Assessing First-Order Emulator Inference for Physical Parameters in Nonlinear Mechanistic Models

Mevin B. HOOTEN, William B. LEEDS, Jerome FIECHTER, and Christopher K. WIKLE

We present an approach for estimating physical parameters in nonlinear models that relies on an approximation to the mechanistic model itself for computational efficiency. The proposed methodology is validated and applied in two different modeling scenarios: (a) Simulation and (b) lower trophic level ocean ecosystem model. The approach we develop relies on the ability to predict right singular vectors (resulting from a decomposition of computer model experimental output) based on the computer model input and an experimental set of parameters. Critically, we model the right singular vectors in terms of the model parameters via a nonlinear statistical model. Specifically, we focus our attention on first-order models of these right singular vectors rather than the second-order (covariance) structure.

Key Words: Bayesian; Hierarchical model; Nonparametric statistics; Spatio-temporal statistics.

1. INTRODUCTION

Given the complexity of many real-world processes, mathematical models for such processes are often represented in some type of discrete setting (e.g., finite differences, spectral, Galerkin expansions) to facilitate implementation by digital computers. Often, this results in a "forward model" integration or simulation model that is thought to represent the real-world process of interest. For example, consider the type of models used in the study of ocean ecosystems that relate numerous ecosystem quantities to each other

Mevin B. Hooten (✉) is an Assistant Unit Leader and Assistant Professor, Colorado Cooperative Fish and Wildlife Research Unit, U.S. Geological Survey, Fort Collins, CO, USA (E-mail: *mevin.hooten@colostate.edu*). William B. Leeds is a PhD Candidate (E-mail: *wbl8t7@mail.mizzou.edu*) and Christopher K. Wikle is a Professor (E-mail: *wiklec@missouri.edu*), Department of Statistics, University of Missouri, Columbia, MO, USA. Jerome Fiechter is an Assistant Researcher, Ocean Sciences Department, University of California – Santa Cruz, Santa Cruz, CA, USA (E-mail: *fiechter@ucsc.edu*).

Journal of Agricultural, Biological, and Environmental Statistics, Volume 16, Number 4, Pages 475–494
DOI: 10.1007/s13253-011-0073-7

through a set of differential equations (e.g., Fiechter et al. 2009). Such models represent the best scientific understanding of how the natural process might behave, but actual numerical simulation of the ecosystem processes can be quite computationally demanding, taking minutes, hours, or even days to produce a single realization. Although it has long been known that there are various errors associated with such representations, in recent years statisticians have been drawn to the problem in order to quantify the various sources of uncertainty inherent in such approximations. Many authors have proposed approaches whereby the computer model itself could be emulated with simpler, more efficient models (e.g., Bayarri et al. 2007a, 2007b; Bliznyuk et al. 2008; Conti et al. 2009; Craig et al. 2001; Drignei 2008; Frolov et al. 2009; Higdon et al 2004, 2008; Kennedy and O'Hagan 2001; Liu and West 2009; O'Hagan 2006; Rougier 2008; Sacks et al. 1989; van der Merwe et al. 2007). Indeed, this is a well-developed literature, especially in the context of model calibration (i.e., estimation for model parameters that do not have physical or biological meaning, but serve to improve the model's "fit"). Our interest here is the development of a simple, yet flexible, model framework to facilitate inference for model parameters in nonlinear mechanistic models that do have physical or biological meaning. Because of this, we build what we call a "first-order" emulator for the mechanistic model that relates these parameters to the response. This is in contrast to the traditional computer emulation literature that focuses on "second-order" Gaussian process specification (i.e., covariance specification) as is common in the spatial statistics literature (e.g., Kennedy and O'Hagan 2001). Notable exceptions are discussed by van der Merwe et al. (2007) and Frolov et al. (2009), who use a model surrogate to facilitate fast data assimilation. In our specific approach, we consider the singular value decomposition of a collection of computer model ensembles given various parameter inputs (analogous to Higdon et al. 2008), and then model the right singular vectors as a function of the parameters via nonlinear statistical models.

For a given forward computer model (possibly a "physical" or "mechanistic" model), $f(\mathbf{X}, \boldsymbol{\theta})$, it is often of interest to recover (i.e., estimate) the scientifically meaningful $p \times 1$ vector of parameters $\boldsymbol{\theta}$ given an $n \times p$ matrix of input variables $\mathbf{X}$ and observed $n \times 1$ response vector of interest $\mathbf{y}$. The data, $\mathbf{y}$, could be univariate or multivariate, spatially and/or temporally indexed and pertain to a variety of different systems (e.g., phylogenetic, economic, energetic, ecological, environmental). A simple approach to this problem, using minimal assumptions, would be to minimize the discrepancy between the observed response data $\mathbf{y}$ and the model output $f(\mathbf{X}, \boldsymbol{\theta})$ with respect to the parameters, $\boldsymbol{\theta}$, using a loss function $l(\mathbf{y}, f(\mathbf{X}, \boldsymbol{\theta}))$ of choice. For example, assuming additive error, one might choose to reconcile the difference between $\mathbf{y}$ and $f(\mathbf{X}, \boldsymbol{\theta})$, in terms of $\boldsymbol{\theta}$, based on the L_2 norm of the residuals (i.e., using squared error loss): $l(\mathbf{y}, f(\mathbf{X}, \boldsymbol{\theta})) = \|\mathbf{y} - f(\mathbf{X}, \boldsymbol{\theta})\|^2$. With an additional Gaussian assumption and no other constraints on the support of $\mathbf{y}$, we could write this as a parametric nonlinear model where $\mathbf{y} = f(\mathbf{X}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon} \sim \mathrm{N}(\mathbf{0}, \sigma^2\mathbf{I})$. Obviously this is not the most general model specification, but it is probably the most commonly used statistical approach to the problem of "nonlinear regression" nonetheless.

Continuing with this example then, if the computer model was sufficiently simple that a nonlinear least squares approach to parameter estimation could be taken, then an iterative Taylor-series linearization method could be employed. In doing so, one would first expand

the computer model in a truncated Taylor series as $f(\mathbf{X}, \boldsymbol{\theta}) \approx f(\mathbf{X}, \boldsymbol{\theta}^*) + \mathbf{J}(f(\mathbf{X}, \boldsymbol{\theta}))(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$, where $\mathbf{J}$ refers to the Jacobian operator containing partial derivatives of f with respect to the input variables and each parameter in $\boldsymbol{\theta}$ and higher order terms are negligible. Then the model output $f(\mathbf{X}, \boldsymbol{\theta}^*)$ can be moved to the left hand side of the equation and an error term added to the right hand side to yield the linear model: $\mathbf{y} - f(\mathbf{X}, \boldsymbol{\theta}^*) = \mathbf{J}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}$, where $\boldsymbol{\beta}^* = \boldsymbol{\theta} - \boldsymbol{\theta}^*$. Thus, after choosing initial values for $\boldsymbol{\theta}^*$, nonlinear regression proceeds with iteratively obtaining new, less-biased, estimates for $\boldsymbol{\theta}$ by finding the optimal $\boldsymbol{\beta}^*$ at each iteration (often $\hat{\boldsymbol{\beta}}^* = (\mathbf{J}'\mathbf{J})^{-1}\mathbf{J}'(\mathbf{y} - f(\mathbf{X}, \boldsymbol{\theta}^*))$, converting it back to $\boldsymbol{\theta}^*$ and then substituting it in for the next round of optimization.

The method described above is an elegant approach to estimating parameters in nonlinear models, but is only available when the Jacobian can be analytically found or numerically well-approximated and the higher order terms are negligible. However, the notion of linearizing the problem through an expansion of the model output is still attractive. In what follows, we present a means for estimating nonlinear model parameters for the specific case where the scientifically motivated computer model $f(\mathbf{X}, \boldsymbol{\theta})$ is considerably computationally inefficient and the computer model parameter support is thought to be constrained. We evaluate an approach that follows the general "emulator" methods for approximating complicated computer models, but that is intended to be simpler, uses off-the-shelf nonlinear predictors, and is trivial to implement using a Markov Chain Monte Carlo (MCMC) algorithm. We illustrate this approach using two examples: the first, a simple nonlinear relationship resulting from a "power law" model, and the second, a coupled physical-biological computer model for lower trophic level ocean ecosystem dynamics.

2. METHODS

An alternative approach to parameter estimation in the simple example described above might be through Bayesian inference. In principle, by specifying a prior distribution for the physical parameters and variance component (i.e., $\boldsymbol{\theta} \sim [\boldsymbol{\theta}]$ and $\sigma^2 \sim [\sigma^2]$, where the square bracket notation corresponds to a probability density function), one could approximate the posterior distribution of the unknowns given the data (e.g., $[\boldsymbol{\theta}, \sigma^2|\mathbf{y}]$) via MCMC, regardless of the complexity in the computer model $f(\mathbf{X}, \boldsymbol{\theta})$. This approach has the advantage of easily being able to accommodate various constraints on the response and parameter support as well as prior scientific understanding concerning realistic parameter values. Such constraints and additional parameter information are commonplace in projects concerning the construction and utility of computer models. One potential disadvantage to this approach for parameter estimation is that the computer model itself may not be sufficiently efficient to run iteratively in an MCMC algorithm.

2.1. RELATED WORK

As mentioned in the introduction, this is not a new problem in statistics as many researchers have proposed approaches in which the computer model is emulated by a simpler, more efficient, model (i.e., $f^*(\mathbf{X}, \boldsymbol{\theta})$) thus alleviating the need to simulate directly from the original model $f(\mathbf{X}, \boldsymbol{\theta})$ at each MCMC iteration. With the notable exception of the work

by Bliznyuk et al. (2008), the main focus in much of the previous work was on calibration or prediction rather than parameter estimation. For example, the basic premise in the work of Higdon et al. (2008) can be imagined as modeling the dominant structure of $f(\mathbf{X}, \boldsymbol{\theta})$ in the space of $\boldsymbol{\theta}$, whereas Bliznyuk et al. (2008) take a similar approach but instead model the dominant structure of the log-posterior of $\boldsymbol{\theta}$. We follow the former approach more closely and thus consider a highly nonlinear and multidimensional surface $\mathbf{v}(\boldsymbol{\theta})$ spanning the space of $\boldsymbol{\theta}$. Now suppose that through finite experimentation with the computer model itself one can observe points on that surface v_k at locations $\boldsymbol{\theta}_k$ for $k = 1, \ldots, K$, where K is the total number of observations in the computer experiment. Then, one needs only implement a sufficiently flexible model on $\mathbf{v}$ that allows for interpolation such that for any point $\boldsymbol{\theta}^*$, in the support of $\boldsymbol{\theta}$, a prediction $\mathbf{v}^*(\boldsymbol{\theta}^*)$ can be obtained. As long as $\mathbf{v}^*(\boldsymbol{\theta}^*)$ can be transformed back into the space of $f(\mathbf{X}, \boldsymbol{\theta})$ with reasonable efficiency, the computer model can be effectively emulated.

Higdon et al. (2008) arrive at the surfaces ($\mathbf{v}$) by considering the dominant characteristics of the computer model in a Karhunen–Loève expansion of the model output. In a finite computer experiment, they simulate $\mathbf{y}^{(k)}$ for $k = 1, \ldots, K$, collect them in a matrix $\mathbf{Y} = (\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(K)})$, and then decompose the matrix as $\mathbf{Y} = \mathbf{UDV}'$ (Gentle 2007). Now the right singular vectors in the matrix $\mathbf{V}$ correspond to the aforementioned expressions, $\mathbf{v}$, of the dominant structures, $\mathbf{UD}$, in the computer model. Higdon et al. (2008) then use a sophisticated nonstationary spatial model to describe the nonlinear surfaces $\mathbf{v}$ in the space of $\boldsymbol{\theta}$. That is, they allow for flexibility in the model for $\mathbf{v}$ through second-order model structure similar to that commonly used in geostatistical applications. Thus, we refer to this result as a "second-order emulator," and seek to evaluate emulators based on first-order properties of the nonlinear surfaces $\mathbf{v}$. A possible advantage to the first-order emulator, in particular, is that its implementation may be dramatically simplified over the second-order emulator. Additionally, a first-order emulator allows for an explicit nonlinear link between the model output and the parameters. In most cases, mechanistic computer models are constructed as first-order processes.

2.2. FIRST-ORDER EMULATORS

First, consider the model approximation induced through the expansion of the experimental model output ($\mathbf{Y}$) given n-dimensional actual observed response data $\mathbf{y}$:

$$\mathbf{y} = \mathbf{UDv}(\boldsymbol{\theta}) + \boldsymbol{\varepsilon}, \tag{2.1}$$

where, for now, the errors could be modeled as $\boldsymbol{\varepsilon} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma})$ such that $\boldsymbol{\Sigma} \equiv \sigma^2\mathbf{I}$. Consider also a $K \times p$ matrix containing the complete set of experimental parameter vectors: $\boldsymbol{\Theta} = (\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(K)})'$ where p represents the dimensionality of $\boldsymbol{\theta}$. Now, since the experimental output matrix $\mathbf{Y}$ has dimension $n \times K$, the dominant computer model structures $\mathbf{UD}$ are of dimension $n \times K$ and the complete set of right singular vectors $\mathbf{V}$ have dimension $K \times K$. Thus, each column of $\mathbf{V}$ ($\mathbf{v}_i$ for $i = 1, \ldots, n$) is K dimensional. As a side note germane to the Bayesian setting, if the experimental vectors $\boldsymbol{\theta}^{(k)}$ are drawn from the prior distribution, then the columns of $\mathbf{V}$ represent expressions of the dominant orthogonal structures ($\mathbf{UD}$) in the mean of the prior predictive distribution.

At this point, we only need to develop a predictive model for $\mathbf{v}$ based on the observed $\mathbf{V}$ and $\boldsymbol{\Theta}$ from the computer model experiments. Generalizing the approach described by Higdon et al. (2008), we let $\mathbf{V} \sim g(\boldsymbol{\Theta}, \boldsymbol{\beta})$, where we now treat the experimental parameter values ($\boldsymbol{\Theta}$) as covariates and $\boldsymbol{\beta}$ as nuisance parameters in the predictive model. The choice of predictive model g is then left up to the specific modeler and/or application. Ideally, g is a capable model for prediction and is readily able to provide information on the predictive distribution of $\mathbf{v}$ for any value of $\boldsymbol{\theta}$.

2.2.1. Linear First-Order Emulator

As a simple example, consider g to be a linear Gaussian predictive model. In this case, for each singular vector $\mathbf{v}_i$ $(i = 1, \ldots, K)$, we could write $\mathbf{v}_i = \boldsymbol{\Theta}\boldsymbol{\beta}_i + \boldsymbol{\eta}_i$, where $\boldsymbol{\eta}_i \sim \mathrm{N}(\mathbf{0}, \tau^2\mathbf{I})$, and $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$ are $p \times 1$ and $K \times 1$ vectors, respectively. Then keeping in mind that this predictive model need not be Bayesian, it could be fit using a traditional least squares approach where the nuisance parameter estimates are obtained via $\hat{\boldsymbol{\beta}}_i = (\boldsymbol{\Theta}'\boldsymbol{\Theta})^{-1}\boldsymbol{\Theta}'\mathbf{v}_i$ for $i = 1, \ldots, K$. Then predictions for $\mathbf{v}$ are easily obtained for any $\boldsymbol{\theta}^*$ by $\hat{\mathbf{v}}_i = \boldsymbol{\theta}^{*\prime}\hat{\boldsymbol{\beta}}_i$. In fact, collecting all of the predictive model coefficients into a $K \times p$ matrix $\hat{\mathbf{B}} = (\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_K)'$, we can write the full model from (2.1) as

$$\mathbf{y} = \mathbf{U}\mathbf{D}\hat{\mathbf{B}}'\boldsymbol{\theta} + \boldsymbol{\varepsilon}, \tag{2.2}$$

since we could substitute $E(\mathbf{V}) = \boldsymbol{\Theta}\mathbf{B}$ into $\mathbf{Y} = \mathbf{U}\mathbf{D}\mathbf{V}'$ for $\mathbf{V}$.

As a side note, regarding the "design" matrix for this linearized model (2.2), $\mathbf{U}\mathbf{D}\mathbf{B}'$, if we assume the standard least squares estimates for $\hat{\mathbf{B}} = (\boldsymbol{\Theta}'\boldsymbol{\Theta})^{-1}\boldsymbol{\Theta}'\mathbf{V}$, we have the following result:

$$\begin{aligned}\mathbf{U}\mathbf{D}\hat{\mathbf{B}}' &= \mathbf{U}\mathbf{D}\big((\boldsymbol{\Theta}'\boldsymbol{\Theta})^{-1}\boldsymbol{\Theta}'\mathbf{V}\big)\\ &= \mathbf{U}\mathbf{D}\mathbf{V}'\boldsymbol{\Theta}\big(\boldsymbol{\Theta}'\boldsymbol{\Theta}\big)^{-1}\\ &= \mathbf{Y}\boldsymbol{\Theta}\big(\boldsymbol{\Theta}'\boldsymbol{\Theta}\big)^{-1}.\end{aligned}$$

The last equality above indicates that the modeled mean of the response data $\mathbf{y}$ is merely a weighted average of the computer model response output (i.e., $\mathbf{Y}$) resulting from the experiment. In this case, the weights are $\boldsymbol{\Theta}(\boldsymbol{\Theta}'\boldsymbol{\Theta})^{-1}\boldsymbol{\theta}$, where the latter term, $\boldsymbol{\theta}$, is to be estimated. Thus, perhaps surprisingly, the singular value decomposition is not essential for fitting the linear emulator model as long as a transformation from the coefficients, resulting from a regression of $\mathbf{y}$ on $\mathbf{Y}$, to the physical model parameters $\boldsymbol{\theta}$, exists. In this case, the projection matrix $\boldsymbol{\Theta}(\boldsymbol{\Theta}'\boldsymbol{\Theta})^{-1}$ serves as such a transformation. This begs the question: why not estimate generic coefficients, $\boldsymbol{\gamma}$, as weights using the model $\mathbf{y} = \mathbf{Y}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$ and then link $\boldsymbol{\gamma}$ to the physical parameters $\boldsymbol{\theta}$ through a secondary model using the results of the computer model experiment as data? Two possible reasons to retain the SVD specification for this model are: (1) The dimensionality of $\boldsymbol{\gamma}$ could be larger than n. (2) It may be possible to expedite a fitting algorithm by truncating both $\mathbf{U}\mathbf{D}$ and $\mathbf{V}$ such that fewer model fits pertaining to $\mathbf{V}$ are required. This latter notion could be formalized by choosing a number q, such that $q < n$, of right singular vectors to fit and then truncate the matrix $\mathbf{U}\mathbf{D}$

such that it only contains the first q columns, say $\widetilde{\mathbf{UD}}$. Then, having only q coefficient vectors in $\tilde{\mathbf{B}}$ (where $\tilde{\mathbf{B}}$ is $q \times p$), one would proceed by fitting the reduced model as before: $\mathbf{y} = \widetilde{\mathbf{UD}}\tilde{\mathbf{B}}'\boldsymbol{\theta} + \boldsymbol{\varepsilon}$. Though this sacrifices some adherence to the original computer model f, it could improve computational efficiency by orders of magnitude.

Returning to the model specification in (2.2), assuming that $\mathbf{B}$ is known or well-estimated, $\boldsymbol{\theta}$ can then be estimated using either frequentist or Bayesian linear model methods. This form of linearized model has both advantages and disadvantages. Due to the linear assumptions linking $\mathbf{V}$ and $\boldsymbol{\Theta}$, it will be very efficient computationally, but will also be biased if the relationships between right singular vectors and computer model parameters are not linear. Additionally, the variance associated with the estimated $\boldsymbol{\theta}$ (from either a Bayesian or frequentist standpoint) will not be correct unless the relationship between $\mathbf{V}$ and $\boldsymbol{\Theta}$ is deterministic. Thus, these two issues need to be resolved (at perhaps some computational cost) if reliable computer model parameter inference is desired.

Suppose, for the moment, that the linear assumption in the model for $\mathbf{V}$ is valid, then one natural way to formally recognize the uncertainty pertaining to this aspect of the larger model would be through a hierarchical specification. That is, consider the full model in matrix notation:

$$\begin{aligned}\mathbf{y} &= \mathbf{UDv} + \boldsymbol{\varepsilon},\\ \mathbf{v} &= \mathbf{B}'\boldsymbol{\theta} + \boldsymbol{\eta},\end{aligned}$$

or, collapsed into a single model:

$$\begin{aligned}\mathbf{y} &= \mathbf{UDv} + \boldsymbol{\varepsilon}\\ &= \mathbf{UD}\big(\mathbf{B}'\boldsymbol{\theta} + \boldsymbol{\eta}\big) + \boldsymbol{\varepsilon}\\ &= \mathbf{UDB}'\boldsymbol{\theta} + \mathbf{UD}\boldsymbol{\eta} + \boldsymbol{\varepsilon},\end{aligned}$$

where the variance associated with $\boldsymbol{\eta}$ (i.e., τ^2) is known (or at least estimated) since the original $\mathbf{V}$ is a result of the computer model experiment. In terms of the errors $\boldsymbol{\eta}$ and $\boldsymbol{\varepsilon}$, since $E(\boldsymbol{\varepsilon})$ and $E(\boldsymbol{\eta})$ are both equal to zero in practice (though it is possible that there may exist some non-zero bias that could be accommodated), and $\operatorname{var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$, the orthogonal structure of $\mathbf{UD}$ then yields $\operatorname{var}(\mathbf{UD}\boldsymbol{\eta}) = (\mathbf{UD})'\tau^2\mathbf{I}(\mathbf{UD}) = \tau^2\mathbf{I}$. One could then describe the efficiency of the predictive model by assessing the ratio $\rho = \frac{\tau^2}{\tau^2+\sigma^2}$; where small values of ρ indicate a better predictive model and thus a closer approximation to the original computer model.

2.2.2. Nonlinear First-Order Emulator

Clearly, the linear assumption between the right singular vectors, $\mathbf{v}_i$, and $\boldsymbol{\Theta}$, is strong and, depending on the computer model, probably not reasonable. Thus, we seek a more general statistical link between the singular vectors and experimental physical parameter values. With no obvious parametric form available and a desire to make this portion of the emulator "automatic," a nonparametric or semiparametric method could be valuable. That is, as described at the beginning of this section, let our model for the singular vectors

be defined generically as $\mathbf{V} \sim g(\boldsymbol{\Theta}, \boldsymbol{\beta})$, where the coefficients $\boldsymbol{\beta}$ need not have a physical interpretation, but the model itself should be an excellent predictor of $\mathbf{v}^*$ given some $\boldsymbol{\theta}^*$. Now the model for $\mathbf{y}$ can be rewritten as

$$\mathbf{y} = \mathbf{UDv} + \boldsymbol{\varepsilon},$$
$$\mathbf{v} \sim g(\boldsymbol{\theta}, \boldsymbol{\beta}).$$

The main disadvantage to using this nonlinear emulator, over the linear one previously described, is that inference on $\boldsymbol{\theta}$ is not as straightforward. Asymptotic inference on $\boldsymbol{\theta}$ will be intractable if the predictive model g is complicated. Bayesian methods become powerful in this setting because, in addition to the advantages previously discussed, the inference on the physical parameters $\boldsymbol{\theta}$ is scalable with sample size and always available as long as the posterior distribution of $\boldsymbol{\theta}$ can be found.

The posterior distribution of interest in this case can be written as

$$\left[\boldsymbol{\theta}, \sigma^2, \mathbf{v} \middle| \mathbf{y}\right] \propto \left[\mathbf{y} \middle| \mathbf{v}, \sigma^2\right][\mathbf{v}|\boldsymbol{\theta}, \boldsymbol{\beta}][\boldsymbol{\theta}]\left[\sigma^2\right][\boldsymbol{\beta}], \tag{2.3}$$

where the square bracket notation denotes a probability distribution and each of the distributions for single parameters correspond to priors. Typically, this posterior distribution would be computed using an MCMC algorithm, which requires the model for $\mathbf{v}$ to be Bayesian. Moreover, identifiability issues may arise in the joint estimation of $\mathbf{v}$, $\boldsymbol{\theta}$, and $\boldsymbol{\beta}$. To allow for more flexibility in the choice of predictive models we propose a two-stage implementation where the model $\mathbf{V} \sim g(\boldsymbol{\Theta}, \boldsymbol{\beta})$ is fit using the computer model experiment results and then employed for prediction of $\mathbf{v}^*$ at any physical parameter values $\boldsymbol{\theta}^*$. At this point the idea is similar to that described in the preceding linear emulator section, but we wish to accommodate the uncertainty associated with the fit of this nonlinear predictive model in the estimation of $\boldsymbol{\theta}$. Additionally, we are not interested in inference on $\mathbf{v}$, but rather only $\boldsymbol{\theta}$ and σ^2 given the observed data $\mathbf{y}$. Thus, we seek to find the integrated posterior:

$$\left[\boldsymbol{\theta}, \sigma^2 \middle| \mathbf{y}\right] \propto \int \left[\mathbf{y} \middle| \mathbf{v}, \sigma^2\right][\mathbf{v}|\boldsymbol{\theta}][\boldsymbol{\theta}]\left[\sigma^2\right] d\mathbf{v}, \tag{2.4}$$

where $[\mathbf{v}|\boldsymbol{\theta}]$ corresponds to the predictive model for $\mathbf{v}$. The advantage of this approach is that, in an MCMC setting (specifically with Metropolis–Hastings updates), the integration in (2.4) can be achieved through composition sampling. In practice, for a given proposal of $\boldsymbol{\theta}$, say $\boldsymbol{\theta}^*$, we need only be able to simulate a realization, say $\mathbf{v}^*$, from the predictive distribution $[\mathbf{v}|\boldsymbol{\theta}^*]$. This realization is then used in the likelihood portion of the Metropolis–Hastings ratio which is in turn used to accept or reject the proposed $\boldsymbol{\theta}^*$, as usual. If the prior distribution for the variance component σ^2 is then specified to be inverse gamma, a conjugate full-conditional distribution can be used to obtain a Gibbs update in the MCMC algorithm. Since the fitting of the predictive model g occurs prior to the fitting of the main model, computational efficiency of this approach relies merely on the ability of being able to simulate quickly from the predictive model. An alternative way to think of the distribution for the predictive model $[\mathbf{v}|\boldsymbol{\theta}]$ is as a stochastic transformation such that the physical parameters $\boldsymbol{\theta}$ can be modified to have the required singular vector interpretation

and support. Using the form in (2.4), we are able to "integrate over" the $\mathbf{v}$ term in such a way that the uncertainty related to the stochastic transformation is accommodated. This integration ensures that the variance associated with the physical parameters is inflated appropriately based on the accuracy of the predictive model.

2.3. IMPLEMENTATION

Note, in general, any reasonable (and flexible) nonlinear model framework could be used to approximate $g(\boldsymbol{\theta}, \boldsymbol{\beta})$. For purposes of illustration, we explored the use of the nonparametric bagged regression tree approach called "random forests" (e.g., Breiman 2001) to link the right singular vectors $\mathbf{v}_i$ to the set of experimental physical parameter values $\boldsymbol{\Theta}$. In doing so, one would fit q separate random forest models to each of the singular vectors $\mathbf{v}_i$ for $i = 1, \ldots, q$; let these be denoted $g_i(\boldsymbol{\Theta}, \boldsymbol{\beta})$, where $\boldsymbol{\beta}$ correspond to the set of nuisance parameters controlling the structure of the trees in the random forests. Note that we use the term "model" loosely here, in that these nonparametric predictive methods are averaging over an ensemble of regression trees resulting from various random subsets of the data and "fitting" based on a minimization of the "out-of-bag" prediction error (e.g., cross-validation; Hastie, Tibshirani, and Friedman 2009). Even though there is no explicit "model," in the parametric sense, assumed with these methods, they still treat the data as observed random variables and implicitly assume there is *some* probability distribution they arise from. Thus, we describe these predictive methods as models, but this may be somewhat ambiguous.

As discussed in the previous sections, we then treat this nonlinear predictive model as a stochastic transformation, which when given input, say $\boldsymbol{\theta}^*$, it yields output in terms of $\mathbf{v}^*$. Ideally, we would obtain $\mathbf{v}^*$ as a realization from its predictive distribution, however the inherent sparse model structure in the random forest method does not allow for this. Thus, in order to approximate samples from the desired distribution, we first obtain the residuals, $\hat{\boldsymbol{\eta}}^*$, over the entire set of training data $\boldsymbol{\Theta}$. We then select a bootstrap sample of residuals, $\boldsymbol{\eta}^*$, from $\hat{\boldsymbol{\eta}}^*$, and add them to the random forest predictions, $\hat{\mathbf{v}}^*$, for a proposed physical parameter vector, $\boldsymbol{\theta}^*$, to yield the quasi-realizations $\mathbf{v}^* = \hat{\mathbf{v}}^* + \boldsymbol{\eta}^*$. This approach has several advantages, the first being that it is based on an average of out-of-bag predictions (Cutler et al. 2007). That is, when the random forest procedure is applied to a set of data, it builds numerous regression trees based on various subsets of the original data. Since some of the data (approximately one-third) will be left out of each regression tree, predictions of the data then are based on regression trees where the data were not used for fitting. An average of those predictions has been shown to have low bias and low variance (Hastie, Tibshirani, and Friedman 2009) and the resulting residuals, $\hat{\boldsymbol{\eta}}^*$, will not be the traditional residuals based on a difference between the observations and fitted values, but rather, true predictive errors. Then, the advantage to constructing the predictions, $\hat{\mathbf{v}}^*$, at a new proposed value of $\boldsymbol{\theta}^*$ is that, since they were not in any of the bootstrap samples from which the regression trees were fit, the predictions themselves are based on an average over a much larger set of trees, thus reducing the variance further. Due to the construction of the quasi-realizations, $\mathbf{v}^*$, we now are able to achieve the closest possible thing to samples from the predictive distribution of interest while making as few assumptions as possible about

the form of that distribution. An alternative approach to accommodate the uncertainty in the nonlinear model for $\mathbf{v}$ would be to choose an appropriate distribution for $\mathbf{v}$ such that $\mathbf{v} \sim [\mathbf{v}|\hat{\mathbf{v}}^*(\boldsymbol{\theta}^*), \sigma_v^2]$ for a given parameter vector $\boldsymbol{\theta}^*$, prediction $\hat{\mathbf{v}}^*$, and hyperparameter σ_v^2. This approach yields a proper hierarchical probability model but also adds a layer of computational complexity as the state variables $\mathbf{v}$ now must be sampled separately from $\boldsymbol{\theta}$. We retain the former quasi-realization approach because it makes fewer assumptions about the probability structure and is more computationally efficient.

The last step is then to specify the overall emulator model: $\mathbf{y} = \widetilde{\mathbf{UD}}\mathbf{v}(\boldsymbol{\theta}) + \boldsymbol{\varepsilon}$, for $\boldsymbol{\varepsilon} \sim \mathrm{N}(\mathbf{0}, \sigma^2 \cdot \mathbf{I})$, where the priors in this case are $\boldsymbol{\theta} \sim \mathrm{TN}(\mathbf{0}, \sigma_\theta^2 \cdot \mathbf{I})_L^U$ (where L and U refer to lower and upper bounds on the support and could be unbounded to yield a Gaussian distribution if desired) and $\log(\sigma) \sim \mathrm{N}(0, \sigma_\sigma^2)$. In fitting this model from a Bayesian perspective we simply need to sample from the full-conditional distributions $[\boldsymbol{\theta}|\cdot]$ and $[\sigma^2|\cdot]$ iteratively in an MCMC algorithm. Due to the inherent nonlinearity in the random forest predictive model and the nonconjugacy of the priors, we use a Metropolis–Hastings update for each. The update for σ^2 is straightforward and proceeds as it would for a variance component in any other Bayesian model. The update for $\boldsymbol{\theta}$ only requires that we sample a set of physical parameter values $\boldsymbol{\theta}^*$ from a proposal distribution (a random walk in this case) and use those in conjunction with the random forest predictor to obtain a quasi-realization $\mathbf{v}^*(\boldsymbol{\theta}^*)$. This realization is then used in the Metropolis–Hastings ratio as the set of "proposed" values for $\mathbf{v}$, and by doing so, we are able to perform the necessary integration to account for the uncertainty pertaining to the nested stochastic transformation which will result in more accurate inference on the physical parameters $\boldsymbol{\theta}$.

To illustrate the simplicity of this first-order emulator approach, suppose the real data and computer model or its output are available and consider the following sequence of steps for implementation:

(1) Sample parameter sets in either a random or more structured manner from their support (unless already made available along with computer model output).

(2) Obtain output from the computer model using each parameter set as input. If the computer experiment is not adaptive, then this task could be parallelized, further increasing computational efficiency.

(3) Decompose matrix of computer model output to obtain singular values and vectors. Retain those that explain a large portion of the variation in the model output.

(4) Regress, perhaps nonparametrically, the remaining right singular vectors on the parameter vectors (as covariates) from the computer experiment and save resulting predictors.

(5) Within an MCMC algorithm, propose values for the parameter vector.

(6) Obtain predictions (or predictive realizations if possible) for the right singular vectors given the proposed parameters.

(7) Evaluate the Metropolis–Hastings ratio and use it to accept or reject the proposed parameters.

(8) Repeat steps 5–7 until enough MCMC samples are available to make the desired inference.

3. APPLICATIONS

3.1. EXAMPLE: POWER LAW MODEL

We constructed a very simple nonlinear example to illustrate the methods presented herein. Although we recognize that this is a trivial model that is very much amenable to traditional nonlinear regression methods, it provides an illustration of the plausibility of our proposed emulator methodology. Thus, given that power laws are commonly used in studies of ecological processes, we focused our "mechanistic" model on the form $y = \theta_1 x^{\theta_2}$, where x can be thought of as a model input or covariate, y represents the response, and θ_1 and θ_2 are the physical parameters of which we desire estimates. One reason that this model form is appealing is because model fits can be obtained in many different ways. For example, using the nonlinear least squares approach described in the introduction, one would first assume an additive error model for y and then minimize the squared difference between the data y and the model output $\theta_1 e^{\theta_2}$ with respect to a squared error loss function. Alternatively, one may choose to log transform both sides to yield the linear model $\log(y) = \log\theta_1 + \theta_2 \log(x)$, where an additive error assumption on this log-linear model implies a multiplicative error on the original model. In what follows, we use the former conventional method (i.e., nonlinear least squares) to provide comparative fits of the power law model, though it should be noted that there is an ongoing debate in the literature as to which approach for error specification is most appropriate and under what conditions (e.g., Xiao et al. 2011). For the example shown here, we drew x_i (for $i = 1, \ldots, 100$) independently from a uniform distribution on [0, 1] support, set physical parameters at $\theta_1 = 2.25$ and $\theta_2 = 0.5$, and used Gaussian additive error with variance $\sigma^2 = (0.1)^2$. We then sampled $\mathbf{y}$ from a Gaussian distribution with mean $\theta_1 \mathbf{x}^{\theta_2}$ and variance σ^2 to simulate the data that we will later use to fit the model.

To set up the emulator model, as described in the previous section, we first performed the computer experiment by sampling $N = 1{,}000$ physical process parameters independently from uniform distributions on realistic scientific supports [0, 5] and [0, 1], for θ_1 and θ_2, respectively. These parameter values were chosen to represent common forms of power laws observed in nature (e.g., Xiao et al. 2011). This procedure resulted in the 100×1000 experiment output matrix $\mathbf{Y}$ which was then decomposed into $\mathbf{UDV}'$ using the singular value decomposition. The matrix $\mathbf{V}$ was then comprised of 100 right singular vectors, of which $q = 3$ were retained for predictive modeling (accounting for 99.9% of the variation in $\mathbf{Y}$). For priors, we used nearly flat truncated Gaussians for θ_1 and θ_2, centered and truncated on the support [0, 5] and [0, 1], respectively. For $\log(\sigma)$, we used a variance of 1, which for this problem, provides a fairly diffuse prior on the variance component σ^2.

We ran the MCMC algorithm described in Section 2.3 for a series of 10,000 iterations and discarded the first 1,000 as burn-in samples. Calculating posterior statistics based on the remaining samples results in the values shown in Table 1. We can see from Table 1 that our emulator method performs quite well as compared with a conventional NLS fit.

Table 1. True physical parameters values, NLS estimates, posterior means, and 95% credible intervals from emulator model fit for the power law simulation.

Parameter	Truth	NLS Est.	Emulator Post. Mean	Emulator Post. 95% CI
θ_1	2.25	2.22	2.307	(2.146, 2.450)
θ_2	0.5	0.51	0.506	(0.451, 0.557)

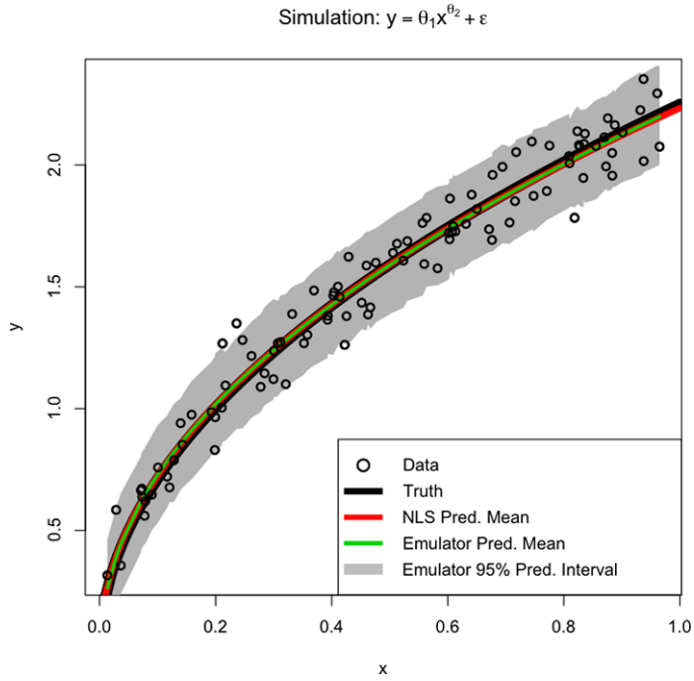

Figure 1. Power law model predictions. Notice that both the emulator and NLS predictive means coincide with the true mean.

Obviously, if a more complicated mechanistic model were used, the NLS approach would not be available. Part of the reason we use the power law model to illustrate our emulator approach before more complicated physical process models is so that we could validate it by comparing fits from other methods.

We also obtained the posterior predictive distribution for the fitted emulator model and compared that to the NLS predictions. We can see from Figure 1 that the model predictions are quite close to the truth and that the predictive interval has approximately the correct coverage; that is, approximately 5 of the 100 observations fall outside the 95% prediction interval.

3.2. EXAMPLE: COUPLED PHYSICAL-BIOLOGICAL MODEL (ROMS-NPZDFE)

The parameter estimation results for the simple nonlinear power law model are encouraging, but the main utility of the nonlinear first-order emulator is to fit more complicated

and computationally inefficient forward physical models that cannot be iterated over in a reasonable amount of time. Thus, for this example, we focus on lower trophic level ocean ecosystem dynamics and an associated coupled physical-biological model that mimics this system. As each forward simulation for this model requires approximately 30 minutes to obtain and time was limited on our server for this study, we explore the model space with a computer experiment based on 50 randomly selected parameter values ($\boldsymbol{\Theta}$) and associated sets of model output ($\mathbf{Y}$). After assessing the emulator model strengths and weaknesses using a "leave-one-ensemble-member-out" cross-validation, we use the full computer experiment output to fit a model using real data based on satellite ocean color observations from SeaWiFS (Sea-viewing Wide Field-of-view Sensor).

The 50-member computer experiment ensemble was generated using the ROMS-NPZDFe coupled physical-biological model for the Coastal Gulf of Alaska (CGOA; from ca. 50 to 62 degrees N and 140 to 164 degrees W). The ocean circulation component is an implementation of the Regional Ocean Modeling System (ROMS; Haidvogel et al. 2008), and the lower trophic level ecosystem component is provided by a six-compartment Nutrient-Phytoplankton-Zooplankton-Detritus (NPZD) model (Powell et al. 2006) with iron limitation (NPZDFe; Fiechter et al. 2009).

ROMS is a hydrostatic, primitive equation model that uses a terrain-following vertical coordinate and a split-mode technique to efficiently solve for the baroclinic and barotropic components of the circulation. The model grid has a horizontal resolution of about 11 km and 42 non-uniform vertical levels. The NPZDFe model is used to express the lower trophic level ecological processes in the pelagic region of the CGOA. The processes described by the governing equations include autotrophic growth of phytoplankton controlled by light, a single macro-nutrient (nitrate), and a micro-nutrient (iron). Natural loss processes for phytoplankton and zooplankton via mortality and remineralization of detritus and iron are linear in P, Z, D, and Fe, respectively. The lower trophic level ecosystem model is coupled to the ocean circulation model by solving a transport equation in ROMS for each of the NPZDFe model compartments at every time step. A vertical sinking velocity is also imposed on detritus.

The CGOA shelf region undergoes significant physical and biological variability on monthly, seasonal, and interannual timescales due to the different oceanographic processes impacting the region (e.g., Stabeno et al. 2004). Primary production (chlorophyll) variability is dominant on seasonal timescales with a strong spring bloom typically occurring in April–May, followed by a weaker fall bloom in September-October. Chlorophyll concentrations are comparatively weaker during summer, and drop to very low levels during winter because of severe light limitation on phytoplankton growth. On interannual timescales, oceanic mesoscale processes (Crawford, Brickley, and Thomas 2007) and atmospheric forcing (Fiechter and Moore 2009) have been identified as contributors to variability in primary production in the CGOA. Due to this connection between chlorophyll (which can be measured remotely) and phytoplankton (which is the quantity modeled), the computer model output of interest is the 'P' in the acronym NPZDFe (that is, in our notation $\mathbf{Y} \equiv f(\mathrm{P})$, for some function f).

To perform the ROMS-NPZDFe computer model experiment, the 50-member ensemble was generated by choosing 7 of the 17 biological parameters to vary randomly: the phy-

Table 2. Biological parameter bounds for the ROMS-NPZDFe Model.

Parameter	Lower Bound	Upper Bound
PhyIS	0.010	0.030
Vm NO3	0.500	1.500
K NO3	0.500	1.500
ZooGR	0.325	0.975
DetRR	0.500	1.500
K FeC	8.450	25.35
FeRR	0.250	0.750

toplankton maximum growth rate (`Vm_NO3`), half-saturation constant for nitrate (`K_NO3`) and iron (`K_FeC`) and light response (`PhyIS`); the zooplankton maximum grazing rate (`ZooGR`); and the detritus and iron remineralization rates (`DetRR` and `FeRR`, respectively). To avoid generating unrealistic solutions, each parameter value for a given member in the ensemble was sampled from a truncated Gaussian distribution with mean parameter μ_j (for $j = 1, \ldots, p$) and standard deviation parameter $\sigma_j = 0.25\mu_j$). To avoid generating unrealistic solutions, the parameter support in the truncated normal was restricted to $\mu_j \pm 0.5\mu_j$; these parameter supports are shown in Table 2. The ensemble output was then decomposed as described in Section 2, to yield the left and right singular vectors and singular values. In this case, we found that $q = 3$ singular vectors accounted for approximately 99% of the variation in our computer model experiment and thus we truncated the **UD** and **V** matrices accordingly. It is important to note that even though the degree of truncation improves computation there may be higher order structure in the process that gets neglected by omitting the less important singular vectors. The result is that there may be less statistical learning for any parameters controlling that high order structure and is one of the major disadvantages to emulation based on Karhunen–Loève expansions of the model output.

For this emulator model, we make one change to the overall specification presented in the previous sections. Given that the observed data represents biological concentrations, it has naturally non-negative support and hence we specify a truncated Gaussian likelihood that is left-truncated at zero. Now, the data model can be represented as $\mathbf{y} \sim \text{TN}(\mathbf{UDv}, \sigma^2)_0^\infty$, where we still assume that $\mathbf{v} \sim g(\boldsymbol{\theta}, \boldsymbol{\beta})$ and use a random forest predictor for g. The truncated Gaussian is appealing in this situation because it is more general than the Gaussian and allows the variance to decrease near the boundary, aiding the model fit at small values of $\mathbf{y}$ (e.g., Cangelosi and Hooten 2009). We ran the MCMC algorithm for 100,000 iterations (with a burn-in period of 10,000 iterations) for a sequence of model fits using each of the 50 ensemble members as data with the remaining 49 as the computer experiment output. Effectively, this procedure resulted in a simulation study whereby we assessed the ability of the emulator model to recover the "known" parameter values. The 95% marginal credible intervals for each biological parameter and each model fit are shown in Figure 2. Additionally, we checked the credible interval coverage for insight into which biological parameters were identifiable and well-estimated (Table 3).

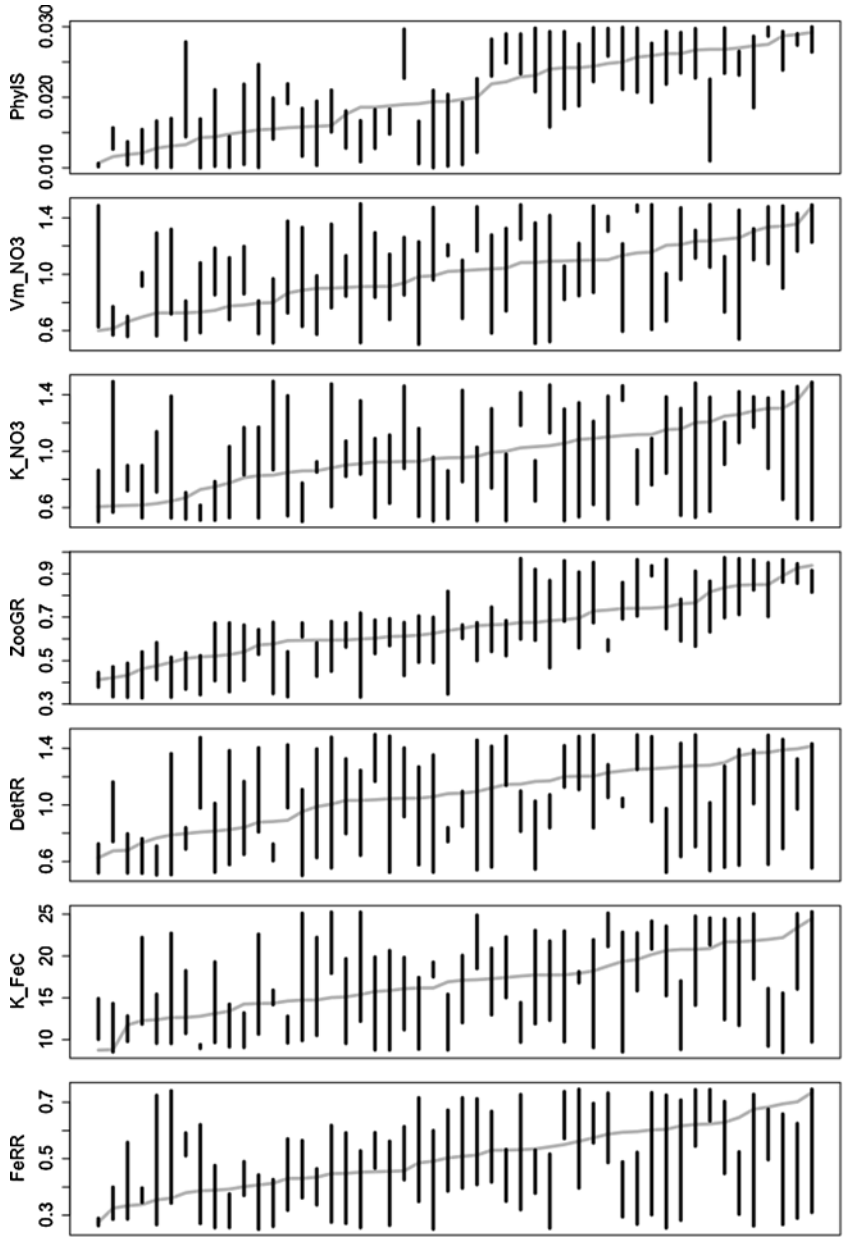

Figure 2. 95% credible intervals (vertical bars) resulting from nonlinear emulator model fits using each of the 50 sets of simulated data from the ROMS-NPZDFe model. For each biological model parameter, the "true" values (gray line) were ordered from smallest to largest for illustration purposes only. From top to bottom, y-axis units are: m^2/W, day^{-1}, mmolN/m^{-3}, day^{-1}, day^{-1}, mmolFe/molC, and day^{-1}.

We also fit the ROMS-NPZDFe nonlinear first-order emulator model to remotely sensed ocean color observations from SeaWiFS that have been processed to represent chlorophyll with units mgChl/m^3. To reconcile the SeaWiFS chlorophyll values with the 'P' variable in the ROMS-NPZDFe computer model, we transformed the latter such that

Table 3. Credible interval coverage for ROMS-NPZDFe physical parameters.

Parameter	95% CI	90% CI	80% CI
PhyIS	96	94	88
Vm NO3	86	84	76
NO3	90	90	78
ZooGR	96	96	94
DetRR	96	92	82
K FeC	88	84	76
FeRR	84	80	74

$y = \text{chlorophyll} \cdot (60/12) \cdot (16/106)$, where the multiplicative ratios correspond to the carbon-to-chlorophyll and nitrogen-to-carbon contents within phytoplankton cells, respectively. Thus, our response variable $\mathbf{y}$ has the same units as the modeled phytoplankton.

Using the same number of MCMC iterations and burn-in period as the simulated data, uniform prior distributions for $\boldsymbol{\theta}$ with the bounds for support described in Table 2, and a log-Gaussian prior on the variance component (i.e., $\log(\sigma) \sim \text{N}(0, 10)$), we obtained posterior distributions for the ROMS-NPZDFe biological parameters (Figure 3). An important computational note is that the statistical emulator model fit takes approximately 20 minutes on a 2×2.93 GHz 6-Core Intel Xeon Workstation with 32 GB of memory. To fit a similar statistical model based on the exact ROMS-NPZDFe computer model, we would need $100{,}000 \times 30 = 3$ million minutes, or approximately 2,083 days!

To illustrate the computer model and statistical model in the space of the response variable, Figure 4a shows the transformed SeaWiFS data ($\mathbf{y}$) and the computer model output ($\mathbf{Y}$), whereas Figure 4b displays the posterior predictive 95% credible interval for $\mathbf{y}$ given the statistical model fit.

Overall, the simulation study indicated that the statistical emulator model allows us to identify some, but not all of the parameters. From Figure 2, we see that the credible intervals for parameters `PhyIS` and `ZooGR` follow the truth fairly well, but with credible interval coverages higher than would be expected (Table 3). The other parameters seem to have more accurate credible interval coverages, but do not indicate a great deal of statistical learning has occurred (Figure 2). Thus, while the posterior distributions for these parameters may be accurate, the utility of the results is limited as compared with that of `PhyIS` and `ZooGR`. In general, this example demonstrates a potential use for these emulators as assessors of model parameter identifiability. In our case, it appears that only two or three of the parameters are identifiable in the ROMS-NPZDFe computer model. It is well known that there is substantial dependence between parameters of these models and there are typically not enough data available to adequately estimate all of them. This is known as the "underdetermination problem" in marine biogeochemical modeling (e.g., Ward et al. 2010).

Using the "real" SeaWiFS data, we find marginal posterior distributions that indicate learning for the `PhyIS`, `K_NO3`, `ZooGR`, and `DetRR` parameters, but the remainder of the parameters show little difference from their diffuse priors. Specifically, the zooplankton

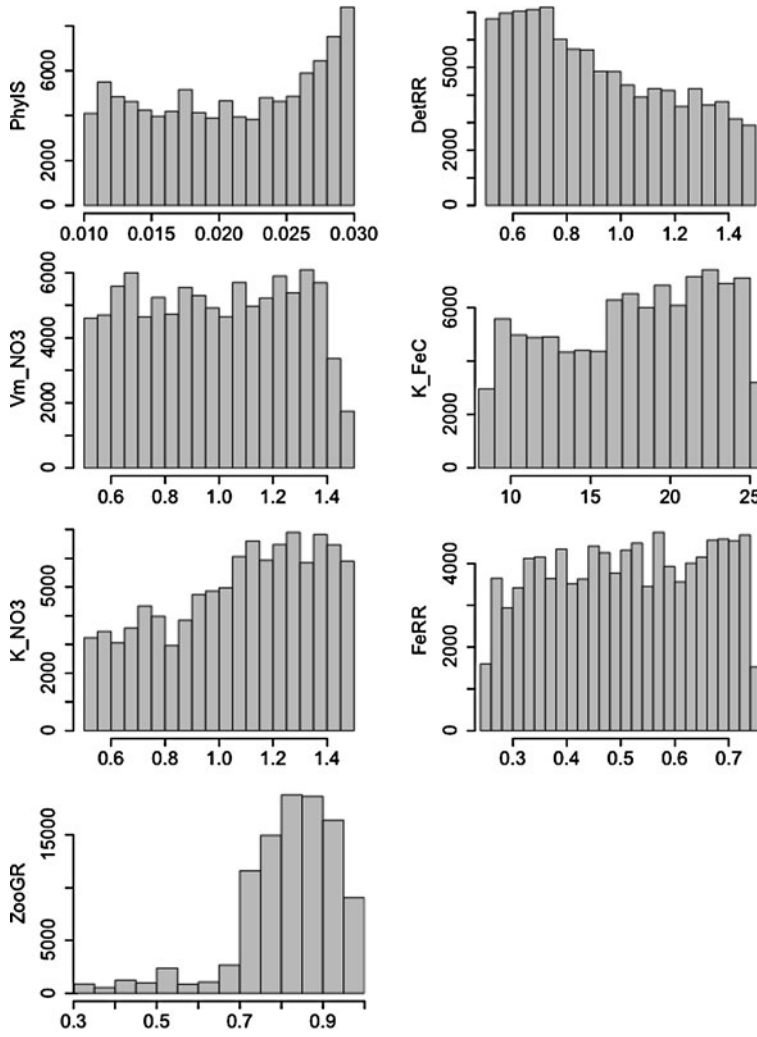

Figure 3. Marginal posterior distributions for the physical parameters, $\boldsymbol{\theta}$, resulting from the nonlinear emulator model fit using the SeaWiFS data and the ROMS-NPZDFe physical-biological model. For each marginal distribution, the x-axis indicates the compact support of the parameter. From top to bottom and left to right, x-axis units are: m^2/W, day^{-1}, mmolN/m^{-3}, day^{-1}, day^{-1}, mmolFe/molC, and day^{-1}.

grazing rate (ZooGR) seems to be the most informed by the SeaWiFS data and indicates that during this particular period the zooplankton grazing rate is nearly as high as it could be and still provide scientifically reasonable phytoplankton values. Similarly, the detritus remineralization rate (DetRR) is low during this period, possibly as a function of the high ZooGR. The posterior predictive inference in Figure 4 indicates that, though the credible intervals in the early part of the year (i.e., day < 100), when phytoplankton is low, are wide, the general statistical emulator model is able to capture the smaller peak later in the year (i.e., $200 <$ day < 300) by inflating the truncated normal variance parameter. If better posterior predictive inference were desired, one might want to employ a bias-correction in the likelihood (e.g., Cangelosi and Hooten 2009) so that smaller values have a more

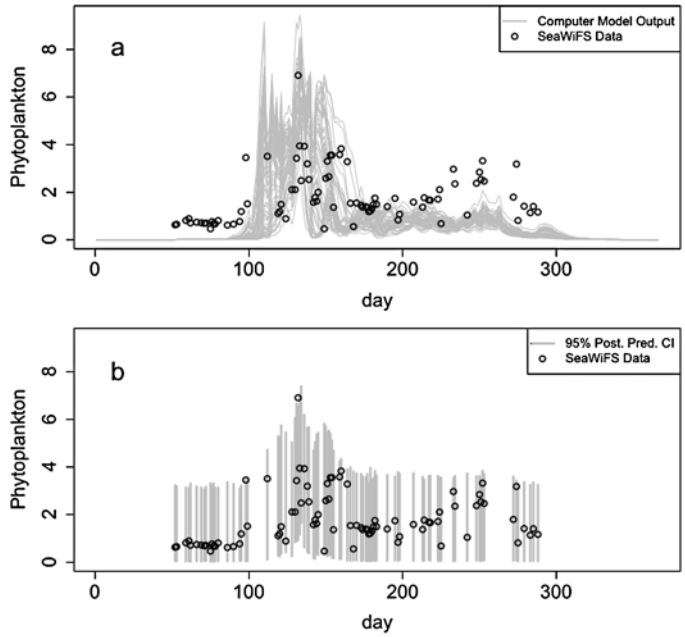

Figure 4. (a) ROMS-NPZDFe physical-biological deterministic model output $\mathbf{Y}$ based on the parameter values ($\boldsymbol{\Theta}$) used in the computer experiment and SeaWiFS data $\mathbf{y}$. (b) Posterior predictive 95% credible intervals for $\mathbf{y}$. Units on the y-axis are mmolN/m^3.

representative posterior predictive variance. In this case, we are primarily interested in parameter estimation and based on our simulation study we have found that the most identifiable ROMS-NPZDFe model parameters are `PhyIS` and `ZooGR`, and our inference should focus primarily on them.

4. DISCUSSION

Overall, we have evaluated a nonlinear first-order emulator model for the purposes of estimating parameters in complicated and computationally expensive computer models. The particular approach we used relies on the ability to predict right singular vectors (resulting from a decomposition of computer model experimental output) based on the computer model input and an experimental set of parameters. A main thrust of this research was to find an intuitive and readily implementable method for inverse inference on computer models that was similar in spirit to the nonlinear second-order emulators previously developed (e.g., Higdon et al. 2008). Thus, we employed a random forest predictive approach to perform the first-order emulation component of the model. Aside from the fact that this method has been proven to be a good nonlinear predictor in most situations (Hastie, Tibshirani, and Friedman 2009), it was especially appealing because of the minimal tuning required in the algorithm. It should be noted that, even though the underlying predictor we use is somewhat automatic, the MCMC algorithm itself is still based on Metropolis–Hastings updates and thus requires some tuning of the proposal distributions. However, this

could also be made more automatic using available adaptive sampling procedures (e.g., Roberts and Rosenthal 2009). In terms of overall computation, our findings echo those of previous emulator studies, in that mimicking a complicated computer model with a simpler nonlinear statistical model can result in huge computational savings.

In terms of fitting the models, we cannot claim that this first-order emulator approach is faster than second-order approaches, however it is definitely simpler and easier to implement (i.e., to design and write computer code for). In fact, after the computer experiment has been performed, it only requires a few a priori calculations (i.e., singular value decomposition and random forest fits) and a rudimentary MCMC algorithm with Metropolis–Hastings updates for model parameters.

We demonstrated the capability of our approach with a very simple nonlinear power law model and also with a much more complex coupled physical-biological model. Our results indicate that the first-order emulator model seems to work well for situations where the variation in the computer model output can be explained by a small set of left singular vectors and the computer model parameters themselves are identifiable. The emulator approach we describe here may not be ideal for situations where (1) the computer model output exhibits small scale behavior that cannot be expressed with a limited number of principal components, or (2) if a massive number of computer model runs are required to sufficiently explore the relevant space of the computer model output. In scenario 1, the first-order emulator model may still perform well, but if the dimensionality of the retained singular vectors is large it will require more computational time to fit. As a partial remedy to scenario 2, more optimal sampling strategies could be employed in the computer experiment to better explore the parameter space (e.g., latin hypercube design or an optimal adaptive sampling design).

Of course, in many complicated nonlinear physical models, inverse inference could be difficult even using the exact computer model itself because of inherent correlations and dependencies in the model structure and parameters. Thus, at the very least, statistical emulator models that focus on parameter estimation could be used to evaluate which model parameters are identifiable before the model is fit using "real" observations. The procedure of using the mechanistic model to simulate data and then evaluating the statistical model performance based on these simulations is merely a standard simulation study and could be employed in any statistical model that allows for direct simulation of observed quantities (including second-order emulators). However, for computationally intensive computer models specifically, the speed of the first-order emulator approach may allow for insight concerning identifiability at a low computational cost and could even serve as a form of exploratory analysis tool that would precede the use of a more powerful emulator or exact model.

The methodology presented here is not limited by the choice of the random forest model for the right singular vector. Any reasonable nonlinear and/or nonparametric statistical model could be used (e.g., radial basis functions, splines, neural networks, etc.). For example, we have implemented a radial basis function-based first-order emulator and it gives comparable results to the random forest model results presented here. Such models may have more natural mechanisms to include uncertainty in the predictions of the right

singular vectors than those presented here. Perhaps more importantly, it is likely that a hybrid approach that includes both a first-order structure and a second-order structure would be more efficient, as is the case, for example, in universal kriging applications in spatial statistics. This would be particularly appealing in multivariate settings in which realistic second-order structures might be difficult to parameterize, but residual dependence structures from some first-order emulator might be parameterized by relatively simple structure. This is a topic for future research.

ACKNOWLEDGEMENTS

The authors thank Ralph Milliff, Jeremiah Brown, Mark Berliner, Andrew Moore, and Adele Cutler for providing data, preliminary analyses, and helpful suggestions pertaining to this project. Funding for this project was provided through NSF OCE-0814934, NSF OCE-0815030, and ONR N00014-10-1-0518. Any use of trade names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

[Published Online November 2011.]

REFERENCES

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C. H., and Tu, J. (2007a), "A Framework for Validation of Computer Models," *Technometrics*, 49, 138–154.

Bayarri, M. J., Berger, J. O., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R. J., Paulo, R., Sacks, J., and Walsh, D. (2007b), "Computer Model Validation with Functional Output," *The Annals of Statistics*, 35, 1874–1906.

Bliznyuk, N., Ruppert, D., Shoemaker, C. A., Regis, R., Wild, S., and Mugunthan, P. (2008), "Bayesian Calibration of Computationally Expensive Models Using Optimization and Radial Basis Function Approximation," *Journal of Computational and Graphical Statistics*, 17, 270–294.

Breiman, L. (2001), "Random Forests," *Machine Learning*, 45, 5–32.

Cangelosi, A. R., and Hooten, M. B. (2009), "Models for Bounded Systems with Continuous Dynamics," *Biometrics*, 65, 850–856.

Conti, S., Gosling, J. P., Oakley, J. E., and O'Hagan, A. (2009), "Gaussian Process Emulation and Dynamic Computer Codes," *Biometrika*, 96, 663–676.

Craig, P. S., Goldstein, M., Rougier, J. C., and Seheult, A. H. (2001), "Bayesian Forecasting for Complex Systems using Computer Simulators," *Journal of the American Statistical Association*, 96, 717–729.

Crawford, W. R., Brickley, P. J., and Thomas, A. C. (2007), "Mesoscale Eddies Determine Phytoplankton Distribution in Northern Gulf of Alaska," *Progress in Oceanography*, 75, 287–303.

Cutler, D. R., Edwards, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., and Lawler, J. J. (2007), "Random Forests for Classification in Ecology," *Ecology*, 88, 2783–2792.

Drignei, D. (2008), "Fast Statistical Surrogates for Dynamical 3D Computer Models of Brain Tumors," *Journal of Computational and Graphical Statistics*, 17, 859–884.

Fiechter, J., and Moore, A. M. (2009), "Interannual Spring Bloom Variability and Ekman Pumping in the Coastal Gulf of Alaska," *Journal Geophysical Research*, 114, C06004.

Fiechter, J., Moore, A. M., Edwards, C. A., Bruland, K. W., Di Lorenzo, E., Lewis, C. V. W., Powell, T. M., Curchitser, E. N., and Hedstrom, K. (2009), "Modeling Iron Limitation of Primary Production in the Coastal Gulf of Alaska," *Deep Sea Research II*, 56, 2503–2519.

Frolov, S., Baptista, A. M., Leen, T. K., Lu, Z., and van der Merwe, R. (2009), "Fast Data Assimilation Using a Nonlinear Kalman Filter and a Model Surrogate: An Application to the Columbia River Estuary," *Dynamics of Atmospheres and Oceans*, 48, 16–45.

Gentle, J. E. (2007), *Matrix Algebra: Theory, Computations, and Applications in Statistics*, New York: Springer.

Haidvogel, D. B., Arango, H., Budgell, W. P., Cornuelle, B. D., Curchitser, E. N., Di Lorenzo, E., Fennel, K., Geyer, W. R., Hermann, A. J., Lanerolle, L., Levin, J., McWilliams, J. C., Miller, A. J., Moore, A. M., Powell, T. M., Shchepetkin, A. F., Sherwood, C. R., Signell, R. P., Warner, J. C., and Wilkin, J. (2008), "Ocean Forecasting in Terrain-Following Coordinates: Formulation and Skill Assessment of the Regional Ocean Modeling System," *Journal of Computational Physics*, 227, 3595–3624.

Hastie, T., Tibshirani, R., and Friedman, J. (2009), *Elements of Statistical Learning* (2nd ed.), New York: Springer.

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004), "Combining Field Data and Computer Simulations for Calibration and Prediction," *SIAM Journal on Scientific Computing*, 26, 448–466.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), "Computer Model Calibration Using High-Dimensional Output," *Journal of the American Statistical Association*, 103, 570–583.

Kennedy, M. C., and O'Hagan, A. (2001), "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society: Series B*, 63, 425–464.

Liu, F., and West, M. (2009), "A Dynamic Modelling Strategy for Bayesian Computer Model Emulation," *Bayesian Analysis*, 4, 393–412.

O'Hagan, A. (2006), "Bayesian Analysis of Computer Code Outputs: A Tutorial," *Reliability Engineering and System Safety*, 91, 1290–1300.

Powell, T. M., Lewis, C. V. W., Curchitser, E. N., Haidvogel, D. B., Hermann, A. J., and Dobbins, E. L. (2006), "Results from a Three-Dimensional, Nested, Biological-Physical Model of the California Current System and Comparisons with Statistics from Satellite Imagery," *Journal of Geophysical Research*, 111 (C0), 7018.

Roberts, G. O., and Rosenthal, J. S. (2009), "Examples of Adaptive MCMC," *Journal of Computational and Graphical Statistics*, 18, 349–367.

Rougier, J. C. (2008), "Efficient Emulators for Multivariate Deterministic Functions," *Journal of Computational and Graphical Statistics*, 17, 827–843.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–423.

Stabeno, P. J., Bond, N. A., Hermann, A. J., Kachel, N. B., Mordy, C. W., and Overland, J. E. (2004), "Meteorology and Oceanography of the Northern Gulf of Alaska," *Continental Shelf Research*, 24, 859–897.

van der Merwe, R., Leen, T. K., Lu, Z., Frolov, S., and Baptista, A. M. (2007), "Fast Neural Network Surrogates for Very High Dimensional Physics-Based Models in Computational Oceanography," *Neural Networks*, 20, 462–478.

Ward, B. A., Friedrichs, M. A. M., Anderson, T. R., and Oschlies, A. (2010), "Parameter Optimisation Techniques and the Problem of Underdetermination in Marine Biogeochemical Models," *Journal of Marine Systems*, 81, 34–43.

Xiao, X., White, E. P., Hooten, M. B., and Durham, S. L. (2011), "On the Use of Log-Transformation Versus Nonlinear Regression for Analyzing Biological Power-Laws," *Ecology*, 92, 1887–1894.