Geosci. Model Dev., 6, 1157–1171, 2013
www.geosci-model-dev.net/6/1157/2013/
doi:10.5194/gmd-6-1157-2013

Geoscientific Model Development
Open Access

Failure analysis of parameter-induced simulation crashes in climate models

D. D. Lucas[1], R. Klein[1,2], J. Tannahill[1], D. Ivanova[1], S. Brandon[1], D. Domyancic[1], and Y. Zhang[1]

[1]Lawrence Livermore National Laboratory, Livermore, CA, USA
[2]Department of Astronomy, University of California, Berkeley, CA 94720, USA

Correspondence to: D. D. Lucas (ddlucas@alum.mit.edu)

Received: 3 January 2013 – Published in Geosci. Model Dev. Discuss.: 24 January 2013
Revised: 28 June 2013 – Accepted: 2 July 2013 – Published: 7 August 2013

Abstract. Simulations using IPCC (Intergovernmental Panel on Climate Change)-class climate models are subject to fail or crash for a variety of reasons. Quantitative analysis of the failures can yield useful insights to better understand and improve the models. During the course of uncertainty quantification (UQ) ensemble simulations to assess the effects of ocean model parameter uncertainties on climate simulations, we experienced a series of simulation crashes within the Parallel Ocean Program (POP2) component of the Community Climate System Model (CCSM4). About 8.5 % of our CCSM4 simulations failed for numerical reasons at combinations of POP2 parameter values. We applied support vector machine (SVM) classification from machine learning to quantify and predict the probability of failure as a function of the values of 18 POP2 parameters. A committee of SVM classifiers readily predicted model failures in an independent validation ensemble, as assessed by the area under the receiver operating characteristic (ROC) curve metric (AUC > 0.96). The causes of the simulation failures were determined through a global sensitivity analysis. Combinations of 8 parameters related to ocean mixing and viscosity from three different POP2 parameterizations were the major sources of the failures. This information can be used to improve POP2 and CCSM4 by incorporating correlations across the relevant parameters. Our method can also be used to quantify, predict, and understand simulation crashes in other complex geoscientific models.

1 Introduction

Modern global three-dimensional climate models are extraordinarily complex pieces of science (e.g., Randall et al., 2007; Gent et al., 2011; The HadGEM2 Development Team, 2011) and software engineering (Easterbrook et al., 2011; Rugaber et al., 2011; Easterbrook, 2010). They contain over a million lines of code (Easterbrook and Johns, 2009; Easterbrook, 2012) and use hundreds to thousands of files, functions, and subroutines to solve equations of state and conservation laws for the flows of matter, energy, and momentum within and between the atmosphere, oceans, land, and other reservoirs of the Earth system (Washington and Parkinson, 2005). They also use numerous algorithms of biological, chemical, geologic, and anthropogenic processes to simulate the cycles of carbon, nitrogen, sulfur, aerosols, ozone, greenhouse gases, and other climate-relevant quantities of interest. To compound this complexity, these algorithms operate across many orders of magnitude in space and time, and contain constituents that exist in gas, liquid, solid and mixed phases.

Given this enormous range of scientific complexity, climate models are vulnerable to many types of software design and implementation issues. Climate models are developed in a manner analogous to large open source and agile software projects (Easterbrook and Johns, 2009). Based on current best understanding, small groups of scientists create, test, and refine modules for select climate processes or sub-systems (e.g., atmospheric convection or aerosol microphysics). Their software changes are committed upstream to the climate model codebase, and the cycle is repeated until the model simulations reproduce desired features (i.e.,

model validation). Varying amounts of software testing are conducted throughout the cycle, but formal code verification practices (e.g., see D'Silva et al., 2008) are only recently starting to be considered for climate model development (Clune and Rood, 2011; Farrell et al., 2011). Nonetheless, the concentration on sound science, as opposed to software correctness, has led to climate models that contain fewer software defects than other comparably sized projects (Pipitone and Easterbrook, 2012).

Software issues aside, many potential problems still arise with scientific representations in complex models. As code verification can be used to find software bugs, emerging tools being developed in the field of uncertainty quantification (UQ) (see National Research Council Report, 2012) can help pinpoint scientific discrepancies in simulation models, the knowledge of which can be used to guide and improve model development. Primary UQ targets for climate models are schemes containing parameters with adjustable values. These schemes represent physical processes that are not fully understood or cannot be directly simulated at the model resolutions of interest (e.g., Stensrud, 2009). Parameterizations like this are often developed in isolation, so they can respond in unexpected ways when inserted in nonlinear climate models and coupled to other parameterizations. Small perturbations to the values of the adjustable parameters can amplify and lead to large changes in simulation outputs. In some cases, the simulations may fail altogether.

We report here on a series of simulation crashes encountered while running perturbed parameter UQ ensembles of the Community Climate System Model Version 4 (CCSM4) (Gent et al., 2011; CCSM4, 2012). Treating the simulation crash problem as a black box in which we know only the values of the input parameters and a binary outcome flag indicating whether the simulations ultimately failed or were completed, information that does not require detailed scientific knowledge, we present a method that successfully predicted crashes in independent simulations and pinpointed the model parameters that caused the failures.

Numerous studies have applied UQ techniques to climate models similar to CCSM4 (e.g., Murphy et al., 2004; Stainforth et al., 2005; Jackson et al., 2008; Sanderson, 2011; Shiogama et al., 2012). However, analogous simulation failures and crashes have been reported far less often, though we suspect that they occur more frequently than indicated by the relatively limited number of documented cases (i.e., a reporting bias). Failures, crashes, or bifurcations have been reported for climate models of both intermediate complexity (Webster et al., 2004; Annan et al., 2005; Edwards et al., 2011) and full complexity (see supplementary discussion in Stainforth et al., 2005). The failures in these cases were attributed to numerical instabilities, or to particular climate phenomena, such as the collapse of the Atlantic meridional overturning circulation or accelerated changes through positive feedbacks in the models. Webster et al. (2004) and Edwards et al. (2011) present methods for calculating the probability of failure for an input set of parameter values. In addition, Edwards et al. (2011) use the failure probability to design and prescreen ensemble members in follow-up ensembles.

Our analysis and approach are similar to Edwards et al. (2011), but with the added challenges of applying them to a climate model system that is computationally more demanding, uses smaller ensemble sizes, has more parameter uncertainty dimensions, and exhibits fewer simulation failures. To help overcome these challenges, we use machine learning algorithms to calculate and predict the failure probability. As climate models and other geoscientific codes become more complex and UQ studies more commonplace, we fully expect parameter-induced simulation crashes to occur in these models with a greater frequency. Our failure analysis method will be beneficial for quantifying and determining the causes of these crashes.

2 Overview of climate simulations

Different sets of perturbed parameter UQ ensembles were executed as part of a broad effort to quantify and constrain uncertainties in the atmospheric, sea ice, and ocean model components of CCSM4 (Gent et al., 2011). The failures reported here occurred during simulations that perturbed parameter values in the Parallel Ocean Program (POP2), the ocean component of CCSM4. For these experiments, POP2 was coupled with the sea ice model, while data-based components were used for the land and atmosphere. The simulations were integrated for 10 yr, and the system was forced with climatological air–sea flux data using normal year forcing from Large and Yeager (2009). Further details about POP2 and the UQ ensembles are given below.

2.1 Ocean model and parameters

POP2 is a state of the art depth-level model of the general ocean circulation that solves the 3-D primitive equations of rotational fluid dynamics and thermodynamics with standard approximations of Boussinesq and hydrostatics. It is developed and maintained at Los Alamos National Laboratory (Smith et al., 2010) and is the ocean component of CCSM4 developed at the National Center for Atmospheric Research (Gent et al., 2011; Danabasoglu et al., 2012). The current simulations use the displaced-pole coordinate grid with the pole centered over Greenland and have a nominal horizontal resolution of 1°. Vertically it resolves 60 depth levels with resolution varying from 10 m in the upper ocean (surface to 200 m) to 250 m in the deeper ocean. Refer to Smith et al. (2010) and Danabasoglu et al. (2012) for more information.

The ocean model parameters perturbed in this study were selected by POP2 model developers. They are used in six different sub-grid scale parameterizations to simulate the effects of horizontal and vertical turbulent mixing in the oceans. The

Table 1. Parameters sampled in the CCSM4 parallel ocean model.

	Parameter[a]	[low, default, high]	Scale[b]	Module	Description
1	vconst_corr	$[0.3, 0.6, 1.2] \times 10^{7}$	lin	hmix_aniso	variable viscosity parameter (vconst_1, vconst_6)
2	vconst_2	[0.25, 0.5, 2.0]	log	hmix_aniso	variable viscosity parameter
3	vconst_3	[0.16, 0.16, 0.2]	lin	hmix_aniso	variable viscosity parameter
4	vconst_4	$[0.5, 2.0, 10.0] \times 10^{-8}$	log	hmix_aniso	variable viscosity parameter
5	vconst_5	[2, 3, 5]	lin	hmix_aniso	variable viscosity parameter
6	vconst_7	[30.0, 45.0, 60.0]	lin	hmix_aniso	variable viscosity parameter
7	ah_corr	$[2.0, 3.0, 4.0] \times 10^{7}$	lin	hmix_gm	diffusion coefficient for Redi mixing (ah) and background horizontal diffusivity within the surface boundary layer (ah_bkg_srfbl)
8	ah_bolus	$[2.0, 3.0, 4.0] \times 10^{7}$	lin	hmix_gm	diffusion coefficient for bolus mixing
9	slm_corr	[0.05, 0.3, 0.3]	log	hmix_gm	maximum slope for bolus (slm_b) and Redi terms (slm_r)
10	efficiency_factor	[0.05, 0.07, 0.1]	lin	mix_submeso	efficiency factor for submesoscale eddies
11	tidal_mix_max	[25.0, 100.0, 200.0]	log	tidal	tidal mixing threshold
12	vertical_decay_scale	$[2.5, 5.0, 20.0] \times 10^{4}$	log	tidal	vertical decay scale for tide induced turbulence
13	convect_corr	$[1.0, 10.0, 50.0] \times 10^{3}$	log	vertical_mix	tracer (convect_diff) and momentum (convect_visc) mixing coefficients in diffusion option
14	bckgrnd_vdc1	[0.032, 0.16, 0.8]	log	vmix_kpp	base background vertical diffusivity
15	bckgrnd_vdc_ban	[0.5, 1.0, 1.0]	lin	vmix_kpp	Banda Sea diffusivity
16	bckgrnd_vdc_eq	[0.01, 0.01, 0.5]	log	vmix_kpp	equatorial diffusivity
17	bckgrnd_vdc_psim	[0.1, 0.13, 0.5]	log	vmix_kpp	maximum PSI induced diffusivity
18	Prandtl	[4.0, 10.0, 20.0]	log	vmix_kpp	ratio of background vertical viscosity and diffusivity

[a] Individual `_corr` parameters (numbers 1, 7, 9, and 13) are used to represent the correlated pair of parameters given in the description. For example, values drawn for `vconst_corr` are assigned to `vconst_1` and `vconst_6`. [b] Linear and logarithmic scales are used for parameter ranges that have ratios of high/low < 5 and high/low ≥ 5, respectively.

parameters and their uncertainty ranges are summarized in Table 1. Parameters 1–6 are used to capture horizontal mixing of momentum with spatially anisotropic viscosity (Large et al., 2001; Smith and McWilliams, 2003). Parameters 7–9 are used for horizontal mixing of tracers via isopycnal eddy-induced transport (Gent and McWilliams, 1990). Parameters 10–12 are used in recently developed schemes to simulate submesoscale and mixed-layer eddies (Fox-Kemper et al., 2008) and abyssal tidal mixing (Jayne, 2009). Parameters 13–18 are used for vertical convection and vertical mixing with the K-profile parameterization (KPP) scheme (Large et al., 1994).

2.2 UQ ensembles

Table 2 summarizes the UQ ensemble simulations. Three separate ensemble studies were conducted, each consisting of 180 simulations. The table also indicates the contributions of the studies to different types of analysis. For instance, studies 1 and 2 were used to train machine learning algorithms to learn about simulation crashes (see Sect. 4.3), while study 3 was used to test the ability to predict simulation crashes (see Sect. 4.4). Out of 540 total simulations, there were 46 failures, with the failures occurring at various times during the integration period. Each of the three studies used a Latin hypercube method to sample the values of the 18 POP2 parameters and a different random seed to generate the ensemble. The model parameters were represented using standard uniform or log-uniform probability distribution functions normalized to [0, 1] using the ranges (low and high values) and scales (linear and logarithm) noted in Table 1.

The Latin hypercube method is a stratified, space-filling variant of Monte Carlo sampling that is used extensively in UQ and uncertainty analysis (McKay et al., 1979; Helton and Davis, 2003). This sampling approach has superior variance reduction properties over standard Monte Carlo sampling for some problems (Stein, 1987). For an ensemble size of N, Latin hypercube splits each of the D parameter distributions into N intervals of equal probability, resulting in a multi-dimensional grid with N^D separate bins. For our case, $D = 18$ and $N = 180$. Ensemble members are obtained by selecting parameter values from different bins chosen at random, with the important constraint that the bins are chosen so that each interval along every parameter dimension is sampled only one time per ensemble. An example of a five-member Latin hypercube ensemble for two parameters is given by bins with indices $(1, 4)$, $(2, 2)$, $(3, 5)$, $(4, 3)$, and $(5, 1)$, which is one of 120 valid possibilities for this configuration. For our three UQ ensembles, the actual Latin hypercube sample points are illustrated in Figs. 1 and 2 for four of the POP2 parameters. These figures show that the Latin hypercube sample point coverage is uniform and dense in one and two dimensions.

Ensembles were generated using the Lawrence Livermore National Laboratory UQ Pipeline (Walter, 2010; Tannahill et al., 2011), which is a Python-based scientific workflow system for running and analyzing concurrent UQ simulations on high performance computers. Using a simple, non-intrusive interface to simulation models, it provides strategies for sampling high dimensional uncertainty spaces, analyzing ensemble output, constructing surrogate approximation models (e.g., Forrester et al., 2008),

Table 2. Latin Hypercube Studies of the CCSM4 Parallel Ocean Program.

Study	Simulations	Successes	Failures	Failure rate	Data used in Section				
					3	4.3	4.4	4.5	5
1	180	160	20	11.1 %	✓	✓			✓
2	180	168	12	6.7 %	✓	✓			✓
3	180	166	14	7.8 %	✓		✓	✓	✓
Total	540	494	46	8.5 %					

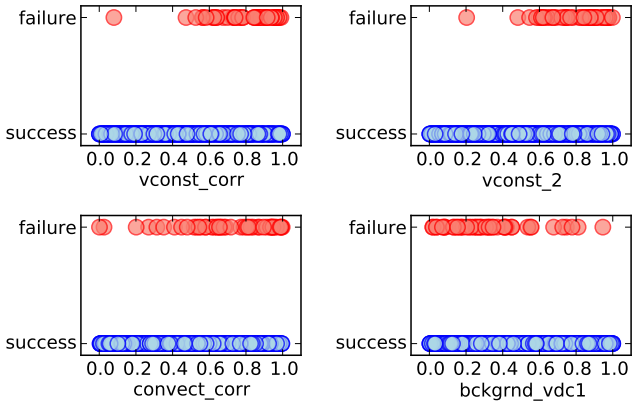

Fig. 1. Climate model simulation successes and failures are shown for one-dimensional projections of the values of four ocean parameters in 540 Latin hypercube experiments that sampled 18 model parameters. Parameter values are normalized using the ranges in Table 1.

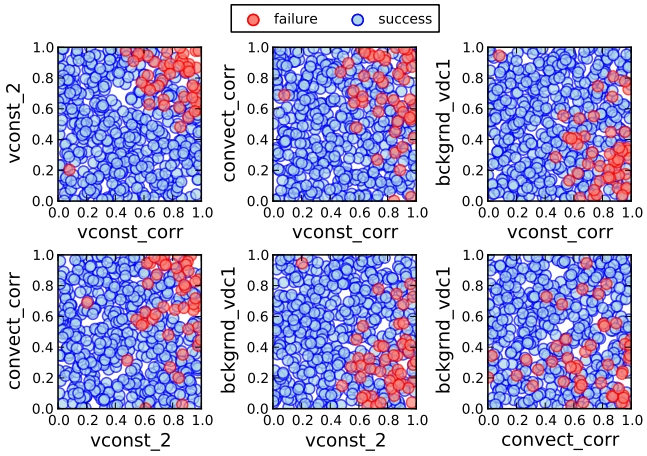

Fig. 2. Same as Fig. 1, but showing the two-dimensional projections for the same four model parameters.

incorporating observational data, performing statistical inferences, and estimating parameter values and probability distributions using maximum likelihood and Bayesian methods. Of the many capabilities provided by the UQ Pipeline, the failure analysis presented here uses the simulation parameter values and a method for calculating parameter sensitivities.

3 Descriptive failure analysis

Figures 1 and 2 show simulation successes and failures for the three Latin hypercube studies (540 runs) as a function of the values of 4 of the 18 parameters sampled in POP2. Similar figures were generated for the other parameters, but are not displayed to keep the discussion brief and because the failures are highly sensitive to changes in these parameters (see Sect. 5). It is not possible to directly visualize the dependencies in high dimensions, so the figures show the outcomes projected in one and two parameter dimensions (Figs. 1 and 2, respectively).

From the figures it is clear that the failures are generally concentrated around high values of parameters `vconst_corr` and `vconst_2`, and at low values of `backgrnd_vdc1`. A weaker dependence of the failures on high values of `convect_corr` is also apparent. The analysis presented in following sections does not require a detailed understanding of the physical reasons that connect parameter values to simulation failures, though we briefly summarize the connections to help with the interpretation.

The parameters `vconst_corr` and `vconst_2` are part of the anisotropic horizontal viscosity parameterization applied to the momentum equations in POP2 (Smith et al., 2010). Their values are subject to three main constraints, considering the physical processes and limitations to maintain numerical stability; their lower bounds are constrained by the grid's Reynolds number representing the ratio between advection and diffusion, the Munk boundary layer constraint is needed to represent western boundary currents (Jochum et al., 2008), and their upper bounds are limited by a linear diffusion stability requirement specified by a viscous Courant–Friedrichs–Lewy (CFL) condition, which depends on the integration time step (one hour in this study) and grid resolution (Griffies, 2004; Large et al., 2001). High values of these parameters may trigger the limit set by the CFL condition and is the likely reason for the model failures seen in the experiments. The `bckgrnd_vdc1` parameter is used to set the background diffusivity for diapycnal mixing from internal waves in the KPP vertical mixing parameterization

(Large et al., 1994). Reducing the values of bckgrnd_vdc1 and other bckgrnd parameters increase the numerical noise in the solution and consequently cause numerical instability. Similarly, increasing the value of convect_corr, which increases diffusivity and viscosity in the implicit KPP vertical mixing scheme, leads to instabilities in the vertical density profile. For detailed descriptions of all of the POP2 parameters used in the current study, please refer to Smith and McWilliams (2003), Large et al. (2001), and Danabasoglu et al. (2012).

In spite of the obvious relationships between the parameter values and simulation outcomes, other features present in the figures suggest that the ability to determine the causes of the failures is potentially complicated. Figure 2, for instance, indicates that there are strong correlations between failed simulations and pairs of parameter values. As one example, failures occur at the combination of high values of vconst_corr and low values of backgrnd_vdc1. These two parameters reside in different modules in POP2 (hmix_aniso, and vmix_kpp, respectively), which makes it difficult for POP2 model developers and users to discover and attribute simulation failures to correlations in these parameters.

A more important complication arises from the overlap of simulation successes and failures in the low dimensional projections shown in the figures. Some simulations appear to fail in the same general vicinity of parameter space where other simulations succeed, and vice versa. To illustrate, the upper right portion of the scatterplot between vconst_corr and vconst_2 in Fig. 2 contains a high density of failures and successes. Another notable example is the isolated failure event shown in the lower left hand corner of the same scatterplot.

These overlaps can lead to serious misclassification errors in statistical models used to predict failures as a function of parameter values. Two types of misclassification errors can occur. Simulations that are predicted to fail, but actually succeed are false positives or type I errors; those that are predicted to succeed, but actually fail are false negatives or type II errors (see Sect. 4 for further details). Imbalanced data, in which the population of one class greatly outnumbers the populations of other classes, are associated with class overlap (Prati et al., 2004), and the POP2 outcomes are highly imbalanced (i.e., 46 failures out of 540 simulations). Another related explanation is that higher parameter dimensions, and possibly a non-linear decision boundary, are required to effectively separate the outcomes.

Statistical approaches more powerful than the descriptive relationships illustrated in Figs. 1 and 2 are therefore needed to attack our problem. As described in the remaining sections, we turned to algorithms and diagnostics developed in the fields of pattern recognition, machine learning, and signal detection. These methods provided us with the ability to predict simulation failures in advance of running the model and a tool to quantify the causes of the failures. This latter capability can be used to improve POP2 by making it more robust to parameter changes.

4 Probabilistic failure classification

For a given set of model input parameters, a POP2 simulation will either succeed or fail. We denote these outcomes by a two-class categorical variable in which failures belong to class $\mathcal{C}_\mathrm{f}$ and successes belong to class $\mathcal{C}_\mathrm{s}$. The present discussion considers only a single failure class, but we recognize that simulations can fail for a variety of reasons (e.g., lack of iterative convergence, numerical instabilities, etc). Without difficulty, the two-class methodology described below can be extended to handle multiple modes of failure through multi-class classification.

Our goal for probabilistic failure classification is to determine the probability that a POP2 simulation will fail for a vector of model input parameters $\boldsymbol{x} = (x_1, x_2, \ldots, x_{18})$. We denote this using the conditional probability $\mathcal{P}(\mathcal{C}_\mathrm{f}|\boldsymbol{x})$. Using Bayes' rule, the posterior conditional probability can be written

$$\mathcal{P}(\mathcal{C}_\mathrm{f}|\boldsymbol{x}) = \frac{\mathcal{P}(\boldsymbol{x}|\mathcal{C}_\mathrm{f})\mathcal{P}(\mathcal{C}_\mathrm{f})}{\mathcal{P}(\boldsymbol{x}|\mathcal{C}_\mathrm{f})\mathcal{P}(\mathcal{C}_\mathrm{f}) + \mathcal{P}(\boldsymbol{x}|\mathcal{C}_\mathrm{s})\mathcal{P}(\mathcal{C}_\mathrm{s})}, \quad (1)$$

where $\mathcal{P}(\boldsymbol{x}|\mathcal{C}_i)$ and $\mathcal{P}(\mathcal{C}_i)$ correspond to class-conditional densities and class priors, respectively. By introducing a variable λ representing the natural logarithm of the likelihood-odds ratio,

$$\lambda = \ln\left[\frac{\mathcal{P}(\boldsymbol{x}|\mathcal{C}_\mathrm{f})}{\mathcal{P}(\boldsymbol{x}|\mathcal{C}_\mathrm{s})}\,\frac{\mathcal{P}(\mathcal{C}_\mathrm{f})}{\mathcal{P}(\mathcal{C}_\mathrm{s})}\right], \quad (2)$$

Eq. (1) can be rewritten as the "S-shaped" logistic sigmoid function

$$\mathcal{P}(\mathcal{C}_\mathrm{f}|\boldsymbol{x}) = \frac{1}{1+\exp(-\lambda)}. \quad (3)$$

The λ term is a function of $\boldsymbol{x}$ and takes values in $(-\infty, \infty)$. As illustrated in Fig. 3, the sigmoid function is bounded between 0 and 1, inclusive. This formalism provides a mechanism to transform an input vector of model parameter values to a probability that the corresponding simulation will fail or succeed.

It is also useful to mention that the methods presented here can be applied to a much broader set of problems. Any geoscientific model output that varies continuously is amenable to probabilistic failure analysis by thresholding or discretizing the output, and then classifying the "failures" and "successes" as cases that fall on different sides of the threshold or fall within different bins. For instance, if a 5 K difference in global average surface temperature between a climate model simulation and a reference case is deemed excessive, then climate model instances above and below this threshold can be categorized as failures and successes, respectively. Equation (3) would then provide a probability that a new model

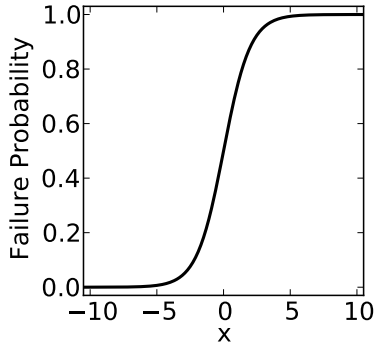

Fig. 3. Logistic sigmoid function defined in Eq. (3) with $\lambda(x) = x$.

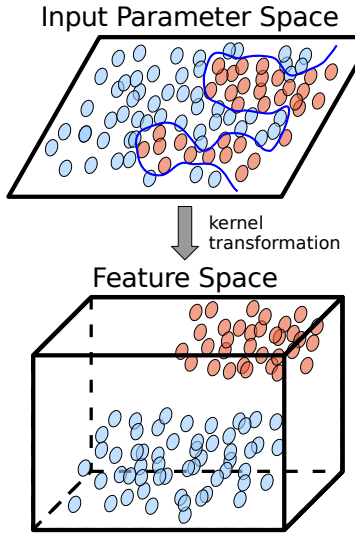

Fig. 4. Conceptual image showing the separability of the red and blue classes through kernel transformations in SVMs.

case would "fail" by simulating temperatures that exceed the 5 K threshold, information that developers could use to improve their models.

4.1 SVM classification

Support vector machine (SVM) classification (Vapnik, 1995; Cortes and Vapnik, 1995; Burges, 1998) from the fields of pattern recognition and supervised machine learning (Bishop, 2007; Kotsiantis, 2007) is used to assign a simulation to class $\mathcal{C}_f$ or $\mathcal{C}_s$ for input vector $\boldsymbol{x}$. This type of classification problem can also be handled using other methods, such as logistic regression (Hosmer and Lemeshow, 2000), neural networks (Bishop, 2007), decision trees (Breiman et al., 1984), and random forests (Breiman, 2001). We limit our attention to SVMs, however, given our familiarity with the algorithm and its excellent performance on our climate model application.

Briefly, the SVM method is based on maximizing the distance between parallel hyperplanes that separate the classes (i.e., the margin), while allowing for misclassifications from overlapping data points during training (i.e., a soft margin). For non-linearly separable classes, the hyperplanes are determined by transforming the input space to a higher-dimensional feature space using kernel functions. The purpose of the transformation is to make it easier to separate the classes, as illustrated conceptually in Fig. 4. The support vectors are the training points that lie on the margin for classes that are separable, and lie on or within the margin for classes that are not. New input vectors $\boldsymbol{x}$ are assigned to a class using the sign of the predictive decision function:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_s} y_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b, \tag{4}$$

where $f(\boldsymbol{x}) > 0$ and $f(\boldsymbol{x}) < 0$ are assigned to classes $\mathcal{C}_f$ and $\mathcal{C}_s$, respectively. The sum in Eq. (4) is over the N_s support vectors from the training set, $y_i \in \{-1, 1\}$ is a binary outcome indicator variable, $K(\boldsymbol{x}_i, \boldsymbol{x})$ is the kernel function, and b and α_i are, respectively, bias and Lagrange multiplier terms determined through constrained optimization of the margin. Refer to Burges (1998), Bishop (2007), or Chang and Lin (2011) for further details.

The decision function in Eq. (4) assigns inputs to a class, but does not provide a probability of class membership. An extension to the standard SVM approach was therefore developed (Platt, 1999) that derives class probabilities by fitting λ in Eq. (2) to a two parameter function using the training data and cross validation. A variation of this procedure is implemented in the `LIBSVM` package (Chang and Lin, 2011), which we used to fit SVM classifiers that calculate the failure probabilities $\mathcal{P}(\mathcal{C}_f|\boldsymbol{x})$.

As described in more detail in Sect. 4.3, we also applied an ensemble learning approach (Dietterich, 2000) to create a "committee" of SVM classifiers. Each classifier in the committee contributes a vote (i.e., failure probability), and the votes are tallied in different ways to predict simulation failures and to assess the performance of the classification system.

4.2 Classification performance

Using a format known as the "confusion matrix" in machine learning, Fig. 5 summarizes the four possible outcomes for the two-class simulation failure problem. Classifiers that correctly predict actual failures and successes are labeled true positives (TP) and true negatives (TN), respectively; those that incorrectly predict actual failures and successes are denoted false negatives (FN) and false positives (FP), respectively. Of the numerous ways to combine these outcomes to assess classifier performance, we focus on the true positive rate (TPR) and false positive rate (FPR), which are given by the expressions

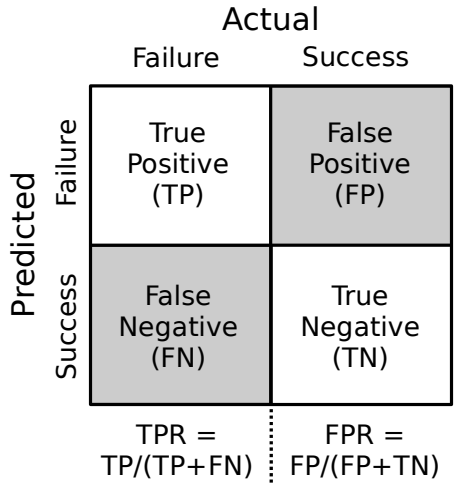

Fig. 5. The confusion matrix showing the four possible outcomes for a two-class simulation failure problem.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{5}$$

and

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{6}$$

Perfect classifiers have TPR and FPR values of 1 and 0, respectively. As noted previously, we use SVM classifiers that provide probabilities of class membership. The assignment to a particular class, and resulting TPR and FPR values, therefore depends upon a specified decision variable and threshold value. If decisions are made using the failure probability with a threshold of 0.5, for example, then probabilities above and below this threshold will be assigned to classes $\mathcal{C}_\mathrm{f}$ and $\mathcal{C}_\mathrm{s}$, respectively.

The quantities in Eqs. (5) and (6) are combined into a convenient diagram used in signal detection and decision analysis known as a receiver operating characteristic (ROC) curve (Swets, 1988; Fawcett, 2006). ROC curves plot the FPR (horizontal axis) versus TPR (vertical axis) of a decision variable as the threshold is varied from $+\infty$ to $-\infty$. A perfect classifier is represented in ROC space by the vertical line connecting points $(0, 0)$ and $(0, 1)$, followed by the horizontal line connecting points $(0, 1)$ and $(1, 1)$. A classifier that makes random assignments, on the other hand, is represented by the diagonal line connecting points $(0, 0)$ and $(1, 1)$. The predictive capability of a classification system can therefore be assessed by a single number, the area under the ROC curve (AUC) (e.g., Marzban, 2004). As a rough rule of thumb, a classifier with an AUC score of about 0.8 or higher is useful for discrimination. The AUC score is used in following sections to train SVM classifiers and to test their performance on independent simulation failure data.

Table 3. Predictions and outcomes of simulation crashes in study 3.

Run	μ_c	σ_c	Predicted*			Actual
			D_avg	D_sum	D_snr	
002	0.47	0.13	Success	Failure	Success	Success
006	0.54	0.14	Failure	Failure	Failure	Failure
015	0.37	0.10	Success	Success	Failure	Success
017	0.42	0.12	Success	Failure	Failure	Failure
027	0.25	0.09	Success	Success	Success	Failure
044	0.04	0.02	Success	Success	Success	Failure
060	0.80	0.10	Failure	Failure	Failure	Failure
073	0.52	0.15	Failure	Failure	Failure	Failure
088	0.63	0.11	Failure	Failure	Failure	Failure
095	0.47	0.15	Success	Failure	Success	Success
097	0.83	0.09	Failure	Failure	Failure	Failure
120	0.49	0.13	Success	Failure	Failure	Failure
141	0.88	0.09	Failure	Failure	Failure	Success
148	0.76	0.12	Failure	Failure	Failure	Failure
155	0.31	0.08	Success	Success	Failure	Failure
166	0.64	0.11	Failure	Failure	Failure	Failure
173	0.75	0.12	Failure	Failure	Failure	Failure
177	0.67	0.14	Failure	Failure	Failure	Failure

* Actual successes predicted by all decision criteria are not reported here for the sake of brevity. Predictions from D_avg and D_sum were made before the study, while those from D_snr were determined retrospectively.

4.3 Supervised learning of simulation failures

The three UQ studies listed in Table 2 were performed in succession to one another. After completing studies 1 and 2, but before starting study 3, we trained a committee of SVM classifiers to learn about the simulation failures. The goal was to use the committee to predict the outcomes of study 3 before those simulations even began to run. This section describes the training procedure, while the following two sections describe the performance on study 3.

The training set consisted of the 360 simulations from studies 1 and 2, which had 32 simulation failures and 328 successes. Given the relatively small ratio of the number of failure events to the number of classifier inputs (i.e., 32 / 18), we utilized an ensemble learning approach (Dietterich, 2000) known as bootstrap aggregation (i.e., "bagging") (Breiman, 1996). Bagging quantifies the variability of classifier predictions and can help improve the overall classification performance. The bagging was applied by resampling the training data $N_\mathrm{b} = 100$ times, allowing for duplicates in the samples (i.e., sampling with replacement). This step effectively created 100 different versions of the training data that were used to construct the committee of individual SVM classifiers. Failure predictions were then made by aggregating the votes across the committee, using an equal weight for each classifier. In particular, we computed the mean value (μ_c) and standard deviation (σ_c) of the committee,

$$\mu_c = \frac{1}{N_b}\sum_{i=1}^{N_b} \mathcal{P}_i(\mathcal{C}_f|\boldsymbol{x}) \tag{7}$$

and

$$\sigma_c^2 = \frac{1}{N_b}\sum_{i=1}^{N_b} [\mathcal{P}_i(\mathcal{C}_f|\boldsymbol{x}) - \mu_c]^2, \tag{8}$$

and then combined these quantities in different ways to form decision variables for prediction and ROC analysis (see Sects. 4.4 and 4.5).

The LIBSVM package (Chang and Lin, 2011), which is freely available and open source, was used to train each of the classifiers in the committee. LIBSVM offers two versions of SVM classification (C-support and ν-support) and four standard types of kernel functions (linear, polynomial, Gaussian, and hyperbolic tangent). On the basis of familiarity and experience, we employed C-support classification with Gaussian kernels, $K(\boldsymbol{x}_i, \boldsymbol{x}) = \exp(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}\|^2)$, though we subsequently tested other kernels (see below). The values of two adjustable SVM-related parameters, the kernel width γ and misclassification penalty C, were determined using a cross validation method (Arlot and Celisse, 2010). For each bootstrap replicate of the training dataset, we randomly selected 80 % of the data to construct an individual classifier and used the remaining 20 % to test that classifier. The objective was to find values for γ and C that are the same for all of the classifiers and that maximize their performance on these held-out tests. This task was accomplished by combining the individual tests into a large cross validation test set with 7200 data instances (0.2 test fraction $\times$360 simulations $\times$100 resampling size), and then computing the ROC curve and AUC score on this data. Figure 6 shows the ROC curve using the values that maximized the AUC ($\gamma = 0.1$, $C = 3$, and AUC = 0.93). The area under the curve is well above 0.8, which indicated that the SVM committee could be used for predicting simulation crashes in study 3.

The analysis presented throughout the manuscript utilized only the Gaussian kernels. However, to test the sensitivity to the choice of SVM kernel, we subsequently re-trained the classifiers using the same training data and cross validation technique, but instead applied linear, cubic, and hyperbolic tangent kernel functions. The Gaussian kernels performed slightly better than the other kernels, but all of the kernels still achieved cross validation AUC scores above 0.92. This test suggests that the $\mathcal{C}_f$ and $\mathcal{C}_s$ classes are primarily linearly separable, because the linear kernel performed nearly as well as the nonlinear kernels.

4.4 Predicting simulation failures

Before running the 180 simulations in study 3, we used the SVM classifier committee trained from studies 1 and 2 to predict simulation failures in study 3. These predictions were sent out to group members by email at the beginning of the study (see fourth and fifth columns in Table 3) and were largely validated by the end of the study. The predictions were based on Eqs. (7) and (8). Simulations were assigned to the $\mathcal{C}_f$ class using decision criteria denoted by

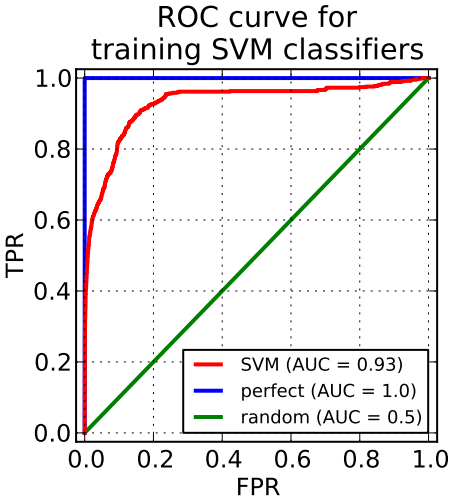

Fig. 6. Receiver operating characteristic for the bootstrapped set of individual SVM classifiers assessed using holdout test data. SVM training parameters ($\gamma = 0.1$, $C = 3$) are chosen to maximize the area under the ROC curve.

$$D \equiv \text{decision variable} \geq \text{threshold}. \tag{9}$$

Two initial criteria were selected using the same threshold, but different decision variables. The first criterion used the committee average,

$$D_{avg} \equiv \mu_c \geq 0.5, \tag{10}$$

while the second used the sum of the committee average and standard deviation,

$$D_{sum} \equiv \mu_c + \sigma_c \geq 0.5. \tag{11}$$

The second criterion was chosen to account for variability across committee members by categorizing some simulations as $\mathcal{C}_f$ even though they had a committee mean below 0.5. After all of the simulations were completed, a third criterion based on the signal-to-noise ratio from the committee, D_{snr}, was also considered and analyzed (see Sect. 4.5).

The predictions and actual outcomes are summarized in Table 3 and displayed in Figs. 7 and 8. As noted in Table 2, there were 14 actual simulation failures and 166 successes in the study. The classifier committee performed exceedingly well using the two initial criteria, D_{avg} and D_{sum}. Referring to the confusion matrix in Fig. 7, both made 174 correct predictions (TP + TN, 96.7 % accuracy) and only six incorrect predictions (FP + FN). The incorrect predictions, however, are distributed differently for the two criteria. D_{avg} had more

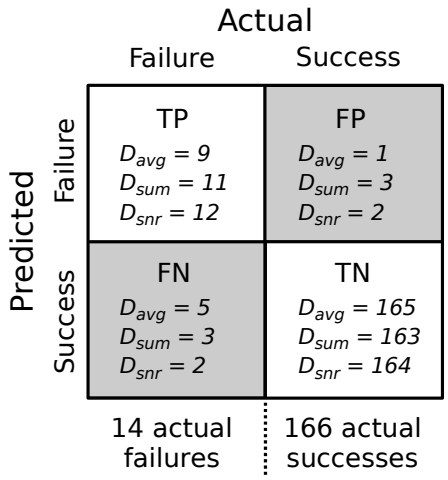

Fig. 7. The confusion matrix for predictions of 180 simulations in study 3 using the SVM committee with three different decision criteria (D_{avg}, D_{sum}, and D_{snr}).

FNs than FPs, while D_{sum} had an equal number of each. Because of this difference, D_{avg} and D_{sum} operate at different points in ROC space. D_{avg} has ROC coordinates of (1/166, 9/14), while D_{sum} operates at (3/166, 11/14). Based on their Euclidean distance from a perfect classifier, which is given by $\left[\mathrm{FPR}^2 + (\mathrm{TPR} - 1)^2\right]^{1/2}$, we conclude that D_{sum} (distance = 0.215) performs better than D_{avg} (distance = 0.357).

To ascertain the cause of the performance difference between D_{sum} and D_{avg}, the top and middle panels in Fig. 8 display the actual outcomes and predictions using the μ_{c} and $\mu_{\mathrm{c}} + \sigma_{\mathrm{c}}$ decision variables for the runs in study 3. The decision criteria are represented by the horizontal lines in the panels. Runs that are on or above the lines were predicted to fail, while those below were predicted to succeed. Correct predictions are displayed in blue (TP and TN), and incorrect predictions in red (FP and FN). The figure indicates, for example, that runs 17 and 120 failed, but were misclassified by D_{avg} because their μ_{c} values were slightly below 0.5. By comparison D_{sum} assigned these runs to the correct class, but also misdiagnosed runs 2 and 95. A visual inspection of the figure shows that, except for the relative position of the threshold, the distribution of points in μ_{c} and $\mu_{\mathrm{c}} + \sigma_{\mathrm{c}}$ look very similar. We therefore attribute the performance difference to the threshold value. If D_{avg} had used a threshold value of about 0.4 instead of 0.5, it would have made the same predictions and had the same performance as D_{sum}.

At this point, we can also compare the predictive performance of our classification system to the "second design" predictions in Edwards et al. (2011) (see Sect. 4.3 therein). We do not compare to their "first design" results (i.e., the table of Sect. 4.2 therein), because those results use the same data to both train and evaluate their statistical model (i.e., they are not predictions). Moreover, Edwards et al. (2011) provide only TN and FN values for this design because they screened out simulations that were predicted to fail. Their model incorrectly classified 26 % of the predicted successes, which we calculate using FN/(FN + TN). By comparison, our system misclassified only 3 and 2 % of the predicted successes using D_{avg} and D_{sum}, respectively.

4.5 Retrospective analysis of simulation failures

After study 3 was completed, we applied the same SVM committee to test the performance of an additional criterion based on the signal-to-noise ratio, $\mu_{\mathrm{c}}/\sigma_{\mathrm{c}}$, as the decision variable. Without prior knowledge about a setting for the threshold for this criterion, it was not used to predict simulation failures in advance. Retrospectively, we used the same values of μ_{c} and σ_{c} that were used for the predictions in Sect. 4.4, but determined and tested a setting for the threshold that maximizes the overall accuracy and minimizes the total number of false outcomes (FP + FN). The resulting criterion is defined by

$$D_{\mathrm{snr}} \equiv \mu_{\mathrm{c}}/\sigma_{\mathrm{c}} \geq 3.53. \quad (12)$$

The performance of this criterion is displayed in Table 3 and Figs. 7 and 8. As shown, D_{snr} outperforms both D_{avg} and D_{sum}. If included in our set of predictions, this criterion would have made 176 correct predictions (97.8 % accuracy) and only four false predictions balanced between two FNs (runs 27 and 44) and two FPs (runs 15 and 141). D_{snr} operates at ROC point (2/166, 12/14), which is a distance of 0.143 from a perfect classifier. The reason for the improved performance is shown more clearly in Fig. 8. The signal-to-noise ratio better separates the failures and successes than either of the other decision variables, although runs 44 and 141 are still grossly misclassified. In spite of the improvement, it is also worth noting that more simulations lie closer to D_{snr} than either D_{avg} or D_{sum}. This implies that the performance of D_{snr} is more sensitive to slight adjustments in the value of the threshold than the other criteria.

In retrospect, we also varied the thresholds for the three decision variables and calculated the FPRs and TPRs for study 3. The resulting ROC curves and fixed locations of the decision criteria are shown in Fig. 9. The ROC curves for μ_{c} and $\mu_{\mathrm{c}} + \sigma_{\mathrm{c}}$ nearly overlap, which confirms the previous statement that these two decision variables perform similarly after accounting for threshold differences. Based on their AUC scores, μ_{c} performs marginally better than $\mu_{\mathrm{c}} + \sigma_{\mathrm{c}}$ because adding committee variability causes some successful simulations to get tallied relatively sooner as FPs (see points with values close to run 27 in Fig. 8). In contrast to these cases, the ROC curve for $\mu_{\mathrm{c}}/\sigma_{\mathrm{c}}$ is noticeably better and has an AUC of 0.966. This occurs because $\mu_{\mathrm{c}}/\sigma_{\mathrm{c}}$ is more effective at separating the classes, which enables it to identify more TPs as the threshold is lowered. Overall, however, all three decision variables perform exceedingly well at classifying new simulation failures. Using these ROC curves, we can choose decision criteria for making new predictions that

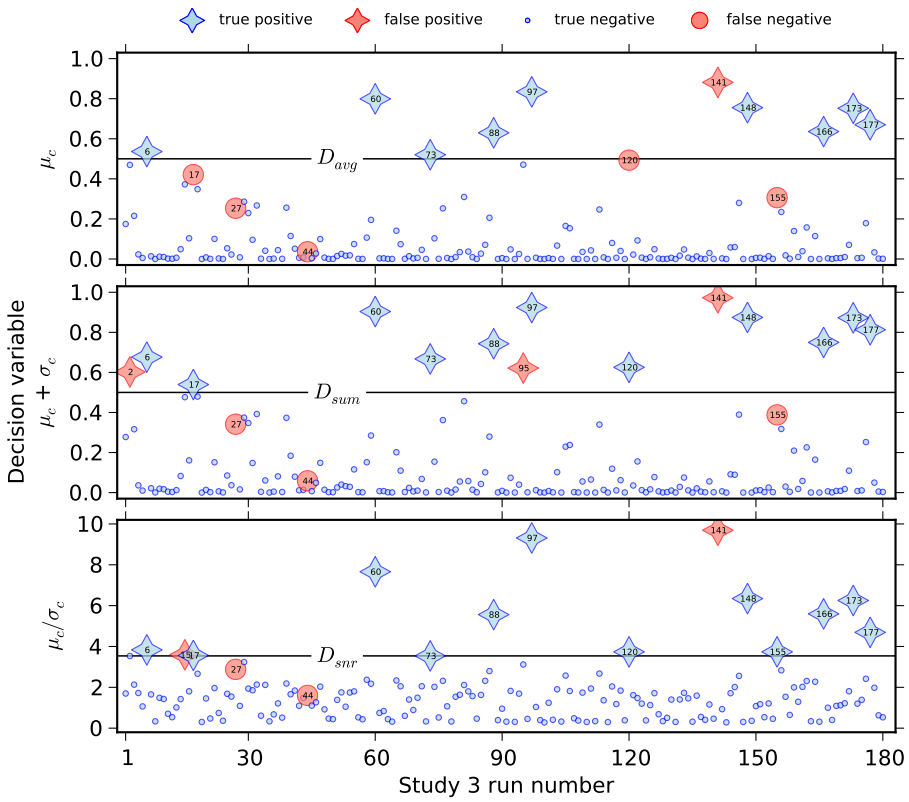

Fig. 8. Actual and predicted outcomes are shown for the 180 simulations in study 3. Predictions are based on three decision variables and thresholds (μ_c and D_{avg}, top; $\mu_c + \sigma_c$ and D_{sum}, middle; μ_c/σ_c and D_{snr}, bottom). The horizontal axis displays simulation numbers based on their order in the ensemble. Simulation failures and successes are shown using stars and circles, respectively. Correct and incorrect predictions are shown in blue and red, respectively.

consider the tradeoffs between TPs and FPs. Slightly lowering the threshold in D_{snr}, for example, will increase the TPR and move it to a point that lies closer to a perfect classifier in ROC space, but this occurs at the expense of also increasing the FPR.

5 Sensitivity analysis of simulation failures

Following on the demonstrated success of our predictions, we used the classifier committee to identify, quantify, and rank the importance of the model parameters responsible for the simulation failures. This information can be used to make the model more robust to parameter perturbations by improving the modules associated with the most sensitive parameters. For this analysis, we drew 10^4 Latin hypercube samples from uniform distributions representing the 18 POP2 parameters, calculated the average failure probability from a committee of SVM classifiers (μ_c) at each of the sample points, and then performed a global sensitivity analysis (Saltelli et al., 2000; Helton et al., 2006) on the parameter-induced variance of log μ_c. All of the available simulation data were used to compute the parameter sensitivities by re-training a new committee of 100 SVM classifiers with the full set of 540 simulations from studies 1–3. The training followed the procedure previously described in Sect. 4.3. Also note that the sensitivity analysis is illustrated below using μ_c as the committee response, but the same general results are obtained using the signal-to-noise ratio (μ_c/σ_c).

5.1 Polynomial chaos expansion of the failure probability

Parameter sensitivities were measured and ranked using Sobol indices (Sobol, 2001; Saltelli et al., 2000), which decompose the variance of log μ_c into contributions from individual parameters and various higher-order combinations of parameters. Polynomial chaos expansions (Wiener, 1938) provide a convenient format for the sensitivity analysis because the squares of the expansion coefficients are directly proportional to Sobol indices (Sudret, 2008; Lucas and Prinn, 2005; Tatang et al., 1997). The distribution of log μ_c was fit to $N_p = 18$ parameters using a second-order polynomial chaos expansion expressed as

$$\log\ \mu_c = a_0 + \sum_{i=1}^{N_p} [b_i P_1(\xi_i) + c_i P_2(\xi_i)] + \sum_{i=1}^{N_p-1} \sum_{j=i+1}^{N_p} d_{ij} P_1(\xi_i) P_1(\xi_j), \quad (13)$$

where ξ_i is the random variable representation of parameter i, $P_n(\xi_i)$ is an n-th order orthogonal polynomial in ξ_i, and the a_0, b_i, c_i and d_{ij} are expansion coefficients to be determined. For the case where the ξ_i are standard uniform random variables, the $P_n(\xi_i)$ are the shifted Legendre polynomials (see Xiu and Karnidakis, 2002) with the following orthogonality property:

$$\int_0^1 P_m(\xi_i) P_n(\xi_i) \mathrm{d}\xi_i = \frac{1}{2n+1} \delta_{mn}, \quad (14)$$

where δ_{mn} is the Kronecker delta function. The first and second order shifted Legendre polynomials are given by

$$P_1(\xi_i) = 2\xi_i - 1 \quad (15)$$

and

$$P_2(\xi_i) = 6\xi_i^2 - 6\xi_i + 1. \quad (16)$$

The coefficients in Eq. (13) were determined through least squares, and higher-order terms were not considered because the second-order expansion fits the data very well (adjusted $R^2 = 0.98$). The resulting fit is given in Table 4, which shows the leading terms of the expansion in two forms.

Analytical expressions for the moments of log μ_c as a function of the POP2 parameters were derived by directly taking expectation values of Eq. (13). The average value and variance are

$$\mathrm{avg}(\log\ \mu_c) = a_0, \quad (17)$$

and

$$\mathrm{var}(\log\ \mu_c) = \overbrace{\sum_{i=1}^{N_p} \left(\frac{b_i^2}{3} + \frac{c_i^2}{5} \right)}^{\text{individual parameters}} + \overbrace{\sum_{i=1}^{N_p-1} \sum_{j=i+1}^{N_p} \frac{d_{ij}^2}{9}}^{\text{pairs of parameters}}. \quad (18)$$

The two groups of terms labeled on the right hand side of Eq. (18) signify variance contributions from individual parameters (linear and quadratic) and pairs of parameters. The fractional values of the squared polynomial chaos expansion coefficients in Eq. (18) follow from application of Eq. (14).

5.2 Sensitivity network of the failure probability

Given a parameter-based decomposition of the variance, we have developed a technique to visualize complex variance

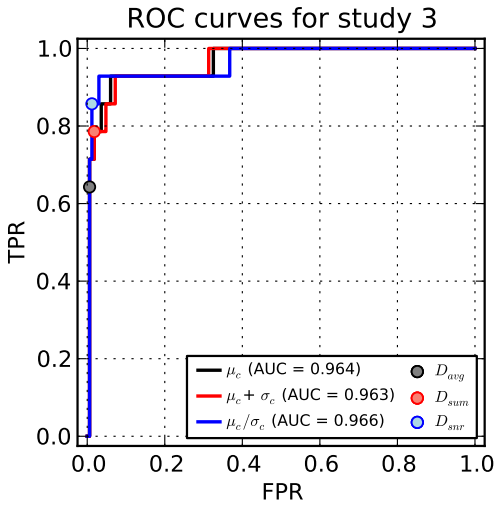

Fig. 9. ROC curves for the 180 simulations in study 3 using the SVM committee with three decision variables (μ_c, $\mu_c + \sigma_c$, and μ_c/σ_c). The locations of the discriminators using the fixed thresholds in D_{avg}, D_{sum}, and D_{snr} are also shown.

connections using network graphs with nodes and edges. The size of node$_i$ in the graph is proportional to the fractional contribution from parameter i,

$$\mathrm{node}_i \propto \frac{b_i^2/3 + c_i^2/5}{\mathrm{var}(\log\ \mu_c)}, \quad (19)$$

while the thickness of edge$_{ij}$ connecting node$_i$ and node$_j$ is proportional to the fractional contribution from joint variations of parameters i and j,

$$\mathrm{edge}_{ij} \propto \frac{d_{ij}^2/9}{\mathrm{var}(\log\ \mu_c)}. \quad (20)$$

The technique has been extended to include higher order effects (e.g., using edge$_{ijk}$ for 3rd-order terms), but this is not needed for the current application. Important parameters on the resulting network graph are represented by nodes that are large or make significant connections to other nodes.

Figure 10 displays the network graph for the variance decomposition of log μ_c. Based on node size and connectivity, the graph indicates that 8 out of the 18 parameters are the main drivers of the simulation failures (see parameters labeled in red in the graph). These eight parameters account for about 95 % of the variance of log μ_c, as quantified using Eq. (18). Of these, `vconst_corr`, `vconst_2`, `convect_corr`, and `bckgrnd_vdc1` stand out distinctly as the top four parameters in the graph. Recall that the same four parameters are described in Sect. 3 and displayed in Figs. 1 and 2. The top four parameters have the largest overall and most heavily connected nodes in the graph, and they collectively account for about 88 % of the variance of log μ_c. The strong connections indicate that the probability of simulation failure depends on correlations between the

Table 4. Polynomial chaos expansion of failure probability.

Expansion	Leading terms*
ξ_i	$\log \mu_c \approx -4.347 + 4.049\ \xi_1 + 3.400\ \xi_2 + 2.267\ \xi_{13} - 1.980\ \xi_{14} - 1.393\ \xi_{16} - 1.253\ \xi_5 - 1.143\ \xi_4 - 1.007\ \xi_{17} - 0.885\ \xi_2\ \xi_1 - 0.796\ \xi_{13}\ \xi_1 - 0.739\ \xi_6 - 0.637\ \xi_{13}\ \xi_2 - 0.610\ \xi_9 + 0.578\ \xi_{14}\ \xi_2 + 0.480\ \xi_{16}\ \xi_2 - 0.471\ \xi_{15} - 0.414\ \xi_1^2 + 0.382\ \xi_5\ \xi_1 + 0.372\ \xi_{14}\ \xi_1 + 0.351\ \xi_{17}\ \xi_2 + 0.320\ \xi_2\ \xi_5 + 0.320\ \xi_8^2 + \ldots$
$P_n(\xi_i)$	$\log \mu_c \approx -2.609 + 1.628 P_1(\xi_1) + 1.546 P_1(\xi_2) + 1.061 P_1(\xi_{13}) - 0.895 P_1(\xi_{14}) - 0.475 P_1(\xi_5) - 0.455 P_1(\xi_{16}) - 0.338 P_1(\xi_4) - 0.311 P_1(\xi_{17}) - 0.245 P_1(\xi_9) - 0.221 P_1(\xi_1) P_1(\xi_2) - 0.199 P_1(\xi_1) P_1(\xi_{13}) + 0.196 P_1(\xi_{12}) + 0.174 P_1(\xi_{10}) + 0.164 P_1(\xi_{11}) - 0.159 P_1(\xi_2) P_1(\xi_{13}) + 0.145 P_1(\xi_2) P_1(\xi_{14}) + 0.133 P_1(\xi_{18}) + 0.120 P_1(\xi_2) P_1(\xi_{16}) + 0.096 P_1(\xi_1) P_1(\xi_5) + 0.093 P_1(\xi_1) P_1(\xi_{14}) + 0.088 P_1(\xi_2) P_1(\xi_{17}) - 0.082 P_1(\xi_6) + \ldots$

* Leading terms are based on the magnitude of the absolute value of the coefficients of the polynomial chaos expansion. Refer to Table 1 for the parameter labels that correspond to the numbers.

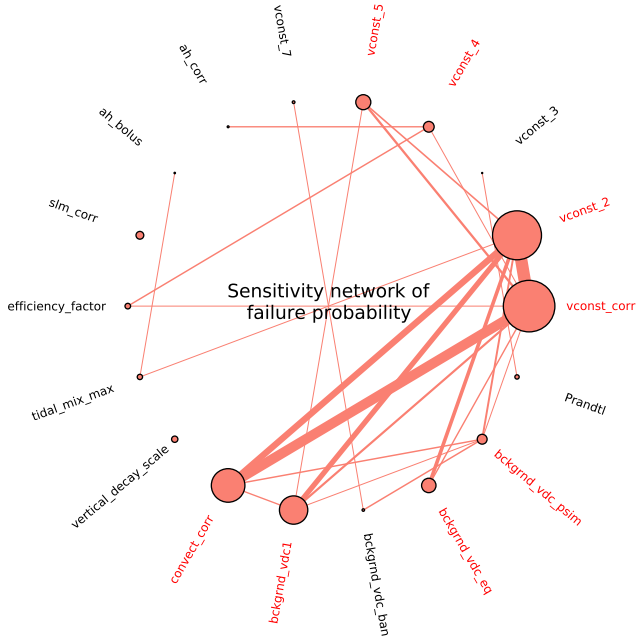

Fig. 10. Sensitivity of the probability of simulation failure to climate model parameters is shown using a network graph. Node size and connector thickness are proportional to sensitivity contributions from individual parameters and pairs of parameters, respectively. The eight parameters labeled in red are the main causes of simulation failures.

top four parameters. The direction of the dependence is determined by inspecting the signs of the corresponding coefficients in the polynomial chaos expansion (i.e., for ξ_1, ξ_2, ξ_{13}, and ξ_{14}). Referring to Table 4, the failure probability increases for increasing values of `vconst_corr`, `vconst_2`, and `convect_corr`, and increases for decreasing values of `bckgrnd_vdc1`, which is in accordance with the results in Figs. 1 and 2.

The variance decomposition therefore validates the descriptive relationships given in Sect. 3. However, it also extends the failure analysis in important ways. Equation (18) quantitatively ranks the effects of the parameters on the simulation failures, which provides a way to prioritize efforts to improve the model. This type of ranking cannot be easily obtained using just the scatterplots in Figs. 1 and 2. Moreover, the scatterplots show the correlations between the parameter values and simulation failures, but the one and two dimensional projections are not sufficient for separating the overlapping $\mathcal{C}_f$ and $\mathcal{C}_s$ classes. Figure 10, on the other hand, very clearly shows that four or more parameter dimensions are required to explain and separate the simulation failures from the successes.

6 Summary and conclusions

We experienced a series of code crashes while running Latin hypercube ensemble simulations that sampled the values of 18 ocean mixing and viscosity parameters in the POP2 component of CCSM4. The crashes occurred for numerical reasons at different combinations of parameter values, which we surmise is due to violations of numerical conditions defined in the model (e.g., CFL as described in Sect. 3). Assuming no special knowledge or physical insight about the specific nature of the crashes, we formulated the simulation crashes as a binary problem (i.e., they fail or succeed) and used machine learning classification to quantify failure probabilities as a function of the 18 model parameters. A highly predictive SVM classification system was trained from a dataset containing only 32 failure instances out of 360 simulations and validated using an independent set of 180 simulations. The resulting classification system had a prediction AUC score exceeding 0.96 and achieved discrimination accuracies above 97 %. Global sensitivity analysis was then used to identify eight model parameters from four different modules that drive high probabilities of failing, the results of which can be used to increase the robustness of CCSM4 to parameter perturbations. These methods can be used to characterize simulation failures in other complex scientific computer

models. The climate ensemble failure dataset used for all of the analysis presented in this manuscript is being made available for public download at three sites (Bache and Lichman, 2013; MLdata.org, 2013; Lawrence Livermore National Laboratory Green Data Oasis, 2013).

Acknowledgements. Computing support for this work came from the Lawrence Livermore National Laboratory (LLNL) Institutional Computing Grand Challenge program. We thank G. Danabasoglu, M. Jochum, and R. Tokmakian for recommending the ocean model parameters and their uncertainty ranges. This work was performed under the auspices of the US Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, was funded by LLNL's Uncertainty Quantification Strategic Initiative Laboratory Directed Research and Development Project under tracking code 10-SI-013, and is released under UCRL number LLNL-JRNL-608873. D. D. L. also received support from the DOE Office of Science through the Scientific Discovery Through Advanced Computing (SciDAC) project on Multiscale Methods for Accurate, Efficient, and Scale-Aware Models of the Earth System.

Edited by: J. Annan

References

Annan, J. D., Hargreaves, J. C., Edwards, N. R., and Marsh, R.: Parameter estimation in an intermediate complexity earth system model using an ensemble Kalman filter, Ocean Model., 8, 135–154, doi:10.1016/j.ocemod.2003.12.004, 2005.

Arlot, S. and Celisse, A.: A survey of cross-validation procedures for model selection, Statistics Surveys, 4, 40–79, 2010.

Bache, K. and Lichman, M.: UCI Machine Learning Repository, available at: http://archive.ics.uci.edu/ml (last access: 30 May 2013), archived on 12 September 2012: http://www.webcitation.org/6AcuZgrsy, 2013.

Bishop, C. M.: Pattern Recognition and Machine Learning, Information Science and Statistics, 1st Edn., Springer, 2007.

Breiman, L.: Bagging predictors, Mach. Learn., 24, 123–140, 1996.

Breiman, L.: Random forests, Mach. Learn., 45, 5–32, 2001.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, P. J.: Classification and Regression Trees, 1st Edn., Chapman and Hall/CRC, 1984.

Burges, C. J. C.: A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discov., 2, 121–167, 1998.

CCSM4: The Community Climate System Model, Version 4, available at: http://www.cesm.ucar.edu/models/ccsm4.0/ (last access: 22 January 2013), archived on 21 July 2010: http://www.webcitation.org/5rOj1F8rL, 2012.

Chang, C. C. and Lin, C. J.: LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol., 2, 27, doi:10.1145/1961189.1961199, 2011.

Clune, T. and Rood, R.: Software testing and verification in climate model development, IEEE Software, 28, 49–55, 2011.

Cortes, C. and Vapnik, V.: Support-vector networks, Mach. Learn., 20, 273–297, 1995.

Danabasoglu, G., Bates, S., Briegleb, B., Jayne, S., Jochum, M., Large, W., Peacock, S., and Yeager, S.: The CCSM4 ocean component, J. Climate, 25, 1361–1389, 2012.

Dietterich, T. G.: Ensemble methods in machine learning, in: Multiple Classifier Systems, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1857, 1–15, 2000.

D'Silva, V., Kroening, D., and Weissenbacher, G.: A survey of automated techniques for formal software verification, IEEE T. Comput.-Aid. D., 27, 1165–1178, 2008.

Easterbrook, S. M.: Climate change: a grand software challenge, in: FoSER, edited by: Roman, G.-C. and Sullivan, K. J., 99–104, ACM, 2010.

Easterbrook, S. M.: Do Over or Make Do? Climate Models as a Software Development Challenge, 2012 Fall Meeting Abstracts, IN14B-01, AGU, San Francisco, Calif., 3–7 December, 2012.

Easterbrook, S. M. and Johns, T. C.: Engineering the software for understanding climate change, Comput. Sci. Eng., 11, 65–74, 2009.

Easterbrook, S. M., Edwards, P. N., Balaji, V., and Budich, R.: Guest editors' introduction: climate change – science and software, IEEE Software, 28, 32–35, 2011.

Edwards, N. R., Cameron, D., and Rougier, J.: Precalibrating an intermediate complexity climate model, Clim. Dynam., 37, 1469–1482, doi:10.1007/s00382-010-0921-0, 2011.

Farrell, P. E., Piggott, M. D., Gorman, G. J., Ham, D. A., Wilson, C. R., and Bond, T. M.: Automated continuous verification for numerical simulation, Geosci. Model Dev., 4, 435–449, doi:10.5194/gmd-4-435-2011, 2011.

Fawcett, T.: An introduction to ROC analysis, Pattern Recogn. Lett., 27, 861–874, 2006.

Forrester, A. I. J., Sobester, A., and Keane, A. J.: Engineering Design via Surrogate Modelling – A Practical Guide, Wiley, 2008.

Fox-Kemper, B., Ferrari, R., and Hallberg, R.: Parameterization of mixed layer eddies, Part I: Theory and diagnosis, J. Phys. Oceanogr., 38, 1145–1165, 2008.

Gent, P. R. and McWilliams, J. C.: Isopycnal mixing in ocean circulation models, J. Phys. Oceanogr., 20, 150–155, 1990.

Gent, P. R., Danabasoglu, G., Donner, L. J., Holland, M. M., Hunke, E. C., Jayne, S. R., Lawrence, D. M., Neale, R. B., Rasch, P. J., Vertenstein, M., Worley, P. H., Yang, Z.-L., and Zhang, M.: The Community Climate System Model Version 4, J. Climate, 24, 4973–4991, 2011.

Griffies, S.: Fundamentals of Ocean Climate Models, Princeton University Press, 2004.

Helton, J. C. and Davis, F. J.: Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems, Reliab. Eng. Syst. Safe., 81, 23–69, 2003.

Helton, J., Johnson, J., Sallaberry, C., and Storlie, C.: Survey of sampling-based methods for uncertainty and sensitivity analysis, Reliab. Eng. Syst. Safe., 91, 1175–1209, 2006.

Hosmer, D. W. and Lemeshow, S.: Applied Logistic Regression, 2nd Edn., Wiley-Interscience, 2000.

Jackson, C., Sen, M., Huerta, G., Deng, Y., and Bowman, K.: Error reduction and convergence in climate prediction, J. Climate, 21, 6698–6709, 2008.

Jayne, S. R.: The impact of abyssal mixing parameterizations in an ocean general circulation model, J. Phys. Oceanogr., 39, 1756–1775, 2009.

Jochum, M., Danabasoglu, G., Holland, M., Kwon, Y.-O., and Large, W. G.: Ocean viscosity and climate, J. Geophys. Res., 113, C06017, doi:10.1029/2007JC004515, 2008.

Kotsiantis, S. B.: Supervised Machine Learning: a Review of Classification Techniques, in: Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, IOS Press, Amsterdam, The Netherlands, The Netherlands, 3–24, 2007.

Large, W. and Yeager, S.: The global climatology of an interannually varying air–sea flux data set, Clim. Dynam., 33, 341–364, 2009.

Large, W. G., McWilliams, J. C., and Doney, S. C.: Oceanic vertical mixing: A review and a model with a nonlocal boundary layer parameterization, Rev. Geophys., 32, 363–403, 1994.

Large, W. G., Danabasoglu, G., McWilliams, J. C., Gent, P. R., and Bryan, F. O.: Equatorial circulation of a global ocean climate model with anisotropic horizontal viscosity, J. Phys. Oceanogr., 31, 518–536, 2001.

Lawrence Livermore National Laboratory Green Data Oasis, available through anonymous ftp at: ftp://gdo148.ucllnl.org/pub/ (last access: 24 June 2013), 2013.

Lucas, D. D. and Prinn, R. G.: Parametric sensitivity and uncertainty analysis of dimethylsulfide oxidation in the clear-sky remote marine boundary layer, Atmos. Chem. Phys., 5, 1505–1525, doi:10.5194/acp-5-1505-2005, 2005.

Marzban, C.: The ROC curve and the area under it as performance measures, Weather Forecast., 124, 1106–1113, 2004.

McKay, M. D., Beckman, R. J., and Conover, W. J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics, 21, 239–245, 1979.

MLdata.org: Machine learning data set repository, available at: http://mldata.org/repository/data/viewslug/climate-model-simulation-crashes/ (last access: 24 June 2013), archived on 24 June 2013: http://www.webcitation.org/6HcrKQ4Lk, 2013.

Murphy, J., Sexton, D., Barnett, D., Jones, G., Webb, M., Collins, M., and Stainforth, D.: Quantification of modelling uncertainties in a large ensemble of climate change simulations, Nature, 430, 768–772, 2004.

National Research Council Report: Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification, The National Academies Press, 2012.

Pipitone, J. and Easterbrook, S.: Assessing climate model software quality: a defect density analysis of three models, Geosci. Model Dev., 5, 1009–1022, doi:10.5194/gmd-5-1009-2012, 2012.

Platt, J. C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in Large Margin Classifiers, MIT Press, 61–74, 1999.

Prati, R. C., Batista, G. E. A. P. A., and Monard, M. C.: Class imbalances versus class overlapping: an analysis of a learning system behavior, in: MICAI, edited by: Monroy, R., Arroyo-Figueroa, G., Sucar, L. E., and Azuela, J. H. S., Lecture Notes in Computer Science, vol. 2972, Springer, 312–321, 2004.

Randall, D., Wood, R., Bony, S., Colman, R., Fichefet, T., Fyfe, J., Kattsov, V., Pitman, A., Shukla, J., Srinivasan, J., Stouffer, R., Sumi, A., and Taylor, K.: Climate Models and Their Evaluation, in: Climate Change 2007: The Physical Science Basis, Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change, edited by: Solomon, S., Qin, D., Manning, M., Chen, Z., Marquis, M., Averyt, K. B., Tignor, M., and Miller, H. L., Cambridge University Press, 2007.

Rugaber, S., Dunlap, R., Mark, L., and Ansari, S.: Managing software complexity and variability in coupled climate models, IEEE Software, 28, 43–48, 2011.

Saltelli, A., Chan, K., and Scott, E. M.: Sensitivity Analysis, Wiley, 2000.

Sanderson, B.: A multimodel study of parametric uncertainty in predictions of climate response to rising greenhouse gas concentrations, J. Climate, 24, 1362–1377, 2011.

Shiogama, H., Watanabe, M., Yoshimori, M., Yokohata, T., Ogura, T., Annan, J. D., Hargreaves, J. C., Abe, M., Kamae, Y., O'ishi, R., Nobui, R., Emori, S., Nozawa, T., Abe-Ouchi, A., and Kimoto, M.: Perturbed physics ensemble using the MIROC5 coupled atmosphere-ocean GCM without flux corrections: experimental design and results, Clim. Dynam., 39, 3041–3056, 2012.

Smith, R., Jones, P., Briegleb, B., Bryan, F., Danabasoglu, G., Dennis, J., Dukowicz, J., Eden, C., Fox-Kemper, B., Gent, P., Hecht, M., Jayne, S., Jochum, M., Large, W., Lindsay, K., Maltrud, M., Norton, N., Peacock, S., Vertenstein, M., and Yeager, S.: The Parallel Ocean Program (POP) reference manual, ocean component of the Community Climate System Model (CCSM), Tech. Rep. LAUR-10-01853, Los Alamos National Laboratory, 141 pp., 2010.

Smith, R. D. and McWilliams, J. C.: Anisotropic horizontal viscosity for ocean models, Ocean Model., 5, 129–156, 2003.

Sobol, I. M.: Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, Math. Comput. Simulat., 55, 271–280, 2001.

Stainforth, D., Aina, T., Christensen, C., Collins, M., Faull, N., Frame, D. J., Kettleborough, J. A., Knight, S., Martin, A., Murphy, J. M., Piani, C., Sexton, D., Smith, L. A., Spicer, R. A., Thorpe, A. J., and Allen, M. R.: Uncertainty in predictions of the climate response to rising levels of greenhouse gases, Nature, 433, 403–406, 2005.

Stein, M.: Large sample properties of simulations using Latin hypercube sampling, Technometrics, 29, 143–151, 1987.

Stensrud, D. J.: Parameterization Schemes: Keys to Understanding Numerical Weather Prediction Models, Cambridge University Press, 2009.

Sudret, B.: Global sensitivity analysis using polynomial chaos expansions, Reliab. Eng. Syst. Safe., 93, 964–979, 2008.

Swets, J. A.: Measuring the accuracy of diagnostic systems, Science, 240, 1285–1293, 1988.

Tannahill, J., Lucas, D. D., Domyancic, D., Brandon, S., and Klein, R.: Data intensive uncertainty quantification: Applications to climate modeling, in: Poster Presented at Super Computing 11, November 12–18, Seattle, Washington, USA, doi:10.1145/2148600.2148610, 2011.

Tatang, M., Pan, W., Prinn, R., and McRae, G.: An efficient method for parametric uncertainty analysis of numerical geophysical models, J. Geophys. Res., 102, 21925–21932, 1997.

The HadGEM2 Development Team: G. M. Martin, Bellouin, N., Collins, W. J., Culverwell, I. D., Halloran, P. R., Hardiman, S.

C., Hinton, T. J., Jones, C. D., McDonald, R. E., McLaren, A. J., O'Connor, F. M., Roberts, M. J., Rodriguez, J. M., Woodward, S., Best, M. J., Brooks, M. E., Brown, A. R., Butchart, N., Dearden, C., Derbyshire, S. H., Dharssi, I., Doutriaux-Boucher, M., Edwards, J. M., Falloon, P. D., Gedney, N., Gray, L. J., Hewitt, H. T., Hobson, M., Huddleston, M. R., Hughes, J., Ineson, S., Ingram, W. J., James, P. M., Johns, T. C., Johnson, C. E., Jones, A., Jones, C. P., Joshi, M. M., Keen, A. B., Liddicoat, S., Lock, A. P., Maidens, A. V., Manners, J. C., Milton, S. F., Rae, J. G. L., Ridley, J. K., Sellar, A., Senior, C. A., Totterdell, I. J., Verhoef, A., Vidale, P. L., and Wiltshire, A.: The HadGEM2 family of Met Office Unified Model climate configurations, Geosci. Model Dev., 4, 723–757, doi:10.5194/gmd-4-723-2011, 2011.

Vapnik, V. N.: The Nature of Statistical Learning Theory, Springer, New York, NY, USA, 1995.

Walter, K.: Narrowing Uncertainties, Science & Technology Review, available at: https://str.llnl.gov/JulAug10/klein.html (last access: 22 January 2013), archived on 22 January 2013: http://www.webcitation.org/6DsDwCRxY, 2010.

Washington, W. M. and Parkinson, C. L.: An introduction to three-dimensional climate modeling, University Science Books, 2nd Edn., 2005.

Webster, M., Scott, J., Sokolov, A., and Stone, P.: Estimating probability distributions from complex models with bifurcations: The case of ocean circulation collapse, J. Environ. Sys., 31, 1–21, doi:10.2190/A518-W844-4193-4202, 2004.

Wiener, N.: The homogeneous chaos, Am. J. Math., 60, 897–936, 1938.

Xiu, D. and Karnidakis, G. E. M.: The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput., 24, 619–644, 2002.