SIAM/ASA J. UNCERTAINTY QUANTIFICATION
Vol. 2, pp. 1–28

Evolving Bayesian Emulators for Structured Chaotic Time Series, with Application to Large Climate Models*

Daniel Williamson† and Adam T. Blaker‡

Abstract. We develop Bayesian dynamic linear model Gaussian processes for emulation of time series output for computer models that may exhibit chaotic behavior, but where this behavior retains some underlying structure. The statistical technology is particularly suited to emulating the time series output of large climate models that exhibit this feature and where we want samples from the posterior of the emulator to evolve in the same way as dynamic processes in the computer model do. The methodology combines key features of good uncertainty quantification (UQ) methods such as using complex mean functions to capture large-scale signals within parameter space, with dynamic linear models in a way that allows UQ to borrow strength from the Bayesian time series literature. We present an MCMC algorithm for sampling from the posterior of the emulator parameters when the roughness lengths of the Gaussian process are unknown. We discuss an interpretation of the results of this algorithm that allows us to use MCMC to fix the correlation lengths, making future online samples from the emulator tractable when used in practical applications where online MCMC is infeasible. We apply this methodology to emulate the Atlantic Meridional Overturning Circulation (AMOC) as a time series output of the fully coupled non–flux-adjusted atmosphere-ocean general circulation model HadCM3.

Key words. dynamic emulation, uncertainty quantification, climate models, Bayesian analysis

AMS subject classifications. 60Gxx, 37N10, 60G15

DOI. 10.1137/120900915

1. Introduction.

Computer models are used in many diverse areas of science to study the behavior of complex physical systems such as the Earth's climate. A computer model is, essentially, a mathematical function of a possibly large number of parameters. The function is often approximated as part of the model code, for example, using discretized solvers to evaluate solutions to partial differential equations or integrals within the mathematical description. Using such a model to make inferences about the physical system in question introduces a number of sources of uncertainty that must be quantified. These sources include structural uncertainty, often referred to as *model discrepancy*, which represents the extent to which the model fails to represent the true physics of the system through either inaccurate modelling or numerical approximation; decision uncertainty, which represents the uncertainty in the model output due to any decision or control parameters and the uncertainty as to what these might eventually be in the real world; and parametric uncertainty due to not knowing which choices of the model parameters lead to model output that best represents the physics of the system.

*Received by the editors December 3, 2012; accepted for publication (in revised form) September 24, 2013; published electronically January 16, 2014. This work was supported by NERC RAPID-RAPIT project NE/G015368/1.
http://www.siam.org/journals/juq/2/90091.html

†College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, UK (d.williamson@exeter.ac.uk).

‡National Oceanography Centre, Southampton, SO14 3ZH, UK (atb299@noc.ac.uk).

A key component of parametric uncertainty derives from our inability to run the computer model at every possible choice of the input parameters. Often called code uncertainty (Kennedy and O'Hagan, 2001), this feature of the analysis of computer experiments has received particular attention from the research community, leading to a general methodology based on emulators. An *emulator* is a stochastic representation of a computer model that generates a prediction for the output of a computer model at any setting of the model parameters and reports a measure of uncertainty for that prediction. Emulators can be evaluated in a fraction of the time it would take to run a computer model, and there are many different techniques for building them. Generally, an emulator is built by using an ensemble of runs at different parameter choices, called the *design*, and using that ensemble to fit a statistical model.

The form of the emulator depends largely on the form of the computer model output. There is a wide literature on how to fit emulators for models with univariate output (Sacks et al., 1989; Haylock and O'Hagan, 1996; Santner, Williams, and Notz, 2003). The models we will study return time series output, and the task will be to emulate these as a function of the model parameters and forcing. The computer models of interest exhibit what we term structured chaotic behavior. Such models are common in the study of complex dynamic systems with computer models, such as those found in climate and finance. The output is chaotic because slight changes to the input parameters or the initial conditions at which the model is run can effect large changes to the output. This chaos, however, retains some structure. The models are deterministic, not stochastic, which means that running the model twice at the same settings of the input parameters and initial conditions will return the same output. Slight perturbations to the parameters may lead to large differences in the exact evolution of the time series. However, global features such as parameter-dependent trends and the character of the variability will be very similar.

When emulating such time series output of a computer model, unless the initial conditions in the model are fixed for purely physical reasons, both at the time the parameters are changed and when any forcing is introduced, our emulators should not interpolate the model runs observed. This is because by choosing a specific time to change the experiment we are introducing sampling uncertainty. For example, when the parameters of a climate model are perturbed, the model must be run for a number of years until it is in equilibrium before any CO_2 forcing experiments can be applied. Once the model is in equilibrium, the point at which forcing is applied is both important in terms of the precise time series returned from the model and irrelevant to the scientific analysis being performed. The sampling uncertainty introduced by perturbing the parameters or altering the forcing in a climate model is known as *internal variability* (see, for example, Hawkins and Sutton, 2009) and must be explicitly modelled as part of a methodology for emulating time series output of computer models with structured chaotic behavior.

The emulation of time series is part of the more general problem of jointly emulating multivariate output of computer models. This has been done using multivariate Gaussian processes (Rougier, 2008; Conti and O'Hagan, 2010), though this is difficult if the output is reasonably high dimensional. A popular solution is to project the output to a lower-dimensional set of basis vectors and to then emulate the coefficients. Bases used in the literature include principal components (Challenor, McNeall, and Gattiker, 2009; Higdon et al., 2008b), wavelets

(Bayarri et al., 2007), and P-splines (Williamson, Goldstein, and Blaker, 2012). In each of these methods a low-dimensional basis is constructed to represent any output of the simulator, and the model output is projected onto that basis. A multivariate emulation of the coefficients using Gaussian processes can then take place. This approach has many advantages, including simplifying the numerical challenge faced in constructing and inverting high-dimensional covariance matrices for Gaussian processes, and allowing a great deal of interpretability to the emulated quantities. For example, basis projection often acts as a smoother, allowing global trends and large-scale signals to be captured. These emulators can account for structured chaos by allowing the smoother to represent the structure and removing sampling uncertainty at the dimension reduction stage.

There are problems with using basis representations too, particularly when modelling time series. It may be hard, if not impossible, to find a suitable basis that allows accurate capturing of all of the structure we require in every run in our design. Although history matching may help with this (see section 3), we may still have difficulty capturing "rare" or even "run specific" events that may be smoothed out, yet still represent behavior in a region of parameter space that is of interest. This would lead to the projected coefficients being a poor representation of the model output for those runs where we fail to do a good job, leading to an emulator not of the computer model, but of a poor fit to it. For most basis expansions, although we might sufficiently capture the smooth behaviors of interest, the structure and character of the variability in the output are not explicitly modelled. We may, by chance, only remove variability that explicitly accounts for the noise in our system through the dimension reduction, but this is unlikely. Draws from the posterior distribution of the emulator are likely to resemble a smooth representation of the output, but even if "white noise" is subsequently added to these draws, they may not look like realistic realizations of the modelled dynamic process as output by the simulator.

Emulators that aim to capture the dynamics of the system have been attempted by Conti et al. (2009) and by Liu and West (2008). Conti et al. (2009) construct what they call dynamic emulators by modelling the one-step transition function of the state vector as a function of the state vector and the model parameters. By accurately modelling how the model moves from one time step to the next, they can evolve draws from the posterior distribution of the output of the simulator at any choice of the input parameters by updating the full state vector of the model directly. For many problems however, the state vector is prohibitively large such that this approach is infeasible.

Liu and West (2008) introduced a dynamic approach to emulation based on dynamic linear models (West and Harrison, 1997; Prado and West, 2010) and Gaussian processes. They modelled the time series output of a computer model as a time varying autoregressive (TVAR) process plus an independent Gaussian process at each time point. Their emulators interpolate the design points. Their approach uses the TVAR process to model temporal behavior common throughout parameter space and uses the Gaussian process to describe independent deviations from this global behavior, for any parameter choice, at each time point. However, the model output may not exhibit the same degree of smoothness throughout its parameter space, and there may be complex parameter-dependent temporal behavior that is more easily captured by fitting a complex, parameter-dependent, global mean function and allowing the Gaussian process residual to do less work by capturing "local" variability around this mean function.

The approach to building emulators by fitting parameter-dependent global mean functions and modelling residual variability with Gaussian processes has been widely applied (see, for example, Craig et al., 2001; Cumming and Goldstein, 2009; Rougier et al., 2009) and was discussed and advocated by Kaufman et al. (2011). We will apply this approach to the emulation of structured chaotic time series output of computer models.

In this paper we extend the dynamic linear model approach of Liu and West (2008) to allow for model output exhibiting structured chaos by adding a white noise process to the residual. We superpose a dynamic regression in the model parameters and forcing to capture features in the evolution of the time series output that change as we move through parameter space in a way captured by simple functions of the parameters. We then allow the Gaussian process to capture the remaining "local" signal. We illustrate our methodology by emulating a time series of an ocean transport called the Atlantic Meridional Overturning Circulation (AMOC), output from the third Hadley Centre climate model HadCM3 (Gordon et al., 2000; Pope et al., 2000), and an atmosphere ocean generalized circulation model (AOGCM) used in the fourth report from the Intergovernmental Panel on Climate Change (IPCC) (Solomon et al., 2007).

The quantity is of particular interest to ocean and climate scientists because of the associated heat transported by the predominantly northward flowing water masses in the upper limb of the AMOC. Observations of the AMOC have been made from April 2004 until the present by a collaboration of UK and US scientists (Rayner et al., 2011), and a recent study found the AMOC to be responsible for nearly 90% of the net meridional ocean heat flux at 26°N (Johns et al., 2011). Studies using both numerical models and observation have found the AMOC to exhibit substantial variability on a wide range of time scales (see, for example, Blaker et al., 2012; McCarthy et al., 2012; Balan Sarojini et al., 2011; Biastoch et al., 2008), and it is expected that the AMOC will reduce in strength by 25% over the next few decades (Bindoff et al., 2007) due to increasing concentrations of atmospheric greenhouse gases.

In section 2 we present the basic details of computer model emulation using Gaussian processes and introduce our methodology for emulating structured chaotic time series output of computer models by generalizing the ideas of Liu and West (2008). In section 3 we describe the application, introduce an ensemble of climate model runs, and discuss the use of history matching to remove unphysical runs prior to emulation of complex time series. In section 4 we discuss the different prior judgements required to build our emulators and describe the choices made in our application. Section 5 introduces an MCMC algorithm for sampling from the posterior when the roughness lengths of the Gaussian process are uncertain and discusses the interpretation of our statistical model. In section 6 we present results and section 7 contains a discussion. Proofs of certain results that drive the Bayesian updating of our emulator that are from the time series literature are included in Appendix A for interested UQ practitioners who may not be familiar with them.

2. Emulators.

2.1. Emulation.

Let $\boldsymbol{x}$ be a set of input and decision parameters for a computer model $\boldsymbol{f}(\boldsymbol{x})$. A typical form for an emulator for component f_i of $\boldsymbol{f}$ is

$$f_i(\boldsymbol{x}) = \sum_j \beta_{ij} g_j(\boldsymbol{x}) \oplus \epsilon_i(\boldsymbol{x}), \tag{1}$$

where the operator $\oplus$ indicates the addition of independent quantities, $\boldsymbol{g}(\boldsymbol{x})$ is a specified vector of regressors, β is a matrix of uncertain coefficients, and $\epsilon_i(\boldsymbol{x})$ is a Gaussian process with stationary correlation function $R(|\boldsymbol{x} - \boldsymbol{x}'|; \boldsymbol{\rho})$ with parameters $\boldsymbol{\rho}$. The first half of this equation is designed to capture large-scale (global) features of the computer model output throughout parameter space, and the residual process models the local variability (Williamson, Goldstein, and Blaker, 2012). Time can be handled by including it as part of the vector $\boldsymbol{x}$ (emulating as a function of time) (see, for example, Rougier, 2008) or by allowing i to indicate time and by specifying beliefs about β with a temporal correlation function for the $\boldsymbol{\epsilon}(\boldsymbol{x})$ (see, for example, Craig et al., 2001).

Given the form in (1), an emulator is constructed by choosing the elements of $\boldsymbol{g}(\boldsymbol{x})$ and the form of the covariance function for $\epsilon_i(\boldsymbol{x})$ and then initializing the random field $\{\beta, \epsilon_i(\boldsymbol{x})\}$ by performing a Bayesian update of our model using prior judgments and an ensemble of runs of the computer model. For example, suppose the covariance function of $\epsilon_i(\boldsymbol{x})$ is $\sigma^2 R(\cdot, \cdot; \boldsymbol{\rho})$. Suppose, in addition, that full prior probability distributions are available on $\{\beta, \sigma, \boldsymbol{\rho}\}$. Then, the random field $\pi(f_i(\boldsymbol{x}))$ with

$$f_i(\boldsymbol{x})|\beta, \sigma, \boldsymbol{\rho} \sim \mathrm{GP}\left(\beta \boldsymbol{g}(\boldsymbol{x}), \sigma^2 R(\cdot, \cdot; \boldsymbol{\rho})\right)$$

can be updated by an ensemble F using Bayes' theorem and the decomposition

$$\pi(f_i(\boldsymbol{x}), \beta, \sigma, \boldsymbol{\rho}|F) = \pi(\beta, \sigma, \boldsymbol{\rho}|F)\pi(f_i(\boldsymbol{x})|\beta, \sigma, \boldsymbol{\rho}, F),$$

with

$$f_i(\boldsymbol{x})|\beta, \sigma, \boldsymbol{\rho}, F \sim \mathrm{GP}\left(m(\boldsymbol{x}), c(\cdot, \cdot)\right),$$

where $m(\boldsymbol{x})$ and $c(\cdot, \cdot)$ depend on $\boldsymbol{g}(\boldsymbol{x})$, $R(\cdot, \cdot)$, and the prior distribution on $\{\beta, \sigma, \boldsymbol{\rho}\}$. Under certain strong assumptions that fix the elements of $\boldsymbol{\rho}$ and that allow β and σ to be integrated out, a closed form is available for $\pi(f_i(\boldsymbol{x})|F)$ through conjugate analysis (see, for example, Haylock and O'Hagan, 1996). When we are not willing to make these assumptions, samples from $\pi(f_i(\boldsymbol{x})|F)$ can be obtained using MCMC.

An alternative to a full probabilistic analysis is a Bayes linear analysis. This approach replaces the specification of a full prior probability distribution for uncertain parameters with specification of means, variances, and covariances and then performs a second order analysis of the resulting model using the ensemble F. Full details of this method can be found in Craig et al. (1996), Cumming and Goldstein (2010), and Williamson and Goldstein (2012).

Traditionally, emulators were built with constant prior mean functions with $g(\boldsymbol{x}) = 1$ and the Gaussian process residual capturing global structure of the function in its parameter space as well as local information close to observed model runs (Sacks et al., 1989; Currin et al., 1991; Santner, Williams, and Notz, 2003). However, this is inefficient in many cases where complex mean functions can be constructed that capture the global behavior and allow the Gaussian process to more accurately capture the residual local variability.

In the computer experiment literature it has now become popular to choose a complex vector $\boldsymbol{g}(\boldsymbol{x})$ so as to model large-scale features of the computer code (see, for example, Craig et al., 1997; Rougier, 2008; Rougier et al., 2009; Williamson, 2010; Vernon, Goldstein, and Bower, 2010). This has the effect of reducing the importance of the Gaussian process to the quality

of the overall fit. In some cases the fit is judged to be adequate enough to have soaked up all of the parameter-dependent behavior of the model, leaving an uncorrelated noise process residual (Rougier et al., 2009; Sexton, Murphy, and Collins, 2011; Williamson et al., 2013). Kaufman et al. (2011) use complex mean functions to restrict the correlation lengths in order to use correlation functions with restricted support and to reduce the computational burden when emulating functions using large ensembles.

2.1.1. The nugget process. The emulators described above will interpolate the observed model runs following Bayesian updating. This is desirable when the computer code is deterministic in many cases. However, there are incidences when this is not appropriate. For example, in many situations, although all parameters in the computer model were varied in the design, only a handful are deemed "active." In these circumstances the Gaussian process will be built using the active variables only. Variability due to the inactive variables, though judged not to contribute a great deal to the overall variability of the output, must be modelled separately.

We may also judge that there is a discrepancy between the Gaussian process and the computer model output, perhaps arising from a misspecified correlation structure or lack of stationarity (Grammacy and Lee, 2012). In these situations we may add a "nugget term," a mean zero "noise" process, to the emulator that indicates variability in the output that is not attributed to the inputs we include in our emulator. The term "nugget" originates from the spatial statistics literature, where it is included in the fitting of Gaussian processes to account for measurement error (Cressie, 1993; Diggle and Ribeiro, 2007). A review and exploration of the effect of the use of a nugget in emulators for computer models can be found in Andrianakis and Challenor (2012). Kleijnen (2009) also discusses emulation for computer models where the output is not deterministic.

2.2. Dynamic linear model Gaussian processes. In the remainder of this article we refer to the computer model output corresponding to parameters $\boldsymbol{x}$ at time t as $f_t(\boldsymbol{x})$, explicitly acknowledging time in our notation. The dynamic approach to emulation introduced by Liu and West (2008) models the emulator as a TVAR process plus an independent Gaussian process via

$$f_t(\boldsymbol{x}) = \sum_{j=1}^{p} \theta_{t,j} f_{t-j}(\boldsymbol{x}) \oplus \epsilon_t(\boldsymbol{x})$$

with $\epsilon_t(\boldsymbol{x}) \sim \mathrm{N}(0, v_t)$ independent over time and with a square-exponential correlation function with uncertain roughness lengths. This specification captures temporal relationships using the TVAR process, yet the modelling makes no specific provision for parameter-dependent temporal relationships, leaving these to be captured by the Gaussian process at each time point. We might expect this kind of model to perform well for simulators whose time series output contains regular features throughout parameter space, for example, peaks at specific times which would appear in the posterior distribution of θ. These features may be more or less exaggerated depending on x, and the Gaussian process $\epsilon_t(\boldsymbol{x})$ will allow this to be captured. However, in models where the temporal behavior depends heavily on the parameters and where the time series can look quite different for different parameter choices, the Gaussian process will have to capture most of the behavior of the model output independently in time. In

this way, the dynamic emulators introduced by Liu and West (2008) are similar to standard emulators with constant mean functions.

We generalize their approach by superposing a dynamic regression in the input parameters x. This superposition preserves the status of the model as a dynamic linear model (West and Harrison, 1997, Chapter 6). We also superpose a nugget process to ensure that our emulators do not interpolate the model runs, as this is not appropriate for models that exhibit structured chaos. The nugget in our model captures internal variability and may also be used to account for any variability due to inactive inputs.

Our model is

$$f_t(\boldsymbol{x}) = \sum_{j=1}^{p} \theta_{t,j} f_{t-j}(\boldsymbol{x}) + \sum_{k=1}^{q} \beta_{t,k} g_k(\boldsymbol{x_t}) \oplus \epsilon_t(\boldsymbol{x}) \oplus \Delta_t, \tag{2}$$

with uncertain parameters $\boldsymbol{\theta}_t = (\theta_{t,1}, \ldots, \theta_{t,p})^T$ and $\boldsymbol{\beta}_t = (\beta_{t,1}, \ldots, \beta_{t,q})^T$, $\epsilon_t(\boldsymbol{x})$ a Gaussian process with $\epsilon_t(\boldsymbol{x}) \sim \mathrm{N}(0, \tau v_t)$ independent over time, and $\mathrm{Cov}\left[\epsilon_t(\boldsymbol{x}), \epsilon_t(\boldsymbol{x'})\right] = \tau v_t R(|x - x'|; \boldsymbol{\rho})$. Δ_t is a nugget residual with $\Delta_t \sim \mathrm{N}(0, \Delta v_t)$, also independent over time. τ and Δ are preassigned scalars that we take here to sum to 1 so that the v_t can be viewed as the variance of the mean zero residual component of the model and τ and Δ are the proportions of the residual variability that are correlated "signal" and "white noise" respectively, though this constraint is not a requirement of the modelling.

The vector $\boldsymbol{g}(\boldsymbol{x_t})$ is a vector of regressors in $\boldsymbol{x_t}$, which is potentially a time series that can be derived explicitly from $\boldsymbol{x}$. The simplest case is $\boldsymbol{x_t} = \boldsymbol{x}$ for all t; however, some computer models require time series inputs that are controlled through a handful of parameters in $\boldsymbol{x}$. It may be useful to allow some terms in our model of the simulator to be described as functions of the elements of these time series rather than the parameters, as we shall see in our application.

Let $\boldsymbol{\psi}_t = (\boldsymbol{\theta}_t, \boldsymbol{\beta}_t)^T$ follow a random walk in time so that

$$\boldsymbol{\psi}_t = \boldsymbol{\psi}_{t-1} + w_t, \qquad w_t \sim \mathrm{N}(0, v_t W_t), \tag{3}$$

for some matrix W_t. We specify a normal inverse gamma prior for $\boldsymbol{\psi}_0$, v_0 with $\boldsymbol{\psi}_0 | v_0, D_0 \sim \mathrm{N}(\boldsymbol{m_0}, v_0 C_0^*)$, $\phi_0 | D_0 \sim \mathrm{G}(n_0/2, d_0/2)$, and $\phi_t = v_t^{-1}$ for all t. The collection D_t represents all hyperparameter information up to time t for all t as well as ensemble data up to time t.

Models for sequences v_t and W_t are based on variance discounting (West and Harrison, 1997; Prado and West, 2010) with

$$W_t | D_{t-1} = \frac{(1 - \delta_w)}{\delta_w} C_{t-1}^* \tag{4}$$

and

$$\phi_t | D_{t-1} = \frac{\gamma_t \phi_{t-1}}{\delta_v}, \tag{5}$$

where $\gamma_t \sim \mathrm{Beta}(\delta_v n_{t-1}/2, (1-\delta_v) n_{t-1}/2)$ so that ϕ_t evolves as a result of independent random shocks γ_t/δ_v. Parameters C_{t-1}^*, n_{t-1}, and d_{t-1} will be derived below as part of a conjugate Bayesian update of the normal inverse gamma model. The parameters δ_v and δ_w are discount

parameters to be specified. With the model above it can be shown (for example, using Mellin transforms) that the prior for ϕ_t can be written as

$$\phi_t | D_{t-1} \sim \mathrm{G}(\delta_v n_{t-1}/2, \delta_v d_{t-1}/2).$$

Let $X_1, \ldots, X_n$ represent the ensemble design points, and define

$$H_t^T = \begin{pmatrix} f_{t-1}(X_1) & \cdots & f_{t-p}(X_1) & g_1(X_{1t}) & \cdots & g_q(X_{1t}) \\ \vdots & & \vdots & \vdots & & \vdots \\ f_{t-1}(X_n) & \cdots & f_{t-p}(X_n) & g_1(X_{1t}) & \cdots & g_q(X_{nt}) \end{pmatrix}$$

and

$$\boldsymbol{\epsilon_t} = (\epsilon_t(X_1), \ldots, \epsilon_t(X_n))^T, \qquad \boldsymbol{F_t} = (f_t(X_1), \ldots, f_t(X_n))^T.$$

Then

$$\boldsymbol{F_t} = H_t^T \boldsymbol{\psi}_t + \boldsymbol{\epsilon_t} + \boldsymbol{\Delta_t}, \qquad \boldsymbol{\epsilon_t} \sim \mathrm{N}(0, v_t \tau \Sigma),$$

with

$$\Sigma_{ij} = R(|X_i - X_j|; \boldsymbol{\rho}), \qquad \boldsymbol{\Delta_t} \sim \mathrm{N}(0, v_t \Delta \mathbb{I}_n).$$

In what follows we present details of the Bayesian update of this model conditioned on $\boldsymbol{\rho}$ with the form of $R(\cdot)$ assumed known and describe how to sample evolutions of the emulator. Though some of the results and recurrence relationships are considered "standard" in the time series literature, we present details here and proofs where appropriate in Appendix A for readers who are familiar with the computer experiment literature but not with dynamic linear model theory.

2.3. Bayesian updating. The updating of $\{\boldsymbol{\psi}_t, v_t\}$ with the ensemble data $\boldsymbol{F_{1:T}}$ happens iteratively from $t = 1, \ldots, T$ by a procedure called forward filtering in the time series literature (West and Harrison, 1997). For each time t the prior distribution of $\phi_t | D_{t-1}$ is given above and the marginal distribution of $\boldsymbol{\psi}_{t-1}$ is $\boldsymbol{\psi}_{t-1} | D_{t-1} \sim T_{\delta_v n_{t-1}}(\boldsymbol{m_{t-1}}, C_{t-1})$ with $C_{t-1} = S_{t-1} C^*_{t-1}$ and $S_{t-1} = d_{t-1}/n_{t-1}$.

From this information, we can derive recurrence relationships that perform the Bayesian updates for all time using

$$\begin{aligned} \boldsymbol{\psi}_t | D_{t-1} &\sim T_{\delta_v n_{t-1}}(\boldsymbol{a_t}, R_t), & \boldsymbol{F_t} | D_{t-1} &\sim T_{\delta_v n_{t-1}}(\boldsymbol{h_t}, Q_t), \\ \phi_t | D_t &\sim \mathrm{G}(n_t/2, d_t/2), & \boldsymbol{\psi}_t | D_t &\sim T_{n_t}(\boldsymbol{m_t}, C_t), \end{aligned}$$

where

$$\begin{aligned} R_t &= \frac{C_{t-1}}{\delta_w}, & \boldsymbol{a_t} &= \boldsymbol{m_{t-1}}, & Q_t &= H_t^T R_t H_t + S_{t-1}(\tau\Sigma + \Delta\mathbb{I}_n), \\ \boldsymbol{h_t} &= H_t^T \boldsymbol{a_t}, & n_t &= \delta_v n_{t-1} + n, & d_t &= \delta_v d_{t-1} + S_{t-1} \boldsymbol{e_t}^T Q_t^{-1} \boldsymbol{e_t}, \\ \boldsymbol{e_t} &= \boldsymbol{F_t} - \boldsymbol{h_t}, & \boldsymbol{m_t} &= \boldsymbol{a_t} + A_t \boldsymbol{e_t}, & C_t &= \frac{S_t}{S_{t-1}}(R_t - A_t Q_t A_t^T) \end{aligned}$$

and $A_t = R_t H_t Q_t^{-1}$. Although these relationships are known in the time series literature and only slightly changed to allow for our augmentations to the model of Liu and West (2008), we include a derivation in Appendix A for interested readers.

2.4. Sample evolutions. Having obtained posterior distributions for the uncertain parameters in our model we can draw samples from the posterior and sample an evolution of our emulator in time based on the sampled values of the known parameters. To draw a sample path from our emulator at a new input x we first sample values of $v_{1:T}$ and $\boldsymbol{\psi}_{1:T}$ and then, for $t = 1 : T$, sample iteratively from $f_t(\boldsymbol{x})|v_{1:T}, \boldsymbol{\psi}_{1:T}, \boldsymbol{F_t}, f_{1:(t-1)}(\boldsymbol{x}) \sim \mathrm{N}(\mu_t(\boldsymbol{x}), \sigma_t^2(\boldsymbol{x}))$ with

$$\mu_t(\boldsymbol{x}) = \sum_{j=1}^{p} \theta_{t,j} f_{t-j}(\boldsymbol{x}) + \sum_{k=1}^{q} \beta_{t,k} g_k(\boldsymbol{x_t}) + \tau \boldsymbol{\rho}_{\boldsymbol{x}}^T (\tau\Sigma + \Delta\mathbb{I}_n)^{-1} \boldsymbol{\xi}_t \tag{6}$$

and

$$\sigma^2(\boldsymbol{x}) = v_t(\tau + \Delta - \tau^2 \boldsymbol{\rho}_{\boldsymbol{x}}^T (\tau\Sigma + \Delta\mathbb{I}_n)^{-1} \boldsymbol{\rho}_{\boldsymbol{x}}), \tag{7}$$

where $\boldsymbol{\xi}_t = (\xi_t(X_1), \ldots, \xi_t(X_n))$, $\xi_t(X_i) = f_t(X_i) - \sum_{j=1}^{p} \theta_{t,j} f_{t-j}(X_i) - \sum_{k=1}^{q} \beta_{t,k} g_k(X_{it})$, and $\boldsymbol{\rho}_{\boldsymbol{x}} = (R(|x - X_1|), \ldots, R(|x - X_n|))^T$. Note here that there is a starting value problem for $p > 0$, where the samples described above are undefined for $t < p + 1$. We discuss this further in section 4.

We first describe how to sample from $\{v_{1:T}, \boldsymbol{\psi}_{1:T}\}$ by backwards sampling. First, sample $v_T \sim \mathrm{G}(n_T/2, d_T/2)$; then, for $t = T-1, \ldots, 1$, sample $\eta_t \sim \mathrm{G}((1-\delta_v)n_t/2, d_t/2)$ and calculate

$$v_t^{-1} = \eta_t + \frac{\delta_v}{v_{t+1}}.$$

Although this result is known in the time series literature, we include a proof that this scheme samples from the correct distribution in Appendix A for interested readers. Given $v_{1:T}$, we sample $\boldsymbol{\psi}_{1:T}$ by first sampling a $\boldsymbol{\psi}_T$ from

$$\boldsymbol{\psi}_T|v_{1:T}, D_T \sim \mathrm{N}(\boldsymbol{m_T}, v_T C_T^*);$$

then, for $t = T - 1, \ldots, 1$, we iteratively sample from

$$\boldsymbol{\psi}_t|\boldsymbol{\psi}_{t+1}, v_t, D_t \sim \mathrm{N}((1 - \delta_w)\boldsymbol{m_t} + \delta_w \boldsymbol{\psi}_{t+1}, (1 - \delta_w) v_t C_t^*),$$

with $C_t^* = C_t/S_t$ for all t. Again, a short proof of this known result is given in Appendix A for interested readers.

3. Emulating the AMOC in HadCM3. Our motivating application involves emulating the AMOC as output by HadCM3 as a function of model parameters and CO_2 forcing. This is part of a National Environment Research Council funded project called RAPIT (Risk Analysis, Probability and Impacts Team) that aims to quantify the risk of rapid change or shutdown of the AMOC. To meet this challenge we will build an emulator for key outputs of HadCM3, such as the AMOC time series, and use calibration (Kennedy and O'Hagan, 2001) to provide probability distributions for the relevant quantities in the real world. In this paper we use the methodology described to construct an evolving emulator for the AMOC that will be used as part of future work.

3.1. The ensemble. The RAPIT ensemble we are using was designed on 27 of HadCM3's parameters, controlling processes such as tracer mixing in the oceans, cloud formation and distribution, and sea ice formation as well as many others. Each ensemble member is associated with a decision parameter controlling the rate of CO_2 concentration increase for the simulation. Each simulation runs through 120 model years, with 50 years devoted to allowing the model to reach equilibrium at preindustrial CO_2 following parameter perturbation and 70 years during which CO_2 concentration is increased by the rate indicated by the decision parameter. This parameter is chosen from the range $[0, 0.04]$, with 0.01 indicating annual 1% increase in CO_2 concentrations and a doubling of CO_2 concentrations by the end of the experiment.

A Latin hypercube design containing $10,000$ different perturbations was designed and the runs were submitted to climate prediction dot net (CPDN, http://climateprediction.net), a distributed computing project through which climate models are distributed to run as background processes on personal computers volunteered by members of the public. With over $35,000$ active hosts, many of which using multicore machines, we can run very large ensembles with this resource. However, the price for obtaining large ensembles on personal computers is that each member takes a very long time to complete. Runtimes differ from machine to machine; however, it takes approximately 28 days for a reasonably fast and dedicated PC to run 40 years of the simulation. Most runs take much longer, and many are incomplete after many months. Details of the design and more on the ensemble can be found in Williamson et al. (2013) and Yamazaki et al. (2013).

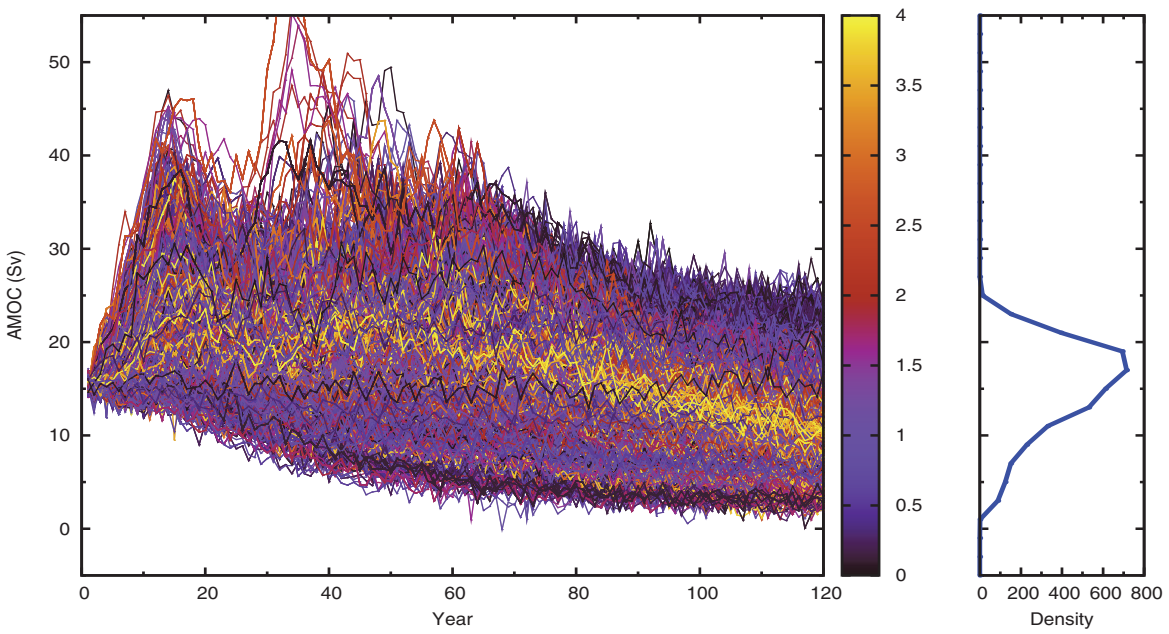

Figure 1. *The AMOC at 26°N in the RAPIT ensemble colored by the annual percentage increase of CO_2 forcing applied after year 50. The right-hand panel shows the density of the "final" AMOC value and was computed by binning the final 20 year means of the AMOC into bins with 2Sv intervals ($1Sv = 1 \times 10^6 \mathrm{m}^3\mathrm{s}^{-1}$).*

Figure 1 shows all of the transient (nonzero rate of CO_2 increase) runs in the RAPIT

ensemble that had completed 120 years of simulation at the time of this writing. The plot shows a very wide range of AMOC behaviors, many of which are considered as "unphysical" by climate scientists. The peak at around year 15, for example, is an artificial synchronous excitation of a mode of climate variability known to exist in HadCM3 which is a result of the introduction of parameter perturbations (Blaker and Williamson, 2014). These unusual behaviors may be difficult to emulate and, due to their artificial nature, emulating them is not likely to be of interest, although their presence in the data may mask behavior that is of interest and make that difficult to model. The solution we present is to use simpler and easier to model outputs from the computer model, such as univariate averages, in order to rule out parts of parameter space that lead to unphysical behavior by history matching.

3.2. History match. So that we may focus on emulating only those parts of parameter space that we cannot rule out as being unphysical, we first perform a history match on the ensemble. History matching is a statistical method that uses observational data in order to rule out regions of parameter space where it is judged to be implausible that the model could mimic the real world given all of the relevant uncertainties. It has been applied in a number of areas, including on computer models for oil reservoirs (Craig et al., 1996; Cumming and Goldstein, 2010) and on models that simulate the formation of galaxies at the beginning of the universe (Vernon, Goldstein, and Bower, 2010).

The method works by linking the computer model to reality via a statistical model that explicitly acknowledges a discrepancy between reality and the simulator. The most popular model uses the "best input approach" (Kennedy and O'Hagan, 2001), which is to say that reality $\boldsymbol{y}$ can be modelled as

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}^*) \oplus \boldsymbol{\eta}, \tag{8}$$

where $\boldsymbol{x}^*$ is the setting of the model parameters referred to as the best input and $\boldsymbol{\eta}$ is model discrepancy. Observed history $\boldsymbol{z}$, usually related to $\boldsymbol{y}$ via $\boldsymbol{z} = \boldsymbol{y} \oplus \boldsymbol{e}$ with mean zero measurement error $\boldsymbol{e}$, is then used to rule out regions of parameter space via the implausibility measure $\mathcal{I}(\boldsymbol{x}) = \max_i\{\mathcal{I}_i(\boldsymbol{x})\}$, where

$$\mathcal{I}_i(\boldsymbol{x}) = \frac{z_i - \mathrm{E}\left[f_i(\boldsymbol{x})\right]}{\sqrt{\mathrm{Var}\left[z_i - \mathrm{E}\left[f_i(\boldsymbol{x})\right]\right]}}. \tag{9}$$

The form of $\mathrm{Var}\left[z - \mathrm{E}\left[f(\boldsymbol{x})\right]\right]$ depends on the method used to link $f(\boldsymbol{x})$ to y; for example, by adopting (8) we get

$$\mathrm{Var}\left[\boldsymbol{z} - \mathrm{E}\left[\boldsymbol{f}(\boldsymbol{x})\right]\right] = \mathrm{Var}\left[\boldsymbol{e}\right] + \mathrm{Var}\left[\boldsymbol{\eta}\right] + \mathrm{Var}\left[\boldsymbol{f}(\boldsymbol{x})\right].$$

Implausibility can be viewed as a standardized distance with "large" values of $\mathcal{I}(\boldsymbol{x})$ for any particular value of $\boldsymbol{x}$ implying an implausible match. "Large" is often taken to be 3, but will depend on the problem. There is also a multivariate version of $\mathcal{I}(\boldsymbol{x})$; see, for example, Vernon, Goldstein, and Bower (2010).

We performed a history match on HadCM3 using four univariate outputs calculated as the mean of the final decade of data in the ensemble. The outputs were global mean surface air temperature (SAT), the northern hemisphere meridional SAT gradient (SGRAD, measured as the difference between averages over 0–20°N and 50–70°N), the average northern

hemisphere seasonal cycle of SAT (SCYC, June–August average minus December–February average), and global mean precipitation. We used a multimodel ensemble of simulations from different AOGCMs from the World Climate Research Programme's Coupled Model Intercomparison Project phase 3 (CMIP3) (Meehl et al., 2007) and second order exchangeability (Goldstein and Wooff, 2007) to link HadCM3 to reality and to define our implausibility measure. The match ruled out 56% of the parameter space of HadCM3. Full details of this analysis and the methodology are the subject of Williamson et al. (2013).

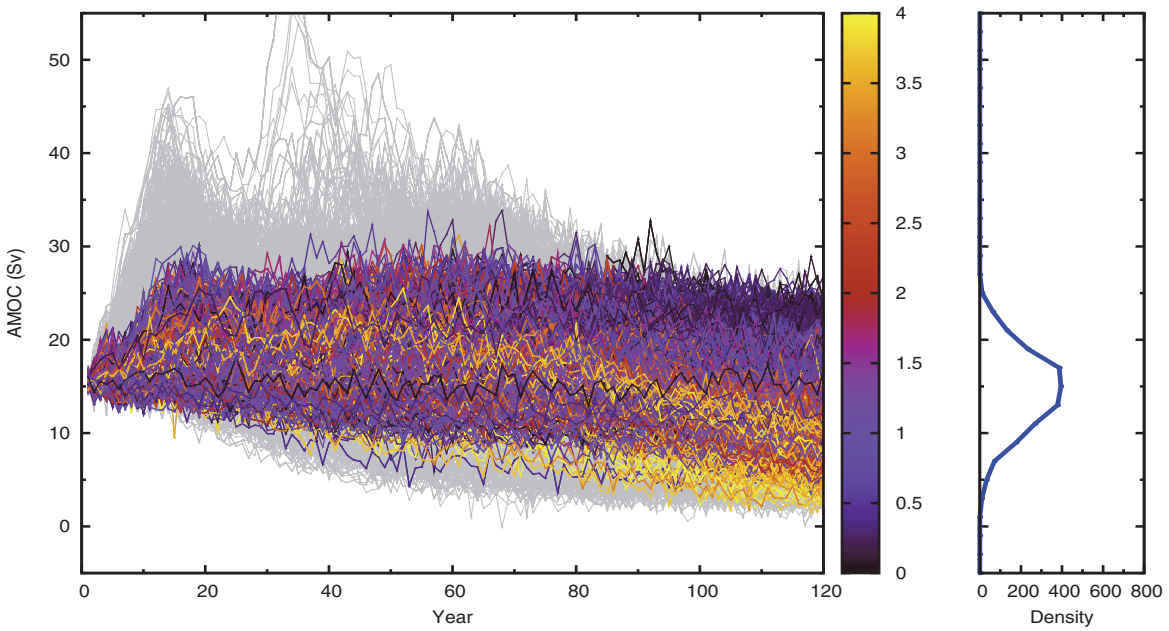

Figure 2. *Transient AMOC projections for HadCM3 in NROY space colored by CO_2 forcing. Grey lines represent ruled out ensemble members. The right-hand panel shows the density of the "final" AMOC value and was computed by binning the final 20 year means of the AMOC into bins with 2Sv intervals.*

Figure 2 shows the transient ensemble members in what we termed not ruled out yet (NROY) space, with the grey lines depicting those ensemble members that were ruled out as implausible by the history match. The NROY ensemble contains a much narrower range of AMOC behaviors and the relationship between CO_2 forcing and AMOC is visible on the plot, which implies that our modeling task will be simpler. By concentrating our efforts on building the complex emulator in NROY space we avoid problems in capturing any difficult to model behaviors that we do not need to and focus our energy on capturing those behaviors that are of interest to scientists studying the model. Note that sampling the final emulator at some point $\boldsymbol{x_0}$ must first evaluate $\mathcal{I}(\boldsymbol{x_0})$ to decide whether or not $\boldsymbol{x_0}$ is within NROY space, before evolving the emulator as described in section 2.4.

The history match identifies NROY space, a subspace of original input space which we would like to emulate the time series output of our model over. Ideally, we would design and run a new ensemble within NROY space so that the runs used to build the dynamic emulator had some desirable property. For example, we may want them to be "space filling," though

care must be taken to define what it means to "fill" unusually shaped subspaces of the initial parameter space. Williamson and Vernon (2014) have developed an efficient algorithm for obtaining large numbers of uniform random draws from NROY space. These draws can be used to select uniform subdesigns that seek to optimize any user-defined design criteria with respect to the full set of uniform samples.

In our application we are not able to design and run a new ensemble on HadCM3. As our original design was large and space filling, there is no a priori reason to think that the 44% of that design that is in NROY space would be biased towards any particular region of NROY space. However, visualizing and designing runs in complicated high-dimensional subspaces of model-input space represents a challenging open avenue of research.

3.3. Requirements of the emulator. The HadCM3 AMOC exhibits structured chaos, which, in this case, means that we might expect very slight perturbations to the model parameters to result in output that retains the same global properties in terms of location and longer term trajectory. However, the annual fluctuations may be very different, with the direction of spikes entirely arbitrary. There may be interesting "events" or significant deviations from the trend in the original or perturbed model, and these may occur at different times. It is important that our emulator does not interpolate the runs in our ensemble so that we do not model what is internal variability as parameter-dependent signal.

However, we would like our emulator draws to retain the character of an AMOC time series as output from the model. Although the direction of spikes is not important due to internal variability, that is not to say that there are not regions of parameter space in which the character of that variability is such that interesting "events" are more frequent or more likely. We can have more confidence in our ability to explore the probability of rapid changes to the AMOC using an emulator of HadCM3 if our emulator appears to evolve like an AMOC time series as output by HadCM3. We will discard the first 40 years of the time series as the spin up phase and emulate the 80-year transient time series as a function of the parameters and forcing.

4. Building the dynamic emulator. In order to construct the emulator (2) with (3), (4), (5) and the specified normal inverse gamma prior for $\boldsymbol{\psi}_0, v_0$, we require a great deal of model and parameter specification. Some of these are common to all challenges in emulating computer models, for example, choosing $\boldsymbol{g}(\boldsymbol{x_t})$, the proportion of noise in the residual, Δ, and the form of $R(|x-x'|;\boldsymbol{\rho})$ and how to handle $\boldsymbol{\rho}$. We must also specify variance discount parameters, δ_v, δ_w, and the hyperparameters of the normal inverse gamma distribution $\boldsymbol{m_0}, C_0^*, n_0$, and d_0.

Just as when building ordinary emulators, it is useful to start by specifying the mean function. In this case, that means specifying the order of the TVAR process and the elements of $\boldsymbol{g}(\boldsymbol{x_t})$. As with the emulation of simple univariate quantities, regression and other standard data-analysis techniques can be used to select functions to be used in $\boldsymbol{g}(\cdot)$. We can perform these exploratory analyses using summaries of parts of the time series or using other output from the computer model.

In our application we make use of the control simulations and choose terms in $\boldsymbol{g}(\cdot)$ to capture both the level of the AMOC at the start of the run and its parameter-dependent response to increase in CO_2 concentrations. To capture the level of the AMOC, we take the mean of the last decade of the discarded first 40 years of all simulations returned from CPDN

($\approx$ 4800 runs). We then use ordinary least squares (OLS) to model these as a function of the eight parameters having the largest effect on the model AMOC.

The selected parameters include five that control aspects of the cloud schemes, among them thickness, distribution, and type of clouds formed (these parameters are vf1, ct, cwland, rhcrit, and eacf), a parameter called entcoef that controls how rapidly convective clouds entrain surrounding air, a parameter called kappa0_si controlling the rate of vertical mixing of water in the HadCM3 ocean, and a parameter called dyndiff that is part of the specification of the dynamics of each model run. The terms in the regression fitted in the eight parameters described above are included in $\boldsymbol{g}(\boldsymbol{x_t})$.

Although we control CO_2 concentrations with one decision parameter, this parameter is used to compute a time series of CO_2 concentrations that is then written to a forcing file for use by HadCM3. We choose to regress on these changing CO_2 concentrations rather than on the decision parameter, as this is more natural in a dynamic linear model setting.

We use returned match pairs to gauge the response to forcing. There were 358 runs where both the transient and control simulation had returned 120 years of data. For each of these pairs we smoothed the control using a cubic polynomial in t and subtracted the smooth control AMOC from the transient AMOC. This smoothing is designed to make sure that the anomaly has variability similar to that of the transient AMOC. We then use stepwise selection to fit a regression to the final year anomaly with no intercept allowing a cubic signal in CO_2 concentration at year 120 and all first and second order interactions with the parameters and the CO_2. The selected model had 17 terms, involving the CO_2 concentration and the other parameters. Each of these was added to $\boldsymbol{g}(\boldsymbol{x_t})$. The terms can be seen in Table 1.

Though internal variability is modelled through Δ_t, there are modes of variability on longer time scales that are of interest to our collaborators. These are typically on time scales of 7–20 years (Allison et al., 2012). We allow the autoregression to capture the behavior on shorter time scales by setting $p = 7$. If our time series were longer, we might choose $p = 20$ and explore these modes more thoroughly; this is planned for the future as the ensemble continues to develop.

The elements of $\boldsymbol{m_0}$ and C_0^* correspond to prior means and variances on the coefficients on the time varying autoregression and the dynamic regression. We set the terms in $\boldsymbol{m_0}$ corresponding to the autoregression to be $1/7$ and set the relevant diagonal elements of C_0^* to be 4 to allow flexibility of the autoregression and to allow individual terms to change sign. We set the terms in $\boldsymbol{m_0}$ and C_0^* relating to those regression terms selected using OLS on the large ensemble of 40 year runs using the expected value and covariance matrix of these coefficients calculated from the OLS fit.

For the remaining terms that model the response to CO_2 forcing we take the view that the method and amount of data used to select the terms in the dynamic regression were not sufficient to have confidence in the fitted coefficients and covariance matrix. We therefore set the remaining terms in $\boldsymbol{m_0}$ to 0 and increase the variance and decrease the correlation of the covariance matrix calculated from the least squares fit before fixing the corresponding terms in C_0^*. We do this by halving the correlations and by doubling the standard deviations in the covariance matrix of the least squares fit. We believe this is preferable to including a diagonal matrix in C_0^*, as our exploratory analysis may have revealed strongly correlated terms in our regression and, although we do not trust the strength of these correlations, we judge

that indicating that there is a positive/negative correlation between two terms in our prior modelling is worthwhile. The remaining terms in C_0^*, corresponding to correlations between the dynamic regression terms and the autoregression terms as well as correlations between those terms involving CO_2 concentration and those not, are set to zero.

For the distribution of v_0 we wanted to allow it to be as data driven as possible. To prevent n_t increasing very rapidly in t, we fix $n_0 = n/(1-\delta_v)$ and allow the variance to be controlled by d_t, whose update equation involves the data values. We set $d_0 = n_0/4$; however, experiments showed that the filtered values for $d_{1:T}$ were not sensitive to this choice.

Choosing the discount factors is a problem common to any dynamic linear modelling with variance discounting. For our emulator we used previous experience with the ensemble. Williamson and Allison (2012) fitted separate simple dynamic linear models to ensemble members and found that values of δ_w between 0.93 and 0.97, depending on the CO_2 rate, produced fits that were appropriately smooth in that application. The models fitted in that application were much less complex and, to ensure against oversmoothing in this application, we set $\delta_w = 0.9$. $\delta_v = 0.8$ was chosen to allow the variance more flexibility to change in time if appropriate. For example, it may be the case that the variance changes as CO_2 increases.

We set $\Delta = 0.5$ so that we judge that half of the residual from our dynamic mean function represents internal variability and uncertainty due to inactive variables. We choose the separable exponential correlation function for the Gaussian process, namely

$$R(|x - x'|; \boldsymbol{\rho}) = \prod_{i=1}^{r} \exp\{-\kappa_i |x_i - x_i'|^{\alpha_i}\},$$

with $\boldsymbol{\rho} = \{\boldsymbol{\kappa}, \boldsymbol{\alpha}\}$ and r the number of variables in x. We follow Bayarri et al. (2007) and fix $\alpha_i = 1.9$ for $i = 1, \ldots, r$ instead of the more common choice of 2 in the computer experiment literature, leading to a "rougher" Gaussian process than is often fitted.

There are many benefits to fixing the roughness lengths $\boldsymbol{\kappa}$, the principal one being a simplification of the Bayesian calculations, both in general emulation problems and specifically in our dynamic approach. For example, if $\boldsymbol{\kappa}$ is fixed, our posterior distribution for the model parameters can be obtained analytically and we can sample from the emulator as described above. An argument is often made that if the mean function is sufficiently well fitted, the correlation in the residual will be very small and the contribution of $\boldsymbol{\kappa}$ to the overall uncertainty in the emulator will be negligible. Indeed this is one of the principal reasons for fitting complex mean functions. In standard emulation, there are heuristics for selecting $\boldsymbol{\kappa}$ based on the order of the monomial terms in the mean function (Vernon, Goldstein, and Bower, 2010; Williamson, Goldstein, and Blaker, 2012).

Though we suspect that the complexity of our mean function will make the correlation in the residual small, an inexperience with this type of emulator leads us to be cautious. It is not clear how the presence of the TVAR terms in our model will impact upon the roughness lengths. The way the functions in $\boldsymbol{g}(\boldsymbol{x_t})$ have been chosen leads us to be uncertain as to the quality of the fit of the mean function as CO_2 forcing begins and we step through time. This means that we may have more correlation in the residual than we hoped. We explore the effect of changing $\boldsymbol{\kappa}$ by choosing a prior distribution for the correlation lengths and implementing an MCMC sampling algorithm.

5. MCMC algorithm. We adapt the MCMC algorithm described in Liu and West (2008) to our augmented model. The algorithm is a block metropolis within a Gibbs sampler. The sampler has two steps: At the ith iteration do the following:

1. Sample $\boldsymbol{\kappa}_{1:r}^{(i)}$ from $P(\boldsymbol{\kappa}|\boldsymbol{\psi}_{1:T}^{(i)}, v_{1:T}^{(i)}, \boldsymbol{F_{1:T}})$.
2. Sample $\boldsymbol{\psi}_{1:T}^{(i)}$, $v_{1:T}^{(i)}$ from $P(\boldsymbol{\psi}_{1:T}, v_{1:T}|\boldsymbol{F_{1:T}}, \kappa_{1:r}^{(i)})$.

Step 2 proceeds as described in sections 2.3 and 2.4, by forward filtering and backwards sampling. Step 1 is achieved through a metropolis step.

Assuming

$$P(\boldsymbol{\kappa}_{1:r}|\boldsymbol{\psi}_{1:T}^{(i)}, v_{1:T}^{(i)}, \boldsymbol{F_{1:T}}) \propto P(\boldsymbol{F_{1:T}}|\boldsymbol{\psi}_{1:T}, v_{1:T}, \boldsymbol{\kappa}_{1:r})P(\boldsymbol{\kappa}_{1:r})$$

and letting $L(\kappa) = P(\boldsymbol{F_{1:T}}|\boldsymbol{\psi}_{1:T}, v_{1:T}, \kappa_{1:r})$, the metropolis step is as follows. Let $\upsilon_j = \log(\kappa_j)$ for $j = 1, \ldots, r$, and, at iteration i, propose

$$\upsilon_j^* = \upsilon_j^{(i-1)} + \mathrm{N}(0, \Omega_j^2),$$

where the Ω_j are chosen depending on the behavior of the corresponding parameter.

The Jacobian for this transformation is

$$J(\boldsymbol{\kappa}) = \prod_{j=1}^{r} \frac{1}{\kappa_j},$$

so that the acceptance probability for υ^* is

$$\frac{L(\boldsymbol{\kappa}_{1:r}^*)P(\boldsymbol{\kappa}_{1:r}^*)J(\boldsymbol{\kappa}_{1:r}^{(i-1)})}{L(\boldsymbol{\kappa}_{1:r}^{(i-1)})P(\boldsymbol{\kappa}_{1:r}^{(i-1)})J(\boldsymbol{\kappa}_{1:r}^*)}.$$

The likelihood is

$$L(\boldsymbol{\kappa}_{1:r}) = |\tau\Sigma + \Delta\mathbb{I}_n|^{-\frac{T}{2}} \exp\left\{-\frac{1}{2}\sum_{t=1}^{T} Z_t/v_t\right\} \prod_{t=1}^{T} v_t^{-\frac{n}{2}},$$

with

$$Z_t = (\boldsymbol{F_t} - H_t^T\boldsymbol{\psi}_t)^T(\tau\Sigma + \Delta\mathbb{I}_n)^{-1}(\boldsymbol{F_t} - H_t^T\boldsymbol{\psi}_t).$$

5.1. Roughness length prior. We use the concept of the "half correlation length" which has formed the basis of heuristics used to fix correlation lengths (Vernon, Goldstein, and Bower, 2010; Williamson and Goldstein, 2012), and has been used by Higdon et al. (2008a) to define a prior distribution over the correlation lengths. The idea, for each roughness length κ_j, is to consider the correlation between $\boldsymbol{x}$ and $\boldsymbol{x}'$ if all elements of $\boldsymbol{x}$ and $\boldsymbol{x}'$ are the same except the jth and if $|x_j - x_j'|$ is half of the range of x_j.

For variables defined on $[0, 1]$, this defines a half length correlation, s_j, with

$$s_j = \exp\{-\kappa_j/2^{1.9}\}.$$

We choose a prior different from that of Liu and West (2008) on s_j, by letting $s_j \sim \mathrm{Be}(1, b_j)$. This allows us control in specifying which variables we think are more or less likely to have

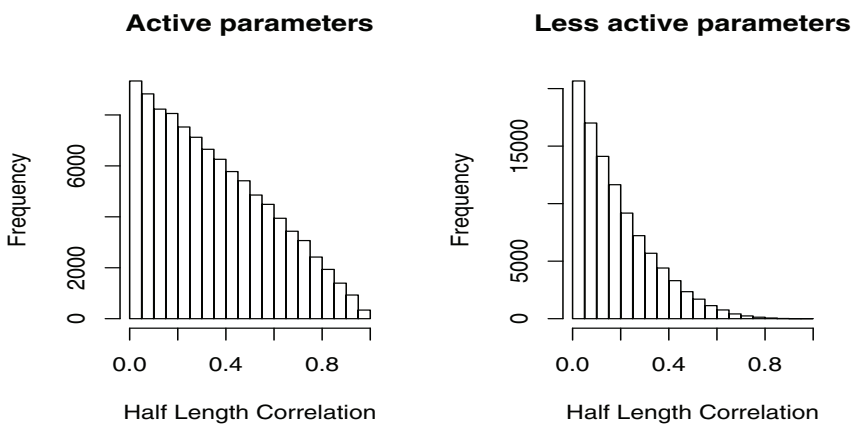

Figure 3. *Histograms for the half length correlation distributions specified by $b_j = 1.9$ (left) and $b_j = 4.5$ (right).*

a larger half length correlation through the one parameter b_j. The resulting prior on the roughness lengths is

$$P(\kappa_j) \propto \exp\left\{-\kappa_j/2^{1.9}\right\}\left(1-\exp\left\{-\kappa_j/2^{1.9}\right\}\right)^{b_j-1}.$$

For variables that were relatively inactive, in that only linear terms were included in $\boldsymbol{g}(\boldsymbol{x_t})$, we chose $b_j = 4.5$ so that a priori there is roughly a 10% chance that the corresponding half length correlation is greater than 0.4. These variables were kappa0_si and dyndiff. For the rest of the parameters, including the CO_2 rate, we chose $b_j = 1.9$ so that there is roughly a 10% chance of the half length correlation being greater than 0.7. The two prior distributions on the half length correlations are shown as histograms in Figure 3.

5.2. Model interpretation. Our interpretation of the output of this MCMC analysis depends very much upon how we view our statistical model. A belief that the chosen climate model output really did follow a dynamic linear model Gaussian process with the specification given above and uncertainty on the collection $\{\boldsymbol{\psi}_{1:T}, v_{1:T}, \kappa_{1:r}\}$ would lead us to interpret our MCMC algorithm as the search for the true settings of the parameters. The posterior distributions correspond to our uncertainty over what the settings of these parameters should be, given the model output and our other prior modelling choices.

Alternatively, if we view the statistical model as a useful tool for expressing our uncertainty about the climate model output, but recognize that the climate model is not really a dynamic linear model Gaussian process, then we do not view $\{\boldsymbol{\psi}_{1:T}, v_{1:T}, \boldsymbol{\kappa}_{1:r}\}$ as having true settings at all. In this interpretation, good choices of the parameters lead to a distribution over the model output that we believe forms an accurate picture of our uncertainty about the output. We can then view the MCMC as the search for these good choices.

The distinction is important. On the one hand, our uncertainty on the output is derived from uncertainty about what the true parameters of our dynamic linear model Gaussian process (DLMGP) should be. On the other, the DLMGP is viewed as a tool with good

choices of the parameters leading to uncertainty on the output that we are prepared to adopt and report as our own.

In the first case, in order to evaluate the uncertainty in the output for any particular value $\boldsymbol{x}$, we must run the MCMC algorithm, discarding burn in and thinning as appropriate until we judge that our samples are approximately independent and we have covered the distribution of the parameters adequately well. For each setting of the parameters in the sample, we can evolve a sample $\boldsymbol{f}(\boldsymbol{x})$. If we want a sample for a new $\boldsymbol{x}$, say $\boldsymbol{x}'$, either we use the existing MCMC output and accept that our sampled $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{x}')$ are not independent, or we conduct a new MCMC analysis for $\boldsymbol{x}'$.

Neither option is attractive in situations where the whole point of the emulator is to allow fast surrogates of the climate model to be used in order to explore its behavior in parameter space. Indeed, for many applications of emulators, a statistical model that requires an MCMC sample in order to evaluate the uncertainty on $\boldsymbol{f}(\boldsymbol{x})$ for any $\boldsymbol{x}$ is not fit for purpose.

If we hold to the second interpretation, that good choices of the parameters lead to a distribution over the model output that we can adopt as reflecting our uncertainty, then the MCMC becomes a useful tool for exploration. Once good choices of the parameters are found, we can pick one of these choices and fix the roughness lengths at this setting. This has the benefit of making samples from our emulator fast, by avoiding the MCMC every time we want to sample $\boldsymbol{f}(\boldsymbol{x})$ at a new $\boldsymbol{x}$.

We can also use the output of the MCMC to perform a sensitivity analysis on our choice of the roughness lengths. Having used the analysis to select a good choice of roughness lengths, we can also select choices consistent with our prior specification from the tails of the sampled distribution. These choices can be compared with our current best choice using more standard emulator diagnostics such as "leave one out" plots. We do this for our application in section 6.

6. Results and diagnostics. We used the regressors fitted on the controls that were used to select functions to enter into $\boldsymbol{g}(\cdot)$ to generate values for times $(1-p), \ldots, 0$, so that we did not lose almost 10% of the time series we were trying to emulate. In many problems we may have previous time series values that are not spinning up and these could be used instead. Else, our time series may be longer and removing the first p points will not affect the inference. The problem of deciding how to perform inference for the first p values for an autoregression is common to many problems in time series analysis.

We set $\Omega_{1:9} = (0.1, 0.1, 0.1, 0.5, 0.1, 0.1, 0.1, 0.1, 0.7)$ based on experience with earlier test runs showing that these choices led to good mixing. We ran the Markov chain for $840,000$ iterations, discarding the first 1000 as burn in and thinning every 400. The histograms of the posterior samples are shown in Figure 4. From top left to bottom right, the panels correspond to the marginal samples for vf1, ct, cwland, rhcrit, eacf, entcoef, kappa0_si, dyndiff, and CO_2 rate. From this figure we can see that the Gaussian process residual is most correlated for close values of the CO_2 rate and cloud parameter rhcrit, with all other parameters contributing very little to the correlation.

We select a member of the sample with high posterior density by visually inspecting the location of a large number of samples on each of the marginal posterior distributions shown in Figure 4 and selecting a sample that is simultaneously close to the mode of each marginal. We then fix the roughness lengths at this choice (represented by the blue vertical lines in Figure

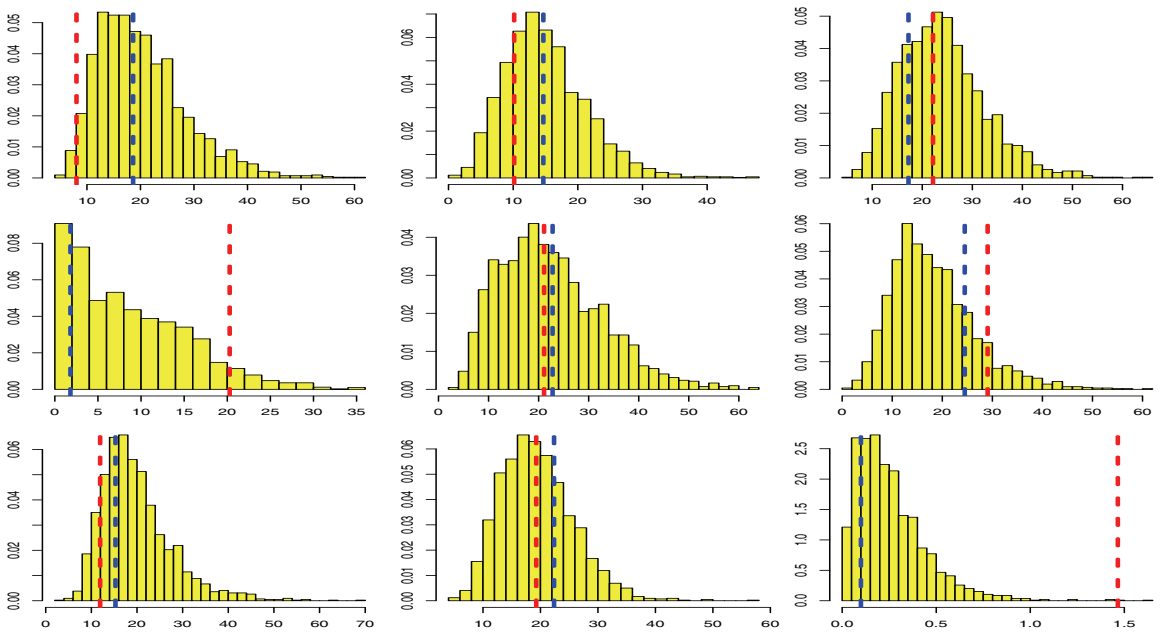

Figure 4. *Histograms from the MCMC sample for roughness lengths on each parameter. The vertical blue lines represent the roughness lengths chosen to complete the emulator. The vertical red lines are a consistent alternative used in a sensitivity analysis. From left to right the roughness lengths correspond to vf*1*, ct, cwland, rhcrit, eacf, entcoef, kappa*0*_si, dyndiff, and* CO_2 *rate.*

4). Having fixed the roughness lengths we can perform more standard emulator diagnostics. Figure 5 shows leave one out plots for 20 of the ensemble members. For each panel in the figure, the dark blue line represents the true value of HadCM3 in the ensemble. We train the emulator using the other ensemble members and draw 100 samples from the posterior distribution (the cyan lines). The pink line is the forecast function, the mean of the emulator derived by integrating out $\boldsymbol{\psi}_{1:T}$ and $v_{1:T}$. Figure 5 shows that our emulator is able to capture various different behaviors as the model evolves in time. We are able to capture the mean level and the different responses to CO_2 forcing in different parts of NROY parameter space.

We explore the sensitivity of our emulator to alternative consistent choices of the roughness lengths by performing similar diagnostics with a member of the MCMC sample that is in the tails of the distribution. This member is shown as the red vertical line in Figure 4 and was chosen so that the sample was in the tails of the most active parameters in terms of the correlation function of the Gaussian process residual, rhcrit, and CO_2 rate. The leave one out diagnostics are shown in Figure 6. The lack of difference between the diagnostics for these two choices of the roughness lengths leads us to conclude that our uncertainty is not sensitive to choices for the roughness lengths and that our autoregression and dynamic regression components have soaked up most of the signal from the model parameters.

We are also interested in whether or not our emulator captures the character of the variability of HadCM3 so that draws from the emulator look like output from the climate model. In Figure 7 we take four HadCM3 runs and compare four draws from the posterior distributions of emulators trained on the rest of the ensemble. The character of the variability for

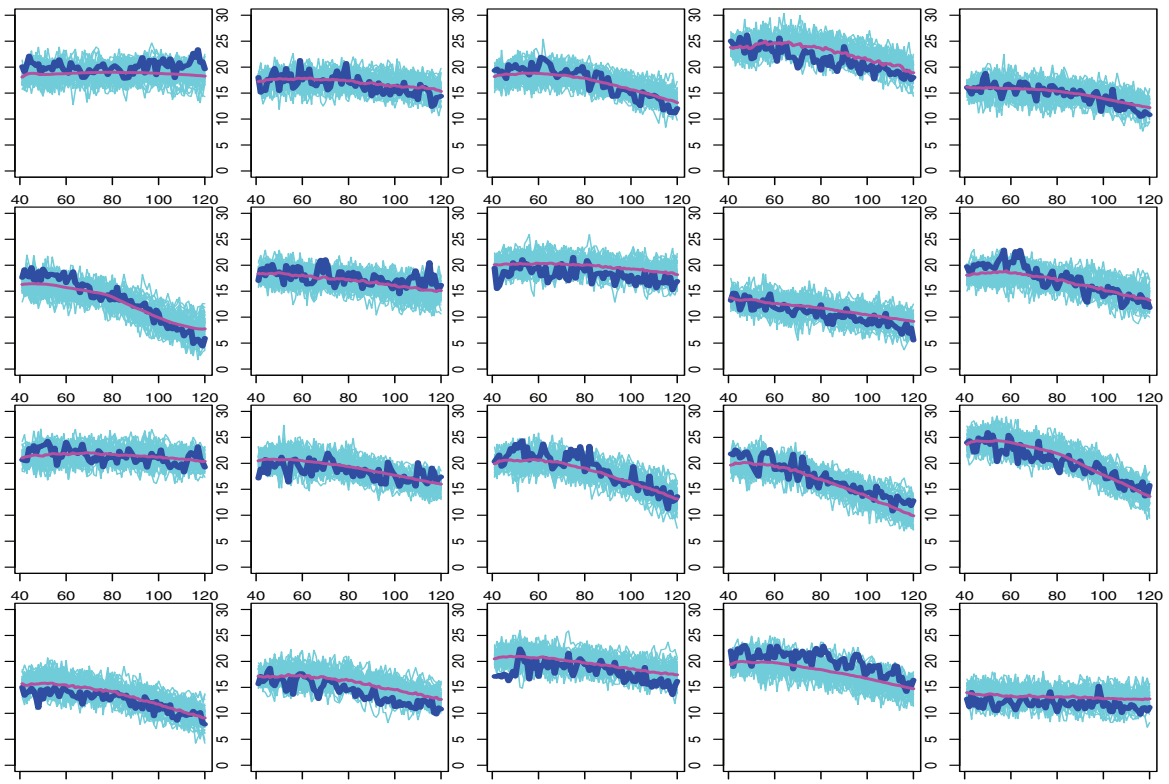

Figure 5. *Leave one out plots for* 20 *of the ensemble members. In each panel the dark blue line is a run from the climate model that the emulator has not seen. The cyan lines are draws from the posterior distribution of the emulator. The pink line is the forecast function (the mean of the emulator with uncertainty in* $\psi_{1:T}$ *and* $v_{1:T}$ *integrated out).*

the draws is such that they are indistinguishable from HadCM3 output, enabling us to use such emulators in future studies on the frequency of "rapid events" or "rapid changes" in the climate model.

7. Discussion. We have developed DLMGPs for the emulation of time series output of computer models that may exhibit structured chaos. Our method extends the method of Liu and West (2008) so that the resulting emulator has some of the key features shown in the UQ literature to be important, including a complex mean function to soak up as much variability due to the model parameters as possible and a nugget process to handle sampling uncertainty and inactive variables.

We have used this methodology to build an emulator for the time series output of an atmosphere-ocean coupled climate model, HadCM3. The emulator captures a range of parameter-dependent temporal behaviors. Due to the way that the samples are evolved, the variability characteristics of samples from the emulator are the same as those from output of the climate model, so that emulator draws and computer model runs look similar.

We present an MCMC algorithm for sampling from the posterior of our emulator when the roughness lengths are unknown. We argue for an interpretation of the DLMGP as a

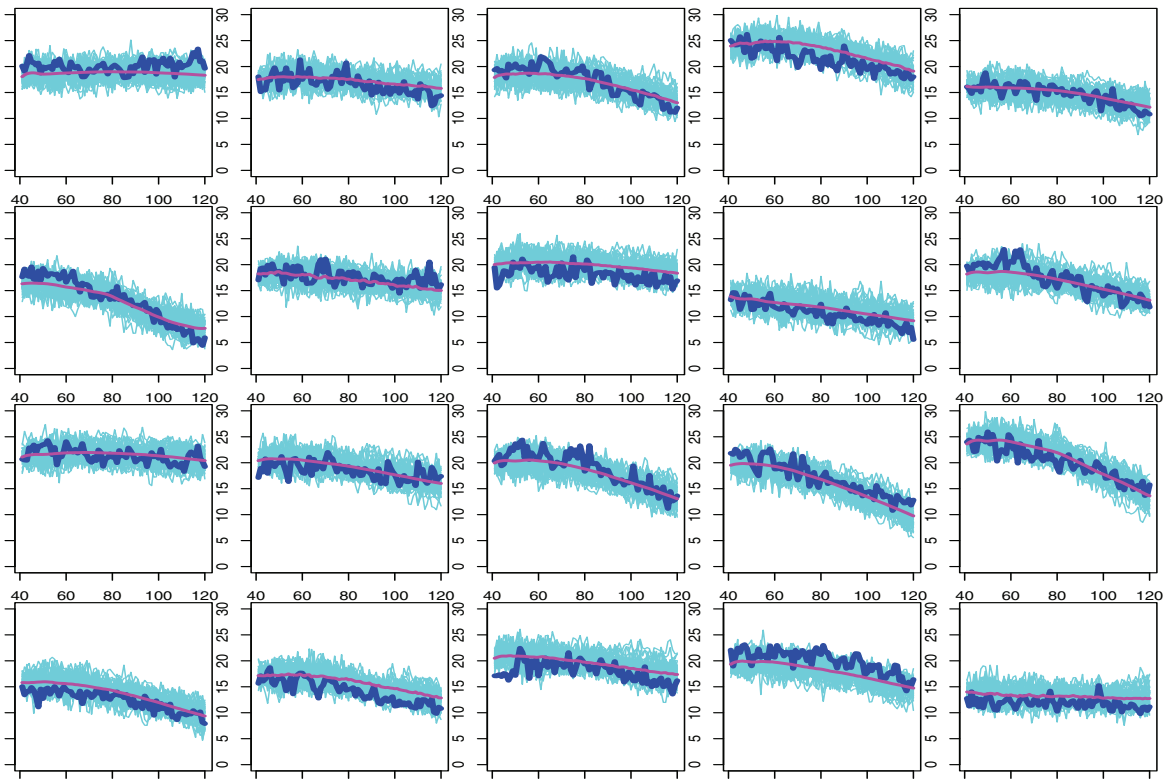

Figure 6. *Leave one out plots for* 20 *of the ensemble members for the alternative setting of the roughness lengths. In each panel the dark blue line is a run from the climate model that the emulator has not seen. The cyan lines are draws from the posterior distribution of the emulator. The pink line is the forecast function (the mean of the emulator with uncertainty in* $\psi_{1:T}$ *and* $v_{1:T}$ *integrated out).*

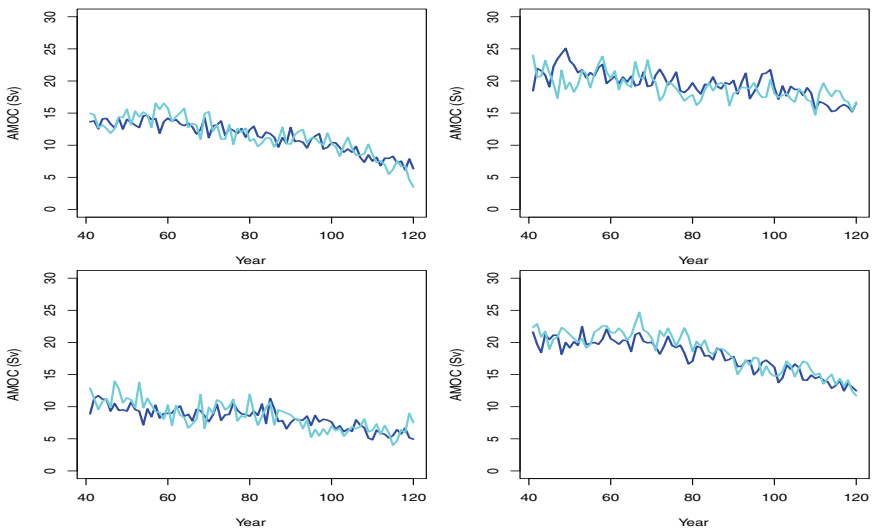

Figure 7. *Here we show four different climate model runs (dark blue) and a sample from the emulator predicting each run (cyan). The emulator has not been trained using the model runs shown.*

useful construct leading to uncertainty on the model output that we can adopt as our own. We then argue that it is natural, under this interpretation, to use the MCMC sample to fix the roughness lengths and to explore the sensitivity of our conclusions to the choice of roughness lengths using different, but consistent, values. Fixing the roughness lengths makes the emulator fit for many practical purposes that an emulator requiring an MCMC to sample from is not.

The model we have developed allows a framework in which statisticians familiar with emulators and other UQ methods can borrow strength from the extensive Bayesian time series literature. The dynamic regression and Gaussian process elements are able to handle parameter-dependent trends and form the "familiar" component of the emulator for the UQ practitioner. The TVAR process allows temporal behaviors to be captured, with more complex forms of the TVAR process able to capture harmonics of multiple frequencies across multiple time series so that known model dynamics can be captured (Prado and West, 2010). Combining these elements of the time series literature with emulators offers a promising avenue of research in UQ.

In cases where the dynamics of the underlying computer model are well understood, an alternative, more sophisticated dynamic emulator might be constructed by replacing the TVAR and regression elements of the emulator with simplified versions of the actual computer code. Reichert et al. (2011) call this "mechanism-based emulation." They argue that some deterministic computer code can be cast as a nonlinear or differential equation-based state space model that might be simplified through linearization. This linearized form of the model would replace the TVAR and dynamic regression components of (2). Though this approach is currently suitable only when the underlying computer model equations are known and simple enough to be manipulated, there may be potential in developing a mixture of the two approaches for complex models such as those found in climate science.

Appendix A. Bayesian update. Throughout this section we make repeated use of the well-known result that if $\theta|v \sim \mathrm{N}(m, vC)$ and $v^{-1} \sim \mathrm{G}(n/2, nS/2)$, then $\theta \sim T_n(m, SC)$. Here we derive the forward filtering equations of section 3. Proving the recurrence relationships follows West and Harrison (1997) (Chapters 2 and 10) adapted to our specific model and proceeds by induction. First, we have $\boldsymbol{\psi}_{t-1}|v_{t-1}D_{t-1} \sim \mathrm{N}(\boldsymbol{m_{t-1}}, v_{t-1}C^*_{t-1})$ and, to facilitate conjugacy, we assume that the variance is updated in the prior, so $\boldsymbol{\psi}_{t-1}|v_{t-1}D_{t-1} \sim \mathrm{N}(\boldsymbol{m_{t-1}}, v_tC^*_{t-1})$. Then

$$\begin{aligned}
\mathrm{E}\left[\boldsymbol{\psi}_t|v_t, D_{t-1}\right] &= \boldsymbol{m_{t-1}} = \boldsymbol{a_t}, \\
\mathrm{Var}\left[\boldsymbol{\psi}_t|v_t, D_{t-1}\right] &= v_t\left(C^*_{t-1} + W_t\right) \\
&= v_t\frac{C^*_{t-1}}{\delta_w} = v_tR^*_t
\end{aligned}$$

and

$$\mathrm{E}\left[\boldsymbol{F_t}|v_t, D_{t-1}\right] = H_t^T\boldsymbol{a_t}, \qquad \mathrm{Var}\left[\boldsymbol{F_t}|v_t, D_{t-1}\right] = v_t(H_t^TR^*_tH_t + \tau\Sigma + \Delta\mathbb{I}_n).$$

This implies $(\boldsymbol{F_t}|v_t, D_{t-1}) \sim \mathrm{N}(\boldsymbol{h_t}, v_tQ^*_t)$, with $\boldsymbol{h_t} = H_t^T\boldsymbol{a_t}$ and $Q^*_t = H_t^TR^*_tH_t + \tau\Sigma + \Delta\mathbb{I}_n$. Defining $R_t = S_{t-1}R^*_t$ and $Q_t = S_{t-1}Q^*_t$ gives $\boldsymbol{F_t}|D_{t-1} \sim T_{\delta_v n_{t-1}}(\boldsymbol{h_t}, Q_t)$, as given in the main text.

Noting that $\mathrm{Cov}\left[\boldsymbol{\psi}_t, \boldsymbol{F_t}|v_t\right] = v_t R_t^* H_t$ and performing the Bayesian update leads to

$$\mathrm{E}\left[\boldsymbol{\psi}_t|v_t, D_t\right] = \boldsymbol{a_t} + A_t \boldsymbol{e_t} = \boldsymbol{m_t}, \qquad A_t = R_t H_t Q_t^{-1}$$

and

$$\begin{aligned}\mathrm{Var}\left[\boldsymbol{\psi}_t|v_t, D_t\right] &= v_t\left(R_t^* - R_t^* H_t Q_t^{*-1} H_t^T R_t^{*T}\right)\\ &= v_t\left(R_t^* - A_t Q_t^* A_t^T\right) = v_t C_t^*,\end{aligned}$$

so that $(\boldsymbol{\psi}_t|v_t, D_t) \sim \mathrm{N}(\boldsymbol{m_t}, v_t C_t^*)$ and $(\boldsymbol{\psi}_t|D_t) \sim T_{n_t}(\boldsymbol{m_t}, C_t)$ with

$$C_t = S_t C_t^* = \frac{S_t}{S_{t-1}}\left(R_t - A_t Q_t A_t^T\right).$$

The update for ϕ_t by D_t follows by Bayes' theorem, as

$$P(\phi_t|D_{t-1}) \propto \phi_t^{\delta_v n_{t-1}/2-1} \exp\{-\delta_v d_{t-1}\phi_t/2\}$$

and

$$\begin{aligned}P(\boldsymbol{F_t}|\phi_t, D_{t-1}) &\propto \left|\frac{Q_t^*}{\phi_t}\right|^{-\frac{n}{2}} \exp\left\{\frac{\phi_t}{2}\boldsymbol{e_t}^T Q_t^{*-1}\boldsymbol{e_t}\right\}\\ &\propto \phi_t^{\frac{n}{2}} \exp\left\{-\frac{\phi_t}{2}\boldsymbol{e_t}^T Q_t^{*-1}\boldsymbol{e_t}\right\},\end{aligned}$$

and by Bayes' theorem

$$P(\phi_t|D_t) \propto \phi_t^{\frac{\delta_v n_{t-1}+n}{2}-1} \exp\left\{-\frac{\phi_t}{2}(\delta_v d_{t-1} + \boldsymbol{e_t}^T Q_t^{*-1}\boldsymbol{e_t})\right\},$$

so that $(\phi_t|D_t) \sim \mathrm{G}(n_t/2, d_t/2)$ with $n_t = \delta_v n_{t-1} + n$ and

$$d_t = \delta_v d_{t-1} + \boldsymbol{e_t}^T Q_t^{*-1}\boldsymbol{e_t} = \delta_v d_{t-1} + S_{t-1}\boldsymbol{e_t}^T Q_t \boldsymbol{e_t}.$$

A.1. Sampling distributions. We first show that the scheme for sampling $v_{1:T}$ samples from the correct distribution expanding a proof in Chapter 10 on page 363 of West and Harrison (1997). Clearly the first sample, that of $v_T|D_T$, is taken from the correct distribution. Now by Bayes' theorem

$$P(\phi_{t-1}|\phi_t D_t) \propto P(\phi_{t-1}|D_{t-1})P(\phi_t|D_{t-1}),$$

with

$$P(\phi_{t-1}|D_{t-1}) \propto \phi_{t-1}^{\frac{n_{t-1}}{2}-1} \exp\{-d_{t-1}\phi_{t-1}/2\},$$

and, using (5),

$$\begin{aligned}P(\phi_t|\phi_{t-1}, D_{t-1}) &= P_{\gamma_t}\left(\frac{\phi_t\delta_v}{\phi_{t-1}}\right)\frac{\delta_v}{\phi_{t-1}}\\ &\propto \phi_t^{\delta_v n_{t-1}/2-1}\phi_{t-1}^{-\delta_v n_{t-1}/2}\left(1 - \frac{\delta_v\phi_t}{\phi_{t-1}}\right)^{(1-\delta_v)n_{t-1}/2-1}\\ &\propto \phi_t^{\delta_v n_{t-1}/2-1}\phi_{t-1}^{1+(\delta_v-1)n_{t-1}/2-\delta_v n_{t-1}/2}\left(\phi_{t-1} - \delta_v\phi_t\right)^{(1-\delta_v)n_{t-1}/2-1}\\ &\propto \phi_t^{\delta_v n_{t-1}/2-1}\phi_{t-1}^{1-n_{t-1}/2}\left(\phi_{t-1} - \delta_v\phi_t\right)^{(1-\delta_v)n_{t-1}/2-1}.\end{aligned}$$

Then

$$\begin{aligned}
P(\phi_{t-1}|\phi_t D_t) &\propto \phi_{t-1}^{n_{t-1}/2-1+1-n_{t-1}/2} \exp\{-d_{t-1}\phi_{t-1}/2\}\phi_t^{\delta_v n_{t-1}/2-1} \left(\phi_{t-1}-\delta_v\phi_t\right)^{(1-\delta_v)n_{t-1}/2-1} \\
&\propto \phi_t^{\delta_v n_{t-1}/2-1} \exp\{-d_{t-1}\phi_{t-1}/2\} \left(\phi_{t-1}-\delta_v\phi_t\right)^{(1-\delta_v)n_{t-1}/2-1} \\
&\propto \exp\{-d_{t-1}\phi_{t-1}/2\} \left(\phi_{t-1}-\delta_v\phi_t\right)^{(1-\delta_v)n_{t-1}/2-1} \\
&\propto \exp\{-d_{t-1}\phi_{t-1}/2\} \exp\{\delta_v\phi_t d_{t-1}/2\} \left(\phi_{t-1}-\delta_v\phi_t\right)^{(1-\delta_v)n_{t-1}/2-1} \\
&\propto \exp\{-(\phi_{t-1}-\delta_v\phi_t)d_{t-1}/2\} \left(\phi_{t-1}-\delta_v\phi_t\right)^{(1-\delta_v)n_{t-1}/2-1}.
\end{aligned}$$

So, defining $\eta_{t-1} = \phi_{t-1} - \delta_v\phi_t$, then $\eta_{t-1} \sim \mathrm{G}((1-\delta_v)n_{t-1}/2-1, d_{t-1}/2)$ and we can obtain samples from $(\phi_{t-1}|\phi_t, D_t)$ by sampling an η_{t-1} and computing $\phi_{t-1} = \eta_{t-1} + \delta_v\phi_t$, which is the algorithm described in the main text.

Turning our attention to the backwards sampling scheme for $\boldsymbol{\psi}_{1:T}|v_{1:T}$, we have already shown that $\boldsymbol{\psi}_T|v_T, D_T \sim \mathrm{N}(\boldsymbol{m_T}, v_T C_T^*)$ when deriving the forward filtering equations. This proof adapts the method shown in Chapter 4 of West and Harrison (1997) to our model. Suppose $\boldsymbol{\psi}_t|v_t, D_t \sim \mathrm{N}(\boldsymbol{m_t}, v_t C_t^*)$. By Bayes' theorem

$$P(\boldsymbol{\psi}_{t-1}|\boldsymbol{\psi}_t, D_t, v_{1:T}) = \frac{P(\boldsymbol{\psi}_{t-1}|\boldsymbol{\psi}_t, D_{t-1}, v_{1:T})P(\boldsymbol{F_t}|\boldsymbol{\psi}_{t-1}, \boldsymbol{\psi}_t, D_{t-1}, v_{1:T})}{P(\boldsymbol{F_t}|\boldsymbol{\psi}_t, D_{t-1}, v_{1:T})},$$

and due to the Markov property of the time series the denominator cancels with the second half of the numerator so that

$$P(\boldsymbol{\psi}_{t-1}|\boldsymbol{\psi}_t, D_t, v_{1:T}) \propto P(\boldsymbol{\psi}_{t-1}|D_{t-1}, v_{1:T})P(\boldsymbol{\psi}_t|\boldsymbol{\psi}_{t-1}D_{t-1}, v_{1:T}).$$

Now, from the forward filtering calculations we have

$$\begin{aligned}
(\boldsymbol{\psi}_{t-1}|D_{t-1}, v_{1:T}) &\sim \mathrm{N}(\boldsymbol{m_{t-1}}, v_{t-1}C_{t-1}^*) \\
(\boldsymbol{\psi}_t|D_{t-1}, v_{1:T}) &\sim \mathrm{N}(\boldsymbol{a_t}, v_{t-1}R_t/S_{t-1}) \\
&\sim \mathrm{N}(\boldsymbol{a_t}, v_{t-1}R_t^*),
\end{aligned}$$

so, using the Bayesian normal updating equations we get

$$(\boldsymbol{\psi}_{t-1}|\boldsymbol{\psi}_t, D_{t-1}) \sim \mathrm{N}(\boldsymbol{\kappa_t}, \Xi_t)$$

with

$$\begin{aligned}
\boldsymbol{\kappa_t} &= \boldsymbol{m_{t-1}} + v_{t-1}C_{t-1}^*(v_{t-1}R_t^*)^{-1}(\boldsymbol{\psi}_t - \boldsymbol{a_t}) \\
&= \boldsymbol{m_{t-1}} + C_{t-1}^*(C_{t-1}^*/\delta_w)^{-1}(\boldsymbol{\psi}_t - \boldsymbol{m_{t-1}}) \\
&= (1-\delta_w)\boldsymbol{m_{t-1}} + \delta_w\boldsymbol{\psi}_t
\end{aligned}$$

and

$$\begin{aligned}
\Xi_t &= v_{t-1}C_{t-1}^* - v_{t-1}C_{t-1}^*(v_{t-1}R_t^*)^{-1}C_{t-1}^*v_{t-1} \\
&= v_{t-1}(C_{t-1}^* - \delta_w C_{t-1}^*) \\
&= v_{t-1}(1-\delta_w)C_{t-1}^*.
\end{aligned}$$

Appendix B. Elements of the dynamic mean function. Table 1 shows the terms used in vector $\boldsymbol{g}(\cdot)$ by the methods described in section 4. Each header in the table refers to one of the parameters. Numbers on the diagonal refer to power terms in $\boldsymbol{g}(\cdot)$ in each of the relevant parameters. The number 1 on the diagonal implies only a linear term was included in $\boldsymbol{g}(\cdot)$. The number 2 implies that both quadratic and linear terms were in $\boldsymbol{g}(\cdot)$. The number 3 implies cubic, quadratic, and linear terms.

Numbers on the upper triangle refer to the inclusion or not of interactions between the two relevant variables. For example, reading from the table, the term (rhcrit $*$ dyndiff) is included in $\boldsymbol{g}(\cdot)$, but the term (vf1 $*$ dyndiff) is not. Variables indicated in bold on the lower triangle refer to three-way interactions that are present in $\boldsymbol{g}(\cdot)$. For example, the terms (ent $*$ ct^2) and (vf1 $*$ cwland $*$ CO2) are both included in $\boldsymbol{g}(\cdot)$. Our emulator does contain a constant term, so the vector $\boldsymbol{g}(\cdot)$ includes the element 1.

Table 1

A table indicating which terms are in $\boldsymbol{g}(\cdot)$ for our dynamic emulator. The upper triangle labels which interaction pairs are present. The diagonal indicates the order of the highest monomial term in that variable. The lower triangle indicates which three-way interactions are included.

	vf1	ct	cwland	rhcrit	eacf	ent	kappa0	dyndiff	CO2
vf1	2	1	1	0	1	1	0	0	1
ct		2	1	1	1	1	1	0	1
cwland		**ct**	1	1	1	1	0	0	1
rhcrit				1	1	1	1	1	1
eacf		**ct**	**ct**		1	1	0	0	1
ent		**ct**	**ct**	**ct**	**ct**	2	1	0	1
kappa0							1	0	1
dyndiff								1	1
CO2	**cwland**	**cwland**	**ent**		**ent**	**vf1**	**ct**	**rhcrit**	2

Acknowledgments. We would like to thank the CPDN team for their work in submitting our ensemble to the CPDN users. We would also like to thank Michael Goldstein for helpful discussions and all of the CPDN users around the world who contributed their spare computing resource as part of the generation of our ensemble. Finally, we would like to thank the associate editor and the two referees for their helpful comments and suggestions for improving the paper.

REFERENCES

L. C. Allison, E. Hawkins, T. Woollings, and L. Jackson (2012), *The Value of an Event-Based Approach to Understanding Decadal Fluctuations in the Atlantic Meridional Overturning Circulation*, manuscript.

I. Andrianakis and P. G. Challenor (2012), *The effect of the nugget on Gaussian process emulators of computer models*, Comput. Statist. Data Anal., 56, pp. 4215–4228.

B. Balan Sarojini, J. Gregory, R. Tailleux, G. R. Bigg, A. T. Blaker, D. R. Cameron, N. R. Edwards, A. P. Megann, L. Shaffrey, and B. Sinha (2011), *High frequency variability of the Atlantic meridional overturning circulation*, Ocean Sci., 7, 471486.

M. J. Bayarri, J. O. Berger, J. Cafeo, G. Garcia-Donato, F. Liu, J. Palomo, R. J. Parthasarathy, R. Paulo, J. Sacks, and D. Walsh (2007), *Computer model validation with functional output*, Ann. Statist., 35, pp. 1874–1906.

A. Biastoch, C. W. Böning, J. Getzlaff, J. M. Molines, and G. Madec (2008), *Mechanisms of interannual-decadal variability in the meridional overturning circulation of the mid latitude North Atlantic Ocean*, J. Climate, 21, pp. 6599–6615.

N. L. Bindoff, J. Willebrand, V. Artale, A. Cazenave, J. Gregory, S. Gulev, K. Hanawa, C. Le Quéré, S. Levitus, Y. Nojiri, C. K. Shum, L. D. Talley, and A. Unnikrishnan (2007), *Observations: Oceanic climate change and sea level*, 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, S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis, K. B. Averyt, M. Tignor, and H. L. Miller, eds., Cambridge University Press, Cambridge, UK, pp. 385–432.

A. T. Blaker, J. J. M. Hirschi, B. Sinha, B. A. de Cuevas, S. G. Alderson, A. C. Coward, and G. Madec (2012), *Large near-inertial oscillations of the Atlantic meridional overturning circulation*, Ocean Model., 42, pp. 50–56.

A. T. Blaker and D. Williamson (2014), *Characteristics and Variability of the Atlantic Meridional Overturning Circulation in a Large Climate Model Ensemble*, manuscript.

P. Challenor, D. McNeall, and J. Gattiker (2009), *Assessing the probability of rare climate events*, in The Oxford Handbook of Applied Bayesian Analysis, A. O'Hagan and M. West, eds., Oxford University Press, Oxford, UK, pp. 403–430.

S. Conti, J. P. Gosling, J. E. Oakley, and A. O'Hagan (2009), *Gaussian process emulation of dynamic computer codes*, Biometrika, 96, pp. 663–676.

S. Conti and A. O'Hagan (2010), *Bayesian emulation of complex multi-output and dynamic computer models*, J. Stat. Plan. Inference, 140, pp. 640–651.

P. S. Craig, M. Goldstein, J. C. Rougier, and A. H. Seheult (2001), *Bayesian forecasting for complex systems using computer simulators*, J. Amer. Statist. Assoc., 96, pp. 717–729.

P. S. Craig, M. Goldstein, A. H. Seheult, and J. A. Smith (1996), *Bayes linear strategies for matching hydrocarbon reservoir history*, in Bayesian Statistics 5, J. M. Bernado, J. O. Berger, A. P. Dawid, and A. F. M. Smith, eds., Oxford University Press, New York, pp. 69–95.

P. S. Craig, M. Goldstein, A. H. Seheult, and J. A. Smith (1997), *Pressure matching for hydrocarbon reservoirs: A case study in the use of Bayes linear strategies for large computer experiments*, in Case Studies in Bayesian Statistics, Vol. III, C. Gatsonis, J. S. Hodges, R. E. Kass, R. McCulloch, P. Rossi, and N. D. Singpurwalla, eds., Springer-Verlag, New York, pp. 36–93.

N. Cressie (1993), *Statistics for Spatial Data*, John Wiley & Sons.

J. A. Cumming and M. Goldstein (2009), *Small sample designs for complex high-dimensional models based on fast approximations*, Technometrics, 51, pp. 377–388.

J. A. Cumming and M. Goldstein (2010), *Bayes linear uncertainty analysis for oil reservoirs based on multiscale computer experiments*, in The Oxford Handbook of Applied Bayesian Analysis, A. O'Hagan and M. West, eds., Oxford University Press, Oxford, UK, pp. 241–270.

C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker (1991), *Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments*, J. Amer. Statist. Assoc., 86, pp. 953–963.

P. J. Diggle and P. J. Ribeiro, Jr. (2007), *Model-Based Geostatistics*, Springer Ser. Statist., Springer, New York.

M. Goldstein and D. Wooff (2007), *Bayes Linear Statistics Theory and Methods*, John Wiley and Sons, Chichester.

C. Gordon, C. Cooper, C. A. Senior, H. Banks, J. M. Gregory, T. C. Johns, J. F. B. Mitchell, and R. A. Wood (2000), *The simulation of SST, sea ice extents and ocean heat transports in a version of the Hadley Centre coupled model without flux adjustments*, Clim. Dyn., 16, pp. 147–168.

R. Grammacy and H. Lee (2012), *Cases for the nugget in modeling computer experiments*, Stat. Comput., 22, pp. 713–722.

E. Hawkins and R. Sutton (2009), *The potential to narrow uncertainty in regional climate predictions*, BAMS, 90, pp. 1095–1107.

R. Haylock and A. O'Hagan (1996), *On inference for outputs of computationally expensive algorithms with uncertainty on the inputs*, in Bayesian Statistics 5, J. M. Bernado, J. O. Berger, A. P. Dawid, and A. F. M. Smith, eds., Oxford University Press, New York, pp. 629–637.

D. Higdon, C. Nakhleh, J. Gattiker, and B. Williams (2008a), *A Bayesian calibration approach to the*

thermal problem, Comput. Methods Appl. Mech. Engrg., 197, pp. 2431–2441.

H. Higdon, J. Gattiker, B. Williams, and M. Rightley (2008b), *Computer model calibration using high-dimensional output*, J. Amer. Statist. Assoc., 103, pp. 570–583.

W. E. Johns, M. O. Baringer, L. M. Beal, S. A. Cunningham, T. Kanzow, H. L. Brydon, J. J. M. Hirschi, J. Marotzke, C. S. Meinen, B. Shaw, and R. Curry (2011), *Continuous, array-based estimates of Atlantic Ocean heat transport at* $26.5^\circ N$, J. Clim., 24, pp. 2429–2449.

C. G. Kaufman, D. Bingham, S. Habib, K. Heitmann, and J. A. Frieman (2011), *Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology*, Ann. Appl. Stat., 5, pp. 2470–2492.

M. C. Kennedy and A. O'Hagan (2001), *Bayesian calibration of computer models*, J. R. Stat. Soc. Ser. B Stat. Methodol., 63, pp. 425–464.

J. P. C. Kleijnen (2009), *Kriging metamodelling in simulation: A review*, European J. Oper. Res., 192, pp. 707–716.

F. Liu and M. West (2008), *A dynamic modelling strategy for Bayesian computer model emulation*, Bayesian Anal., 4, pp. 393–412.

G. McCarthy, E. Frajka-Williams, W. E. Johns, M. O. Baringer, C. S. Meinen, H. L. Bryden, D. Rayner, A. Duchez, and S. A. Cunningham (2012), *Observed interannual variability of the Atlantic meridional overturning circulation at* $26.5^\circ N$, GRL, 39, L19609.

G. A. Meehl, C. Covey, T. Delworth, M. Latif, B. McAvaney, J. F. B. Mitchell, R. J. Stouffer, and K. E. Taylor (2007), *The WCRP CMIP3 multi-model dataset: A new era in climate change research*, Bull. Am. Meteorol. Soc., 88, pp. 1383–1394.

V. D. Pope, M. L. Gallani, P. R. Rowntree, and R. A. Stratton (2000), *The impact of new physical parameterizations in the Hadley Centre climate model: HadAM3*, Clim. Dyn., 16, pp. 123–146.

R. Prado and M. West (2010), *Time Series Analysis: Modeling, Computation and Inference*, CRC Press, Boca Raton, FL.

D. Rayner, J. J. M. Hirschi, T. Kanzow, W. E. Johns, P. G. Wright, E. Frajka-Williams, H. L. Bryden, C. S. Meinen, M. O. Baringer, J. Marotzke, L. M. Beal, and S. A. Cunningham (2011), *Monitoring the Atlantic meridional overturning circulation*, Deep Sea Res. Part II, 58, pp. 1744–1753.

P. Reichert, G. White, M. J. Bayarri, and E. B. Pitman (2011), *Mechanism-based emulation of dynamic simulation models: Concept and application in hydrology*, Comput. Statist. Data Anal., 55, pp. 1638–1655.

J. C. Rougier (2008), *Efficient emulators for multivariate deterministic functions*, J. Comput. Graph. Stat., 17, pp. 827–843.

J. C. Rougier, D. M. H. Sexton, J. M. Murphy, and D. Stainforth (2009), *Emulating the sensitivity of the HadSM3 climate model using ensembles from different but related experiments*, J. Clim., 22, pp. 3540–3557.

J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989), *Design and analysis of computer experiments*, Statist. Sci., 4, pp. 409–435.

T. J. Santner, B. J. Williams, and W. I. Notz (2003), *The Design and Analysis of Computer Experiments*, Springer-Verlag, New York.

D. M. H. Sexton, J. M. Murphy, and M. Collins (2011), *Multivariate probabilistic projections using imperfect climate models part* 1: *Outline of methodology*, Clim. Dyn., 38, pp. 2513–2542.

S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis, K. B. Averyt, M. Tignor, and H. L. Miller, eds. (2007), *Contribution of Working Group* I *to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change,* 2007, Cambridge University Press, Cambridge, UK.

I. Vernon, M. Goldstein, and R. G. Bower (2010), *Galaxy formation: A Bayesian uncertainty analysis*, Bayesian Anal., 5, pp. 619–846.

M. West and J. Harrison (1997), *Bayesian Forecasting and Dynamic Models*, 2nd ed., Springer, New York.

D. Williamson (2010), *Policy Making Using Computer Simulators for Complex Physical Systems; Bayesian Decision Support for the Development of Adaptive Strategies*, Ph.D. thesis, University of Durham, Durham, UK; available online from http://etheses.dur.ac.uk/348/.

D. Williamson and L. Allison (2012), *A Study of Rapid Events in the CPDN Ensemble*, Durham University Tech. Rep., University of Durham, Durham, UK; available online from http://www.maths.dur.ac.uk/users/daniel.williamson/index.html#publications.

D. Williamson and M. Goldstein (2012), *Bayesian policy support for adaptive strategies using computer*

models for complex physical systems, J. Oper. Res. Soc., 63, pp. 1021–1033.

D. Williamson, M. Goldstein, L. Allison, A. T. Blaker, P. Challenor, L. Jackson, and K. Yamazaki (2013), *History matching for exploring and reducing climate model parameter space using observations and a large perturbed physics ensemble*, Clim. Dyn., 41, pp. 1703–1729.

D. Williamson, M. Goldstein, and A. T. Blaker (2012), *Fast linked analyses for scenario based hierarchies*, J. R. Stat. Soc. Ser. C Appl. Stat., 61, pp. 665–692.

D. Williamson and I. R. Vernon (2014), *Efficient uniform designs for multi-wave computer experiments*, preprint, arXiv:1309.3520; J. Roy. Statist. Soc. Ser. B, submitted.

K. Yamazaki, D. J. Rowlands, T. Aina, A. T. Blaker, A. Bowery, N. Massey, J. Miller, C. Rye, S. F. B. Tett, D. Williamson, Y. H. Yamazaki, and M. R. Allen (2013), *Obtaining diverse behaviours in a climate model without the use of flux adjustments*, J. Geophys. Res. Atmos., 118, pp. 2781–2793.