.0.2 –O2	sol_factic_interstitial = 1.0 (0.4)	Fail	Fail	2.3×10^{-7} %	Fail	1.4×10^{-7} %
P1	RH-MIN-LOW	Titan	Intel 15.0.2 –O2	cldfrc_rhminl = 0.85 (0.8975)	Fail	Fail	6.2×10^{-15} %	Fail	4.1×10^{-15} %
P1	RH-MIN-HIGH	Titan	Intel 15.0.2 –O2	cldfrc_rhminh = 0.9 (0.8)	Fail	Fail	8.3×10^{-16} %	Fail	1.2×10^{-14} %
P1	CLDFRC-DP	Titan	Intel 15.0.2 –O2	cldfrc_dp1 = 0.14 (0.10)	Fail	Fail	2.7×10^{-9} %	Fail	3.3×10^{-10} %
P1	UW-SH	Titan	Intel 15.0.2 –O2	uwschu_rpen = 10.0 (5.0)	Fail	Fail	1.3×10^{-9} %	Fail	1.8×10^{-10} %
P1	CONV-LND	Titan	Intel 15.0.2 –O2	zmconv_c0_lnd = 0.0035 (0.0059)	Fail	Fail	2.6×10^{-3} %	Fail	7.4×10^{-6} %
P1	CONV-OCN	Titan	Intel 15.0.2 –O2	zmconv_c0_ocn = 0.0035 (0.045)	Fail	Fail	2.8×10^{-10} %	Fail	4.7×10^{-11} %
P1	NU-P	Titan	Intel 15.0.2 –O2	nu_p = 1.0×10^{14} (1.0×10^{15})	Fail	Fail	1.0×10^{-11} %	Fail	5.5×10^{-14} %
P1	NU	Titan	Intel 15.0.2 –O2	nu = 9.0×10^{14} (1.0×10^{15})	Fail	Fail	8.4×10^{-6} %	Fail	1.8×10^{-10} %
P2	RH-MIN-LOW-2	Titan	Intel 15.0.2 –O2	cldfrc_rhminl = 0.89 (0.8975)	Fail	Fail	1.6×10^{-11} %	Fail	1.4×10^{-9} %
P2	RH-MIN-LOW-3	Titan	Intel 15.0.2 –O2	cldfrc_rhminl = 0.897 (0.8975)	Fail	Fail	1.1×10^{-2} %	Fail	4.7×10^{-4} %
P2	QSMALL	Titan	Intel 15.0.2 –O2	qsmall = 10^{-8} (10^{-18})	Fail	Fail	1.9×10^{-9} %	Fail	8.2×10^{-11} %

[*] 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.

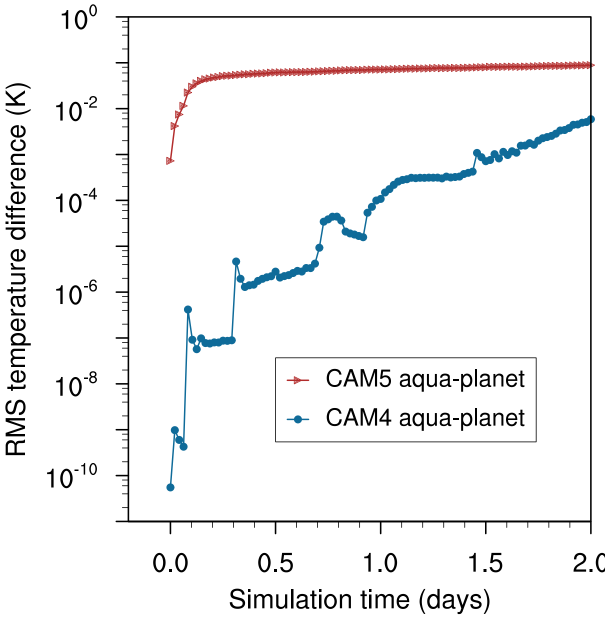

Figure 1. Examples of the evolution of RMS temperature difference (unit: K) caused by random perturbations of order 10^{-14} K imposed on the temperature initial conditions. Blue and red indicate results from CAM4 and CAM5, respectively. All simulations were conducted in the aqua-planet mode and using 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.

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2016-142, 2016
Manuscript under review for journal Geosci. Model Dev.
Published: 28 June 2016

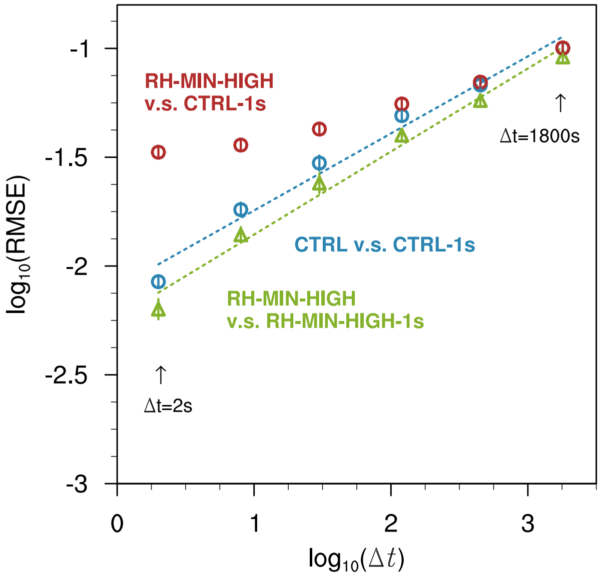

Figure 2. Convergence diagram showing the RMS solution differences calculated using the instantaneous 3D temperature field after 1 h of CAM5 integration. Blue circles and green triangles are the RMS differences relative to reference solutions obtained with the same code but using a 1 s time step. Red 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 $\log_{10}(\Delta t)$.

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2016-142, 2016
Manuscript under review for journal Geosci. Model Dev.
Published: 28 June 2016

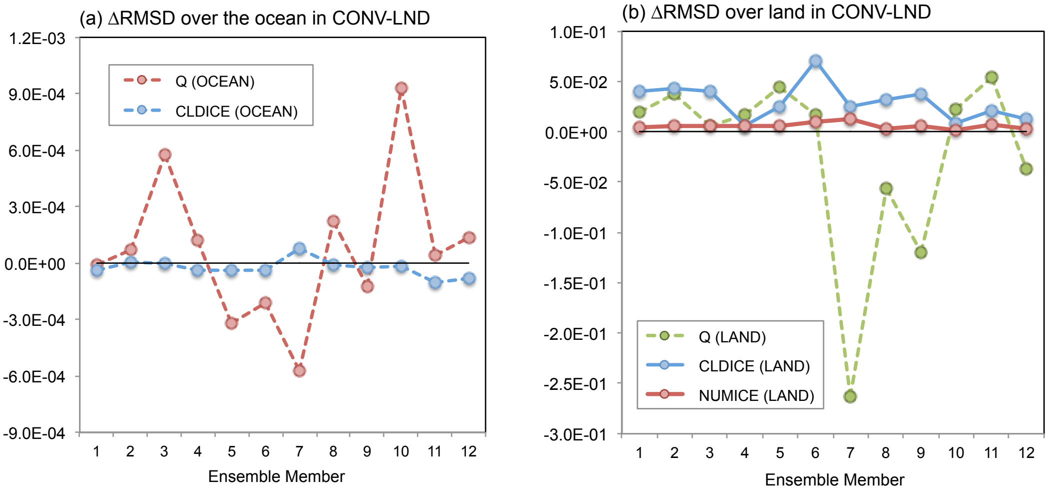

Figure 3. $\Delta\mathrm{RMSD}_{j,m}$ of individual ensemble members after $5\,\mathrm{min}$ of model integration in the "CONV-LND" test case that was designed to fail the TSC test when all variables, domains, and ensemble members are considered (cf. Table 1 and Sect. 4.1). The values have been normalized by the mean RMSD of the trusted ensemble, i.e., $\overline{\mathrm{RMSD}}_{\mathtt{trusted},j}$, 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).

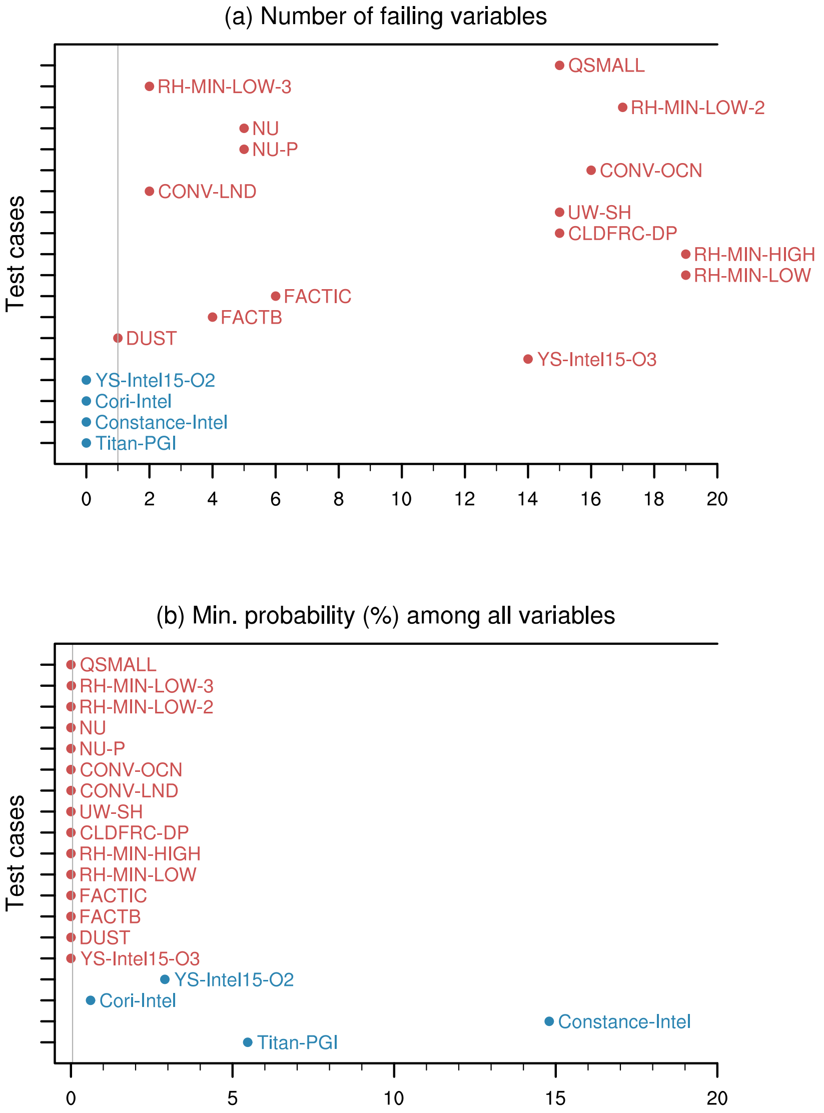

Figure 4. (a) The number of variables (out of a total of $N_{\mathrm{var}} \times N_{\mathrm{dom}} = 20$) that fail the TSC test according to the criterion defined by Eq. (5). (b) The minimum probability $\mathcal{P}_{\mathrm{min}}$ (Eq. 6) in each test case. Red and blue indicate overall fail and pass, respectively. The gray vertical line in panel (b) indicates the threshold probability $\mathcal{P}_0 = 0.05\,\%$. The test cases names appearing to the right of the filled circles are explained in Table 1 and Sect. 4.

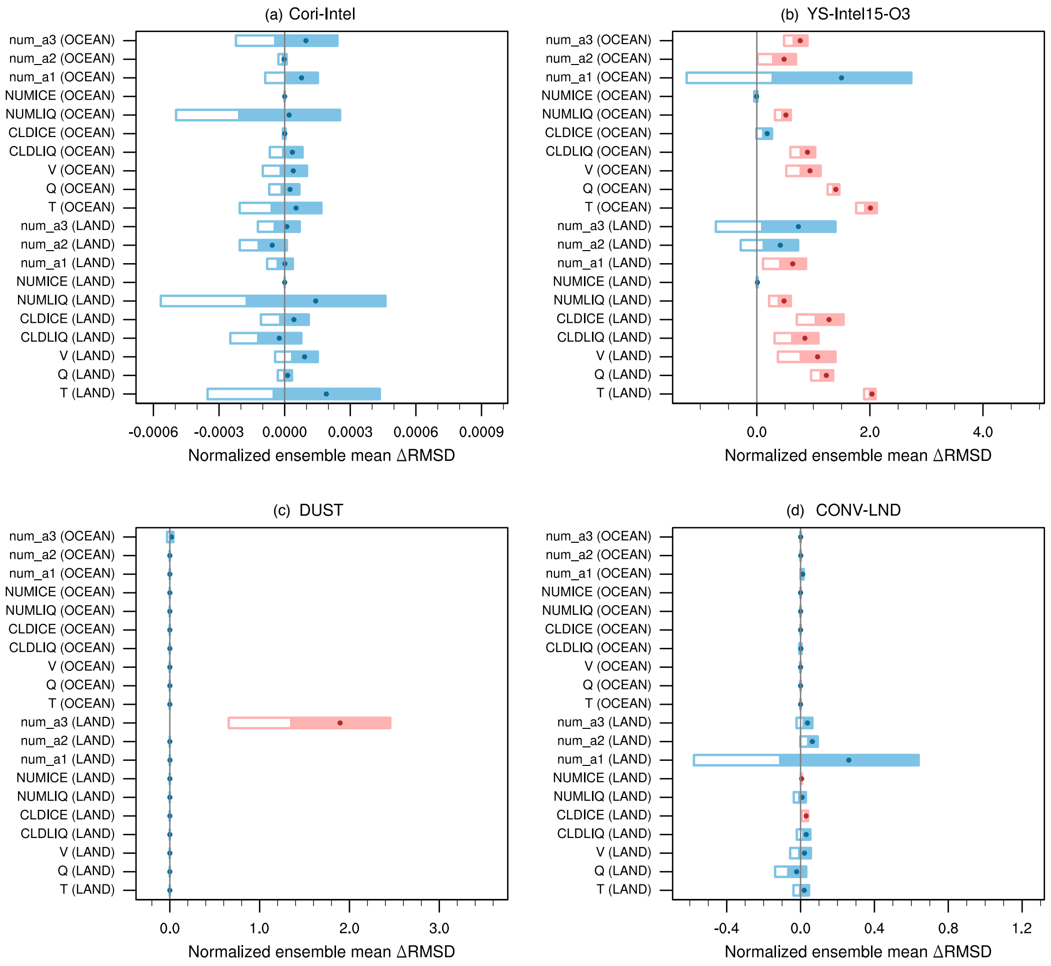

Figure 5. The calculated ensemble mean $\overline{\Delta\mathrm{RMSD}}_j$ (dots) and the $\pm 2\sigma$ range of the mean (filled boxes) where σ denotes the standard deviation. The left end of an unfilled box shows the threshold value corresponding to $\mathcal{P}_0 = 0.05\,\%$ in the one-sided t-test.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

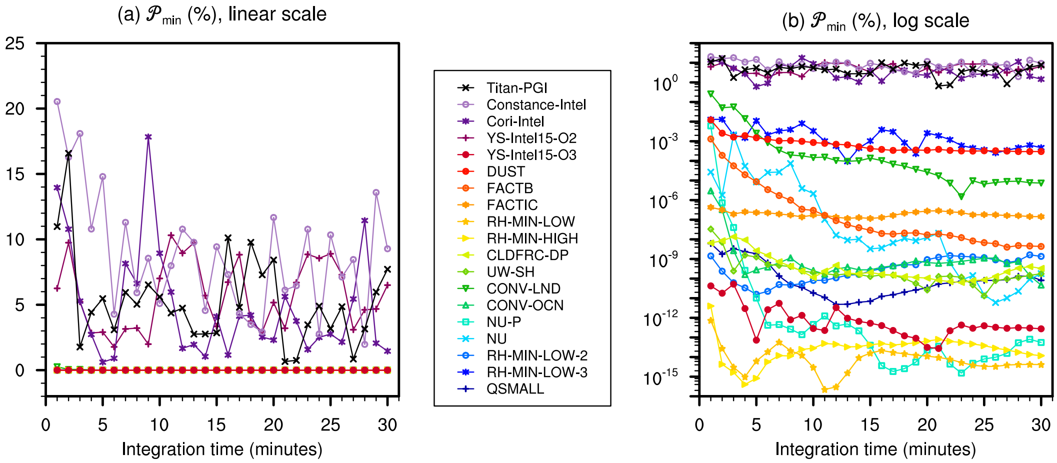

Figure 6. $\mathcal{P}_{\min}$ as a function of model integration time, plotted in linear scale (a) and in logarithmic scale (b).