

Empirical Bayesian Analysis for Computer Experiments Involving Finite-Difference Codes

Dorin DRIGNEI and Max D. MORRIS

Computer experiments are increasingly used in scientific investigations as substitutes for physical experiments in cases where the latter are difficult or impossible to perform. A computer experiment consists of several runs of a computer model or “code” for the purpose of better understanding the *input* \rightarrow *output* relationship. One practical difficulty in the use of these models is that a single run may require a prohibitive amount of computational resources in some situations. A recent approach uses statistical approximations as less expensive surrogates for such computer codes; these provide both point predictors and uncertainty characterization of the outputs. A widely used class of computer codes is the finite-difference solvers of differential equations, which produce multivariate output (e.g., time series). The finite-difference relationship underpins the statistical model proposed here, and we show that this strategy has clear computational and accuracy advantages over a direct multivariate extension of the popular scalar modeling methodology.

KEY WORDS: Computer code; Computer model; Differential equation; Local truncation errors; Maximum likelihood estimation; Non-stationary model.

1. INTRODUCTION

Many complex physical phenomena are investigated through the development and use of deterministic computer simulation models, or “codes.” Computer codes are often complex, reflecting detailed understanding of the physical processes of interest. However, the complexity of the computer codes sometimes limits their usefulness, due to the accompanying demand on computational resources. In some cases it is necessary to develop faster running surrogates for slow-running computer codes. These faster (or “cheaper”) surrogates are approximations of the original codes, however, so the error introduced from using a surrogate should also be characterized.

Statistical models are natural choices as surrogates because they support both output prediction and uncertainty (error) characterization. A statistical approach to this problem has been outlined by Sacks, Welch, Mitchell, and Wynn (1989) and Currin, Mitchell, Morris, and Ylvisaker (1991). These authors discussed modeling of computer codes for which inputs can be of high dimension, but focused on a single scalar-valued output. Input vector values are sometimes called “sites” or “locations” in the input space, following an analogy with geostatistics in which related “kriging” methods are used. A computer experiment consists of a number of code runs at various input configurations. It is desirable to choose these inputs carefully (i.e., design the experiment well), to make the resulting statistical surrogates as efficient as possible. Once the design inputs have been chosen, the computer code is executed and the output is modeled statistically. Sacks et al. (1989) and Currin et al. (1991) estimated the process parameters by maximum likelihood, with the former group taking a kriging approach to prediction. In many cases, the statistical model is described in a Bayesian context, where a stationary Gaussian process is prescribed as a prior for the computer code output as a function of inputs. Currin et al. used the posterior distribution of the output at a

new input site (i.e., one not included in the design), as a surrogate for the slow-running computer code. The hope is that these predictions are of good statistical quality and are also substantially less computationally demanding than new runs of the computer code.

Because the data-generating mechanism is known (albeit complicated), a natural place to look for additional information to strengthen the methodology is the computer code itself. For example, Morris, Mitchell, and Ylvisaker (1993) discussed “enhanced” codes that produce derivatives of outputs with respect to the components of the input vector and showed that using this additional code information increases the precision of the statistical predictions. More recently, Kennedy and O’Hagan (2001) noted that their Bayesian calibration method treats the computer code as a black box, but acknowledged that exploiting additional information about the code could improve the approach. In his discussion of that work, Wynn (2001) argued that “it is useful to know what is going on inside the black box,” alluding to the need to incorporate information about the computer code, especially for differential equation solvers. The solutions of the differential equations, especially in the nonlinear case, exhibit complicated dynamics for which some of the assumptions commonly used in the foregoing statistical methodology may be inappropriate.

The computer codes considered here are finite-difference numerical solvers of differential equations. In such applications interest lies in analyzing the resulting multivariate computer output. We use information about the computer code that is implicit in the approximate *local truncation errors* computed for each run to automatically construct a second-order nonstationary statistical model of the output that realistically mimics the dynamics of the computer code. This is particularly appealing for codes based on nonlinear differential equations, because the simulator proposed here inherits appropriate nonlinear behavior through the derived nonstationary structure. To illustrate the methodology, we present examples based on a competing species differential system from mathematical biology. We show that a nonstationary output model derived from a stationary model of truncation errors is a good representation of the computer code, whereas specifying a statistical model based

Dorin Drignei is Assistant Professor, Department of Mathematics and Statistics, Oakland University, Rochester, MI 48309 (E-mail: drignei@oakland.edu). Max D. Morris is Professor, Department of Statistics and Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, IA 50011 (E-mail: mmorris@iastate.edu). This research is based on Drignei’s doctoral thesis in the Department of Statistics, Iowa State University. This article was revised while Dorin Drignei was a visiting scientist at the National Center for Atmospheric Research and supported by the National Science Foundation (grant DMS-03-55474).

on a stationary prior process fitted directly to the output requires a very computationally expensive likelihood optimization and when the estimation is feasible, results in inferior predictions. The findings of the analysis are strengthened by a proof demonstrating that the method proposed herein has good asymptotic properties.

Section 2 introduces the general computer model, the output data, and a straightforward extension of the approach described by Currin et al. (1991) that accommodates multivariate output indexed by time and space. Section 3 describes the main contribution of this article, a modification of the stochastic process modeling approach that takes advantage of the information in approximate truncation errors. Section 4 briefly describes how auxiliary information in the form of less accurate approximate numerical solutions can be incorporated into either type of model. Section 5 presents an example using a form of the species competition model based on partial differential equations.

2. COMPUTER MODELS AND DIRECT SIMULATORS

Systems of differential equations are mathematical models of physical reality that involve derivatives of functions of interest with respect to time and/or other variables, usually space. The general formulation of the models that we consider in this article is

$$\frac{\partial \mathcal{Y}}{\partial t} = \mathbf{f}\left(\mathcal{Y}, \frac{\partial \mathcal{Y}}{\partial s}, \dots\right), \quad (1)$$

where $t \in [0, L_T]$, $s \in [0, L_S]$, and the dots mean that higher-order derivatives with respect to space could also appear. Moreover, (1) could include higher-order derivatives with respect to time and/or space without necessarily including the corresponding lower-order derivatives. Let E denote the length of \mathcal{Y} ; in general, it coincides with the number of scalar equations of (1). For (1) to have unique solution \mathcal{Y} , initial conditions, $\mathcal{Y}(0, s)$, and boundary conditions, $\mathcal{Y}(t, 0)$, $\mathcal{Y}(t, L_S)$, also must be specified. If (1) includes only the time component, then it is known as an ordinary differential equation (ODE) system; otherwise, it is called a partial differential equation (PDE) system. For the sake of generality, the notation that we develop reflects the PDE model, but the ODE development results simply by removing the spatial component. When more than one spatial dimension exists, the space interval $[0, L_S]$ becomes a (hyper)rectangle and (1) can include partial derivatives with respect to each spatial dimension.

Nonlinear systems of differential equations rarely can be solved analytically, even though solutions exist. In these situations we rely on numerical approximations of the analytical solution. Finite-difference methods approximate the required derivatives by scaled differences of the corresponding function evaluated at neighboring points on a grid of values. We define a *fine* grid of $M_F + 1$ equally spaced points on the time interval $[0, L_T]$ and $S_F + 2$ equally spaced points on the space interval $[0, L_S]$, with adjacent points separated by time increment $\Delta_{t_F} = L_T/M_F$ and space increment $\Delta_{s_F} = L_S/(S_F + 1)$, denoted as $G_F = \{\tau_0, \dots, \tau_{M_F}\} \times \{\lambda_0, \dots, \lambda_{S_F+1}\}$ where $\tau_k = k\Delta_{t_F}$ and $\lambda_i = i\Delta_{s_F}$. The general problem that we address here is the construction of statistical

simulators for fine-grid numerical solutions of a computer code, iteratively expressed as

$$\begin{aligned} \mathbf{Y}(\tau_{k+1}, \lambda_i) &= \mathbf{Y}(\tau_k, \lambda_i) \\ &+ \Delta_{t_F} \mathbf{f}\left(\mathbf{Y}(\tau_k, \lambda_i), \frac{\mathbf{Y}(\tau_k, \lambda_{i+1}) - \mathbf{Y}(\tau_k, \lambda_i)}{\Delta_{s_F}}, \dots\right). \end{aligned} \quad (2)$$

Here the vector $\mathbf{Y}(\tau_k, \lambda_i)$ is E -dimensional and is computed on G_F as follows: $\mathbf{Y}(\tau_0, \lambda_i)$ is given in the initial condition, and for a fixed k , $\mathbf{Y}(\tau_{k+1}, \lambda_i)$ is computed iteratively with respect to i using the given boundary conditions $\mathbf{Y}(\tau_k, \lambda_0)$ and $\mathbf{Y}(\tau_k, \lambda_{S_F+1})$. Relatively finer grids produce more accurate solutions but are more computationally intensive.

2.1 Output Data

“Data” for this exercise consist of the output from a number of runs of the finite-difference solver. A statistical simulator fitted to these data can be used to predict output that would result from code runs at other input vectors. The experimental design for the computer experiment consists of the specification of D input vectors associated with the differential equations. For simplicity of explanation, assume that the input vectors are the initial conditions $\mathbf{Y}(0, \cdot)$, denoted by \mathbf{d}_m , $m = 1, 2, 3, \dots, D$. (The examples presented in this article reflect this choice, but other problems may require consideration of different types of input vectors.) For each of these, a finite-difference scheme is used to obtain numerical solutions on G_F . To ensure numerical convergence of the solver, the grid increments Δ_{t_F} and Δ_{s_F} must be small. In fact, the scientifically interesting characteristics of the output ordinarily can be fully expressed by recording only a small subset of the fine grid output data. Hence we also define a relatively *coarse* regular grid of points consisting of $M_C + 1$ time points and $L_C + 2$ space points, $G_C = \{t_0, \dots, t_{M_C}\} \times \{l_0, \dots, l_{S_C+1}\}$, with time step $\Delta_{t_C} = L_T/M_C$ and space step $\Delta_{s_C} = L_S/(S_C + 1)$, such that $G_C \subset G_F$. Only the fine grid output values associated with points on the coarse grid are incorporated into the statistical simulator. Moreover, because the initial and boundary conditions values are known, these values are not included in the statistical analyses, and thus only the fine grid output values at indices $\{t_1, \dots, t_{M_C}\} \times \{l_1, \dots, l_{S_C}\}$ need to be considered. Hence the number of (individual scalar) output values analyzed is M_C (time points) $\times S_C$ (space points) $\times E$ (output values at each time–space point) $\times D$ (sets of initial conditions, or runs of the code).

2.2 Direct Modeling of Output

Current statistical methodology for computer experiments, such as that described by Sacks et al. (1989), treats the output as a scalar-valued stochastic process, indexed by inputs. Kennedy and O’Hagan (2001, p. 435) have suggested that if each run of the computer code produces a multivariate output, then one can continue to use a scalar-valued process with an index set expanded to include both input and output indexes. This amounts to specifying a $M_C \cdot S_C \cdot E \cdot D$ element mean vector and a $(M_C \cdot S_C \cdot E \cdot D) \times (M_C \cdot S_C \cdot E \cdot D)$ covariance matrix for a complete vector of output from the computer experiment.

Using a product correlation structure simplifies some of the computational issues associated with such large covariance matrices. Let $R_M(t, t')$, t and $t' \in (0, L_T]$, be a correlation function associated with time; $R_S(s, s')$, s and $s' \in (0, L_S)$, be a correlation function associated with space; $R_E(a, b)$, a and $b \in \{1, \dots, E\}$, be a correlation function associated with variables of the output set; and $R_D(\mathbf{d}, \mathbf{d}')$, \mathbf{d} and \mathbf{d}' from the space of allowed input vectors, be a correlation function associated with initial conditions. Then define the correlation between $Y_a(t_u, l_i)$ following from \mathbf{d}_m and $Y_b(t_v, l_j)$ following from \mathbf{d}_n as $R_M(t_u, t_v) \cdot R_S(l_i, l_j) \cdot R_E(a, b) \cdot R_D(\mathbf{d}_m, \mathbf{d}_n)$. Because the scalar output values collected from an experiment correspond to a complete regular grid of times, spatial locations, output variables, and input vectors, the covariance matrix for the full data vector can be written as a Kronecker product of much smaller matrices generated from R_M , R_S , R_E , and R_D . Specifically:

- Let \mathbf{C}_M be the $M_C \times M_C$ time correlation matrix with (u, v) element $R_M(t_u, t_v)$
- Let \mathbf{C}_S be the $S_C \times S_C$ space correlation matrix with (i, j) element $R_S(l_i, l_j)$
- Let \mathbf{C}_E be the $E \times E$ output element correlation matrix with (a, b) element $R_E(a, b)$
- Let \mathbf{C}_D be the $D \times D$ input vector correlation matrix with (m, n) element $R_D(\mathbf{d}_m, \mathbf{d}_n)$.

The correlation functions R_M , R_S , R_E , and R_D , and through them the matrices \mathbf{C}_M , \mathbf{C}_S , \mathbf{C}_E , and \mathbf{C}_D , depend on statistical parameters denoted by ϕ , ψ , ρ , and θ , where θ is typically of the same dimension as \mathbf{d} . Let \mathbf{Y} denote the complete data vector of output values, ordered as time series within space, within output variables, and within initial condition values. Then $\mathbf{Y} \sim \mathcal{N}(\mu\mathbf{1}, \sigma^2\mathbf{\Sigma})$, $\mathbf{\Sigma} = \mathbf{C}_D \otimes \mathbf{C}_E \otimes \mathbf{C}_S \otimes \mathbf{C}_M$, where “ \otimes ” represents the Kronecker product.

The maximum likelihood estimates (MLEs) of μ and σ^2 , given values of the other parameters, are $\hat{\mu} = (\mathbf{1}'\mathbf{\Sigma}^{-1}\mathbf{Y})/(\mathbf{1}'\mathbf{\Sigma}^{-1}\mathbf{1})$ and $\hat{\sigma}^2 = (\mathbf{Y} - \mathbf{1}\hat{\mu})'\mathbf{\Sigma}^{-1}(\mathbf{Y} - \mathbf{1}\hat{\mu})/(M_C \cdot S_C \cdot E \cdot D)$. Estimates of ϕ , ψ , ρ , and θ are obtained numerically by iterative maximization of the likelihood function or, equivalently, by minimizing the objective function

$$l(\phi, \psi, \rho, \theta | \mathbf{Y}) = \log(\hat{\sigma}^2) + \frac{\log(\det(\mathbf{C}_D))}{D} + \frac{\log(\det(\mathbf{C}_E))}{E} + \frac{\log(\det(\mathbf{C}_S))}{S_C} + \frac{\log(\det(\mathbf{C}_M))}{M_C}.$$

Note, however, that this likelihood optimization is potentially very computationally expensive. The estimates $\hat{\mu}$ and $\hat{\sigma}^2$, and thus $l(\phi, \psi, \rho, \theta | \mathbf{Y})$, require more than $O(D^2 M_C^2 S_C^2 E^2)$ algebraic operations because $\mathbf{\Sigma}^{-1} = \mathbf{C}_D^{-1} \otimes \mathbf{C}_E^{-1} \otimes \mathbf{C}_S^{-1} \otimes \mathbf{C}_M^{-1}$, and the additional computational time for determinants and inverses also must be considered. In addition to numerical complexity, the likelihood function is often rather flat for this model; penalized likelihood methods such as the one described by Li and Sudjianto (2005) can sometimes improve estimation accuracy in these cases.

These estimates are then treated as known parameter values, leading to simple empirical Bayesian predictions and predictive standard errors for outputs at new input vectors. Let \mathbf{d}_0 be a new vector not included in the D -point experimental design. The

posterior mean vector at the new site \mathbf{d}_0 , the usual point predictor suggested by minimization of expected squared error loss, is $\hat{\mathbf{Y}}_{\mathbf{d}_0} = \mu\mathbf{1} + [(\mathbf{C}_{\mathbf{d}_0, D} \mathbf{C}_D^{-1}) \otimes \mathbf{I}_{M_C \cdot S_C \cdot E}](\mathbf{Y} - \mu\mathbf{1})$, where $\mathbf{C}_{\mathbf{d}_0, D}$ is the vector of correlations corresponding to the new input \mathbf{d}_0 and outputs at the D design inputs. This expression shows that the posterior mean involves only the correlation between the design vectors, not the correlation matrices \mathbf{C}_E , \mathbf{C}_S , and \mathbf{C}_M . The posterior variance at \mathbf{d}_0 is $\sigma^2(1 - \mathbf{C}_{\mathbf{d}_0, D} \mathbf{C}_D^{-1} \mathbf{C}'_{\mathbf{d}_0, D})(\mathbf{C}_E \otimes \mathbf{C}_S \otimes \mathbf{C}_M)$. The diagonal of the posterior covariance at \mathbf{d}_0 gives the marginal posterior variances for each variable, space, and time; that is, the marginal posterior variances are all equal to $\sigma^2(1 - \mathbf{C}_{\mathbf{d}_0, D} \mathbf{C}_D^{-1} \mathbf{C}'_{\mathbf{d}_0, D})$, which again involves only the correlation between \mathbf{d}_0 and the design set. The prediction standard errors are defined as the square root of the marginal posterior variances.

As noted earlier, a major operational difficulty with this approach is the computational burden involved in maximizing the likelihood function with respect to the covariance parameters. The size of the dense covariance matrix, even in Kronecker-factored form, can lead to very slow (or even practically impossible) iterations for applications of even modest size. In the next section we examine a related modeling approach based on numerical truncation errors, which can substantially reduce this computational burden while simultaneously improving the statistical quality of the predictions.

3. MODELING BASED ON LOCAL TRUNCATION ERRORS

An accuracy measure of a grid finite-difference solver is the local truncation error, defined as the error that would be generated in one time step of an iterative numerical solver, given the analytical solution value at the beginning of the step. Hence for a solver based on the coarse grid that we have defined, the local truncation error is

$$\begin{aligned} \mathbf{T}^{\mathcal{Y}}(t_{k+1}, l_i) &= \mathcal{Y}(t_{k+1}, l_i) \\ &\quad - \left[\mathcal{Y}(t_k, l_i) \right. \\ &\quad \left. + \Delta_{tC} \mathbf{f} \left(\mathcal{Y}(t_k, l_i), \frac{\mathcal{Y}(t_k, l_{i+1}) - \mathcal{Y}(t_k, l_i)}{\Delta_{sC}}, \dots \right) \right] \end{aligned} \quad (3)$$

for $k = 0, \dots, M_C - 1$, $i = 0, \dots, S_C$, where \mathcal{Y} is the analytical solution on the coarse grid. The local truncation error can be viewed as a residual quantity and provides a measure of computational accuracy of the finite-difference scheme.

The precise values of truncation errors are not available in general, because they require knowledge of \mathcal{Y} . However, we can obtain *approximate* local truncation errors by replacing the analytical solutions with their fine-grid approximations in (3),

$$\begin{aligned} \mathbf{T}^{\mathbf{Y}}(t_{k+1}, l_i) &= \mathbf{Y}(t_{k+1}, l_i) \\ &\quad - \left[\mathbf{Y}(t_k, l_i) \right. \\ &\quad \left. + \Delta_{tC} \mathbf{f} \left(\mathbf{Y}(t_k, l_i), \frac{\mathbf{Y}(t_k, l_{i+1}) - \mathbf{Y}(t_k, l_i)}{\Delta_{sC}}, \dots \right) \right] \end{aligned} \quad (4)$$

for $k = 0, \dots, M_C - 1, i = 1, \dots, S_C$. Note that this is *not* simply a difference between fine and coarse numerical solutions. Let \mathbf{T}^Y denote the complete vector of the $M_C \cdot S_C \cdot E \cdot D$ approximate local truncation errors. We specify $\mathbf{T}^Y \sim N(\mu \mathbf{1}, \sigma^2 \Sigma)$ and $\Sigma = \mathbf{C}_D \otimes (\mathbf{C}_E \otimes \mathbf{C}_S \otimes \mathbf{C}_M)$, where “ \otimes ” represents the Kronecker product. In the examples given in Section 5, \mathbf{C}_D is the $D \times D$ input vector correlation matrix with (m, n) element

$$\mathbf{C}_D(m, n) = \exp\left(-\sum_{i=1}^E \theta_i (\mathbf{d}_m(i) - \mathbf{d}_n(i))^2\right),$$

a product of individual Gaussian correlation functions. The space–time stationarity assumption is more suitable for the approximate local truncation errors than for the numerical solutions because theoretical arguments show that local truncation errors have roughly constant magnitude across space time [e.g., $O(\Delta_{i_C}^p) + O(\Delta_{s_C}^q)$ for some specific positive integers p and q], and space–time averages of approximately 0. Moreover, the correlations between truncation errors are much weaker than the correlations between outputs. Indeed, Taylor series arguments (e.g., Ames 1965, pp. 323–324) show that the truncation errors are a combination of higher-order derivatives of the output functions, and it is known (e.g., Ramsay and Silverman 1997, sec. 13.4.1) that derivatives of functions are generally rougher than the functions themselves. To help improve the computational efficiency of the likelihood optimization, independence is assumed herein for the approximate local truncation errors, that is $\mathbf{C}_M = \mathbf{I}_{M_C}$, $\mathbf{C}_S = \mathbf{I}_{S_C}$, and $\mathbf{C}_E = \mathbf{I}_E$.

Two important points should be made immediately. First, although we have chosen to model truncation errors independently in time, in space, and across outputs, this does *not* imply the same kind of independence structure for \mathbf{Y} ; in fact, the output is correlated through all dimensions. This correlation pattern is not easy to characterize analytically, but is induced by the nonlinear relationship between outputs and truncation errors. Second, as noted in the previous section, this option is unavailable in the direct modeling approach, because output functions are generally smooth in time and space—a dense covariance matrix is the only realistic choice in this case. Hence modeling through truncation errors can lead to both faster likelihood optimization (process fitting) and nonstationary behavior in the output predictions that is “naturally” inherited from the dynamic structure of the model. We also prove in the Appendix that if space–time independence is assumed for the approximate local truncation errors, then the statistical predictor of output converges in mean square to the analytical solution of the differential system as the coarse space and time steps approach 0.

As described in Section 2 for the direct modeling approach, we again use the maximum likelihood method to obtain estimates for the parameters $(\mu; \theta; \sigma^2)$, where the overall correlation matrix is now $\Sigma = \mathbf{C}_D \otimes \mathbf{I}_{M_C \cdot S_C \cdot E}$. The MLEs of μ and σ^2 , conditional on the value of θ , are $\hat{\mu} = (\mathbf{1}' \Sigma^{-1} \mathbf{Y}) / (\mathbf{1}' \Sigma^{-1} \mathbf{1})$ and $\hat{\sigma}^2 = (\mathbf{Y} - \mathbf{1} \hat{\mu})' \Sigma^{-1} (\mathbf{Y} - \mathbf{1} \hat{\mu}) / (M_C \cdot S_C \cdot E \cdot D)$. Estimates of θ are obtained numerically by iterative maximization of the likelihood function or, equivalently, by minimizing the objective function,

$$l(\theta | \mathbf{Y}) = \log(\hat{\sigma}^2) + \frac{\log(\det(\mathbf{C}_D))}{D}.$$

It is important to note that both $\hat{\mu}$ and $\hat{\sigma}^2$, and therefore $l(\theta | \mathbf{Y})$, require $O(D^2 \cdot M_C \cdot S_C \cdot E)$ algebraic operations because $\Sigma^{-1} = \mathbf{C}_D^{-1} \otimes \mathbf{I}_{M_C \cdot S_C \cdot E}$ (counting only the elementwise matrix multiplications; the number of operations associated with the inverse and determinant of \mathbf{C}_D is common to all likelihoods presented in this article, so we do not count them in comparisons). Therefore, evaluating this likelihood is less expensive than evaluating the corresponding likelihood based on a dense correlation matrix (required by the direct approach) by a factor of at least $M_C \cdot S_C \cdot E$.

Because the relationship between local truncation errors and outputs can be nonlinear, we use a simulation approach to predict outputs at a new vector \mathbf{d}_0 of initial conditions. The posterior distribution of \mathbf{T}_0^Y is multivariate normal of mean vector $\mu \mathbf{1} + [(\mathbf{C}_{\mathbf{d}_0, D} \mathbf{C}_D^{-1}) \otimes \mathbf{I}_{M_C \cdot S_C \cdot E}](\mathbf{T}^Y - \mu \mathbf{1})$ and covariance matrix $\sigma^2(1 - \mathbf{C}_{\mathbf{d}_0, D} \mathbf{C}_D^{-1} \mathbf{C}'_{\mathbf{d}_0, D}) \mathbf{I}_{M_C \cdot S_C \cdot E}$, and R simulated values $\tilde{\mathbf{T}}_0^Y$ are obtained. Each of these simulated realizations is used to iteratively obtain a simulated output vector through a rearrangement of (4),

$$\begin{aligned} \tilde{\mathbf{Y}}(t_{k+1}, l_i) &= \tilde{\mathbf{Y}}(t_k, l_i) + \Delta_{t_C} \mathbf{f}\left(\tilde{\mathbf{Y}}(t_k, l_i), \frac{\tilde{\mathbf{Y}}(t_k, l_{i+1}) - \tilde{\mathbf{Y}}(t_k, l_i)}{\Delta_{s_C}}, \dots\right) \\ &\quad + \tilde{\mathbf{T}}_0^Y(t_{k+1}, l_i), \end{aligned} \tag{5}$$

for $k = 0, \dots, M_C - 1, i = 1, \dots, S_C$. The iteration starts with $\tilde{\mathbf{Y}}(t_0, l_i)$ which is specified by the initial condition. These iterative calculations are relatively fast, because the coarse grid contains far fewer space–time points than would be required for an accurate finite-difference solver. The median and percentile-based prediction intervals are used to summarize the prediction. In the rest of the article we call the approach presented in this section the TER method.

Finally, recall from Section 2 that in the direct modeling approach, using a product correlation form, the prediction and posterior variance for an unobserved output depend only on the correlations associated with the input vector. Because the other correlation parameters are not involved in output prediction, one might consider assuming $\mathbf{C}_E = \mathbf{I}_E$, $\mathbf{C}_M = \mathbf{I}_{M_C}$, and $\mathbf{C}_S = \mathbf{I}_{S_C}$, for direct modeling, as in the TER approach. This simplification facilitates the likelihood computation, but such severe covariance misspecification leads to less accurate estimates of the remaining parameters. In turn, the misspecified model and (resulting) biased parameter estimates lead to larger prediction errors. (These effects are easy to demonstrate through simulation, but we do not develop the argument further here.) Hence, although truncation errors can be reasonably modeled using a simplified correlation structure, this is not a realistic approach for direct modeling of output.

4. USE OF AUXILIARY INFORMATION

In some situations, numerical solutions corresponding to a grid of intermediate size are available at substantially less computational cost than would be required for fine-grid solutions, although the former are also less accurate than the latter. These can be incorporated as auxiliary information in the statistical model to improve predictions. We consider the use of auxiliary information in the form of a numerical solution on an

intermediate grid G_I of $M_I + 1$ time intervals and $S_I + 2$ space intervals. Here M_I is a positive integer such that $M_F > M_I > M_C$ and S_I is a positive integer such that $S_F > S_I > S_C$, $G_C \subset G_I \subset G_F$, and $\Delta_{t_I} = L_T/M_I$, $\Delta_{s_I} = L_S/(S_I + 1)$. Denote by \mathbf{X} the vector of output computed using the intermediate grid G_I , but recorded only for points in G_C , so that \mathbf{X} is of the same dimension and format as \mathbf{Y} . Denote by \mathbf{T}^X the approximate local truncation errors (again computed on the coarse grid, using the intermediate numerical solution \mathbf{X}). The modeling strategy mirrors the approach presented in the previous sections, except that the prior assumption for the model based on truncation errors is $\mathbf{T}^Y \sim N(\mu_0 \mathbf{1} + \mu_1 \mathbf{T}^X, \sigma^2 \mathbf{\Sigma})$, with $\mathbf{\Sigma}$ defined as in Section 3, and $\mathbf{Y} \sim N(\mu_0 \mathbf{1} + \mu_1 \mathbf{X}, \sigma^2 \mathbf{\Sigma})$ for the direct model, with $\mathbf{\Sigma}$ defined as in Section 2. Note that for the direct model, we still need to assume a dense space–time correlation for residuals, because these are the differences between two highly smooth datasets (\mathbf{Y} and a linear function of \mathbf{X}), whereas the residuals in the TER model can still be considered space–time independent (a characteristic inherited from the truncation error data). Arguments similar to those given in the previous sections show that the likelihood computation of the direct model is more expensive than the likelihood computation of the TER model by at least a factor of $M_C \cdot S_C \cdot E$.

A Bayesian variant of the direct modeling approach presented in Section 2 that takes advantage of intermediate-grid solutions as covariates has been suggested by Kennedy and O’Hagan (2000). In their method, the hyperparameters of the Bayesian model are estimated by maximum likelihood, and thus similar computational difficulties are encountered. The example presented in the next section shows that it is practically impossible to estimate the parameters of a direct model for large output datasets, but an indirect modeling approach of the output based on truncation errors can offer a working alternative.

5. EXAMPLE: A SPACE–TIME COMPETITION MODEL

5.1 Two-Species Model

First, we apply the method based on truncation errors to a system of two PDEs written as

$$\frac{\partial \mathcal{Y}_1}{\partial t} = D_1 \left(\frac{\partial^2}{\partial s_1^2} + \frac{\partial^2}{\partial s_2^2} \right) \mathcal{Y}_1 + \frac{r_1}{K_1} \mathcal{Y}_1 (K_1 - \mathcal{Y}_1) - \frac{r_1 \alpha_{12}}{K_1} \mathcal{Y}_1 \mathcal{Y}_2$$

and

$$\frac{\partial \mathcal{Y}_2}{\partial t} = D_2 \left(\frac{\partial^2}{\partial s_1^2} + \frac{\partial^2}{\partial s_2^2} \right) \mathcal{Y}_2 + \frac{r_2}{K_2} \mathcal{Y}_2 (K_2 - \mathcal{Y}_2) - \frac{r_2 \alpha_{21}}{K_2} \mathcal{Y}_2 \mathcal{Y}_1.$$

The solutions $\mathcal{Y}_1(t, s_1, s_2)$ and $\mathcal{Y}_2(t, s_1, s_2)$ represent the population size for each species at time t and spatial location (s_1, s_2) . The diffusion term appearing in each equation allows each species to spread throughout space. The second term is a species-specific growth term, and the third term allows for interaction between species. This system has been used successfully to model, for example, the displacement of the red squirrel in Britain by the North American gray squirrel at the turn of the last century (Murray 2003, sec. 1.3). In this demonstration, we use $D_1 = D_2 = .01$, $r_1 = r_2 = 8$, $K_1 = K_2 = 2$,

and $\alpha_{12} = \alpha_{21} = 1.75$. The spatial domain is $[0, L_{S_1}] \times [0, L_{S_2}]$ with $L_{S_1} = L_{S_2} = 20$. The boundary conditions specify that the population size on the boundary of the domain is 0 at any time, for each species. This corresponds to a situation where, for example, the spatial domain is an island and the survival conditions on the boundary or beyond are not favorable for the species under study. The spatial spread at time 0 (i.e., the characterization of the initial conditions) is given by $\mathcal{Y}_1(0, s_1, s_2) = n_1 I(s_1, s_2)$ and $\mathcal{Y}_2(0, s_1, s_2) = n_2 I(s_1, s_2)$, where $I(s_1, s_2) = \exp[-.1 * (s_1 - L_{S_1}/2)^2 - .1 * (s_2 - L_{S_2}/2)^2] / \max_{s_1, s_2} \exp[-.1 * (s_1 - L_{S_1}/2)^2 - .1 * (s_2 - L_{S_2}/2)^2]$ (a rescaled, truncated bivariate normal density function) and (n_1, n_2) is the input vector. Gaussian initial conditions (or Gaussian mixtures) are considered a good approximation for the initial conditions in situations where the competition starts at an unknown time but is noticed only after a time lag, and the first observation time is considered the initial time for modeling purposes (Murray 2003, p. 598). The input space is $[.005, .1] \times [.005, .1]$, and we sampled $D = 20$ input vectors (n_1, n_2) according to a maximin Latin hypercube design (see Morris et al. 1993 for details). The temporal interval considered is $[0, 1]$ because, by $t = 1$, the competition is clearly won in most of the cases. With our selected model parameters the species that has a larger population size at time 0 will eventually win the competition, but the speed of the dynamics depends on the values of the initial population sizes. A typical situation is illustrated in Figure 1, corresponding to $n_1 = .085$ and $n_2 = .015$.

The foregoing PDE system is solved numerically on a fine spatiotemporal grid with uniform increments Δ_{t_F} and Δ_{s_F} . Due to system symmetry, we chose the spatial increments to be equal for each dimension. Denote the fine-grid numerical solution at $t = k \Delta_{t_F}$, $s_1 = i \Delta_{s_F}$, and $s_2 = j \Delta_{s_F}$ as $\mathbf{Y}(k, i, j)$. The associated finite-difference scheme is

$$\begin{aligned} Y_1(k+1, i, j) &= Y_1(k, i, j) + \Delta_{t_F} \left[D_1 \right. \\ &\quad \times \left(\frac{Y_1(k, i+1, j) - 2Y_1(k, i, j) + Y_1(k, i-1, j)}{\Delta_{s_F}^2} \right. \\ &\quad \left. \left. + \frac{Y_1(k, i, j+1) - 2Y_1(k, i, j) + Y_1(k, i, j-1)}{\Delta_{s_F}^2} \right) \right. \\ &\quad \left. + \frac{r_1}{K_1} Y_1(k, i, j) (K_1 - Y_1(k, i, j)) \right. \\ &\quad \left. - \frac{r_1 \alpha_{12}}{K_1} Y_1(k, i, j) Y_2(k, i, j) \right] \end{aligned}$$

and

$$\begin{aligned} Y_2(k+1, i, j) &= Y_2(k, i, j) + \Delta_{t_F} \left[D_2 \right. \\ &\quad \times \left(\frac{Y_2(k, i+1, j) - 2Y_2(k, i, j) + Y_2(k, i-1, j)}{\Delta_{s_F}^2} \right. \\ &\quad \left. \left. + \frac{Y_2(k, i, j+1) - 2Y_2(k, i, j) + Y_2(k, i, j-1)}{\Delta_{s_F}^2} \right) \right. \\ &\quad \left. + \frac{r_2}{K_2} Y_2(k, i, j) (K_2 - Y_2(k, i, j)) \right. \\ &\quad \left. - \frac{r_2 \alpha_{21}}{K_2} Y_2(k, i, j) Y_1(k, i, j) \right] \end{aligned}$$

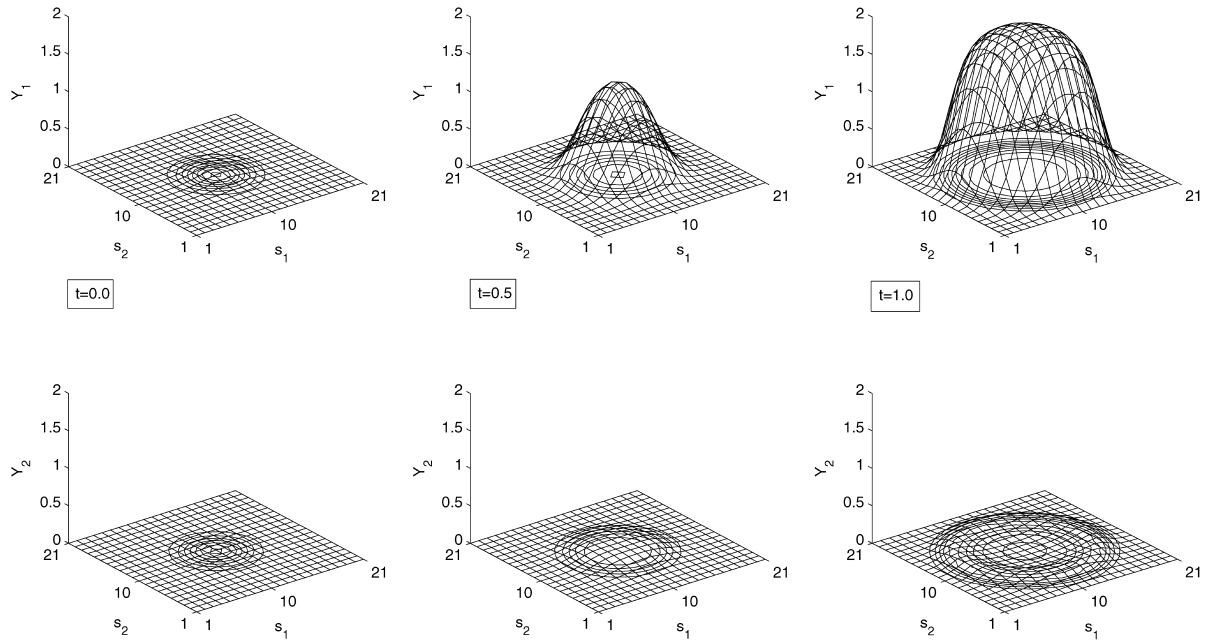


Figure 1. Progression of Competition Between the Two Species ($n_1 = .085$ and $n_2 = .015$).

$$+ \frac{r_2}{K_2} Y_2(k, i, j)(K_2 - Y_2(k, i, j)) - \frac{r_2 \alpha_{21}}{K_2} Y_2(k, i, j) Y_1(k, i, j) \Big].$$

Figure 1 shows that steep spatiotemporal gradients are associated with the solutions, and thus very small increments are needed to resolve them, making the scheme computationally intensive. We also require a coarse grid with increments Δ_{t_C} and Δ_{s_C} , which are multiples of their fine-grid counterparts. The coarse grid used in this example contains $M_C = 40$ temporal points and 20 spatial points in each dimension. The fine grid used contains 12 times more points in each spatiotemporal dimension, and the corresponding numerical solution requires about 3.5 minutes of computational time per run on a 1-GB workstation in MATLAB. Denote by $\mathbf{Y}^C(k, i, j)$ the “reindexed” (for the coarse grid) subset of the fine-grid solution. The approximate local truncation errors are obtained by introducing the fine-grid solution into the scheme corresponding to the coarse grid,

$$\begin{aligned} T_1(k + 1, i, j) &= Y_1^C(k + 1, i, j) - \left[Y_1^C(k, i, j) + \Delta_{t_C} \left[D_1 \right. \right. \\ &\times \left(\frac{Y_1^C(k, i + 1, j) - 2Y_1^C(k, i, j) + Y_1^C(k, i - 1, j)}{\Delta_{s_C}^2} \right. \\ &+ \left. \left. \frac{Y_1^C(k, i, j + 1) - 2Y_1^C(k, i, j) + Y_1^C(k, i, j - 1)}{\Delta_{s_C}^2} \right) \right. \\ &+ \left. \frac{r_1}{K_1} Y_1^C(k, i, j)(K_1 - Y_1^C(k, i, j)) \right. \\ &\left. \left. - \frac{r_1 \alpha_{12}}{K_1} Y_1^C(k, i, j) Y_2^C(k, i, j) \right] \right] \end{aligned}$$

and

$$\begin{aligned} T_2(k + 1, i, j) &= Y_2^C(k + 1, i, j) - \left[Y_2^C(k, i, j) + \Delta_{t_C} \left[D_2 \right. \right. \\ &\times \left(\frac{Y_2^C(k, i + 1, j) - 2Y_2^C(k, i, j) + Y_2^C(k, i - 1, j)}{\Delta_{s_C}^2} \right. \\ &+ \left. \left. \frac{Y_2^C(k, i, j + 1) - 2Y_2^C(k, i, j) + Y_2^C(k, i, j - 1)}{\Delta_{s_C}^2} \right) \right. \\ &+ \left. \frac{r_2}{K_2} Y_2^C(k, i, j)(K_2 - Y_2^C(k, i, j)) \right. \\ &\left. \left. - \frac{r_2 \alpha_{21}}{K_2} Y_2^C(k, i, j) Y_1^C(k, i, j) \right] \right]. \end{aligned}$$

It is assumed that the truncation errors are smoothly correlated in the input space but uncorrelated in time, space, and species dimensions. This allows very fast likelihood maximization of the truncation error model; a single likelihood evaluation takes 1 second. Output prediction, based on $R = 19$ simulations, is then tested at a set of $P = 100$ new input vectors sampled in the input space according to a second maximin Latin hypercube design. The performance of this model has been compared against a benchmark numerical solution (BNCH 1) that uses comparable resources; the associated grid contains twice as many time points and three times as many spatial points in each dimension as the coarse grid. (The truncation errors based model is approximately $R = 19$ times more computationally expensive than a numerical solution computed on the coarse grid, whereas the benchmark numerical solution is approximately 18 times more expensive than a numerical solution computed on the coarse grid.) The TER prediction (including all $R = 19$ simulations) takes about 20 seconds of computational time for a new input vector.

Direct modeling of the output using a dense correlation matrix is not feasible in applications of this size, because it requires a very expensive likelihood maximization. A single likelihood evaluation of the direct model described in Section 2 takes about 1 hour, 40 minutes. This is to be expected because $M_C = 40 = O(10^1)$, $S_{C_1} = 20 = O(10^1)$, and $S_{C_2} = 20 = O(10^1)$. (Because there are two spatial dimensions, the spatial correlation C_S used in the direct modeling is itself a Kronecker product of two correlation matrices, i.e., $C_S = C_{S_1} \otimes C_{S_2}$.) Therefore, the direct modeling likelihood is more expensive than the likelihood of the truncation errors data by at least a factor of $O(10^3)$. For this direct model we could consider using simplifications to facilitate the likelihood computation; however, these simplifications would be made at the expense of model accuracy loss. For example, we could try to fit a model using only a data subset, but this would result in biased parameter estimates. Another simplification that can be considered is to assume a space-specific mean model that ignores the spatial correlation. We tried this approach for this example, but after model fitting, strong spatial correlation still exists, and simply ignoring it will again result in less accurate parameter estimates. The TER method provides both computational efficiency and accurate parameter estimates.

We also used an intermediate-grid numerical solution (with twice as many points in each dimension as in the coarse grid) as covariates, as described in Section 4. A single likelihood evaluation for the truncation error method takes about 2.5 seconds, whereas a single likelihood evaluation for the direct modeling approach takes about 4 hours, 15 minutes. To compensate for the addition of the intermediate numerical solution (which is approximately 8 times more expensive than a numerical solution computed on the coarse grid), the density of grid points used for the benchmark (BNCH 2) was increased to 3 times that of the coarse grid, so that one evaluation was approximately 27 times as expensive as a coarse-grid numerical solution. The covariate-based truncation error prediction including all $R = 19$ simulations takes about 25 seconds of computational time.

The results over the set of $P = 100$ new input vectors are summarized in Table 1. The root mean squared error (RMSE) measures of the truncation error models, with (AI) or without (NAI) auxiliary information, are smaller than those of the benchmark numerical solutions. The AI model has better accuracy than the NAI model (perhaps as expected), but the AI model is also more computationally expensive. The nominal 90% prediction intervals tend to overcover the true fine-grid solution. Figure 2(a) shows boxplots of log maxima over space, time, and species of absolute values of prediction errors at the 100 prediction points. Overall (with the exception of a

few prediction points), the boxplots show that TER NAI performs better than the benchmark numerical solution BNCH 1. The performance of TER AI relative to the benchmark numerical solution BNCH 2 is even better.

5.2 Extension to Several Species

There are real situations in which more than two species must be considered. For example, in the work of Murray (2003, p. 605), the species are several mutating cell populations in a tumor growth. In addition to the diffusion terms, the generalized model includes exponential growth, but Murray pointed out (p. 543) that a logistic growth can be trivially incorporated. The population model that we use in this section includes E equations (species), and each equation contains a diffusion term, a logistic growth term, and $E - 1$ interactions with all other species. More precisely, the PDE system is of the form

$$\frac{\partial \mathcal{Y}_a}{\partial t} = D_a \left(\frac{\partial^2}{\partial s_1^2} + \frac{\partial^2}{\partial s_2^2} \right) \mathcal{Y}_a + \frac{r_a}{K_a} \mathcal{Y}_a (K_a - \mathcal{Y}_a) - \sum_{b \neq a} \frac{r_a \alpha_{ab}}{K_a} \mathcal{Y}_a \mathcal{Y}_b, \quad a = 1, \dots, E.$$

The corresponding computer code becomes increasingly complex and computationally expensive as E increases. For the illustration purposes, we choose $E = 5$ and again set $r_a = 8$, $K_a = 2$, and $\alpha_{ab} = 1.75$, $a \neq b$. The finite-difference scheme and the truncation errors formulas extend naturally from the special case of two species. Due to the increase in the dimension of the input space, we sampled $D = 50$ input vectors according to a maximin Latin hypercube design. Each fine-grid run takes about 11.5 minutes. A single likelihood function evaluation for the TER method takes about 15 seconds without covariates and 40 seconds with covariates. A TER prediction (including all $R = 19$ simulations) takes about 1 minute per run when no covariates are used and 1.5 minutes with covariates. The likelihood maximization of the direct method is again prohibitively expensive; because a single TER likelihood evaluation takes tens of seconds, we would expect a single evaluation of direct modeling likelihood to take tens of hours. (In fact, we stopped the incomplete computation of a single evaluation for the direct likelihood after about 12 hours.) Table 1 gives results over a new set of $P = 100$ prediction inputs, with general patterns similar to those observed for $E = 2$. Figure 2(b) shows boxplots similar to those in Figure 2(a). Comparing TER NAI and the benchmark BNCH 1 according to these boxplots is rather inconclusive, but TER AI shows substantially better overall performance than BNCH 2.

5.3 Numerical Comparison of the TER and Direct Methods

From the previous sections, it is clear that for large-output datasets, results for the direct method are not feasible, and thus comparing it with the TER method is not possible. However, comparison is possible for a simpler case of the example in Section 5.1. Consider the ODE system obtained from the two-species PDE system including only the time component $t \in [0, 3]$ and $D_1 = D_2 = 0$. Set $M_F = 15,000$ (the fine-grid solution takes about 2 seconds) and $M_C = 15$. Set the model parameters as in Section 5.1. Calculation of the likelihood for

Table 1. Prediction Accuracy Measures for the Space–Time Competition Model

	TER NAI	BNCH 1	TER AI	BNCH 2
Two species				
RMSE	.0040	.0158	.0007	.0097
COVER	.9828	—	.9961	—
Five species				
RMSE	.0026	.0052	.0002	.0031
COVER	.9208	—	.9927	—

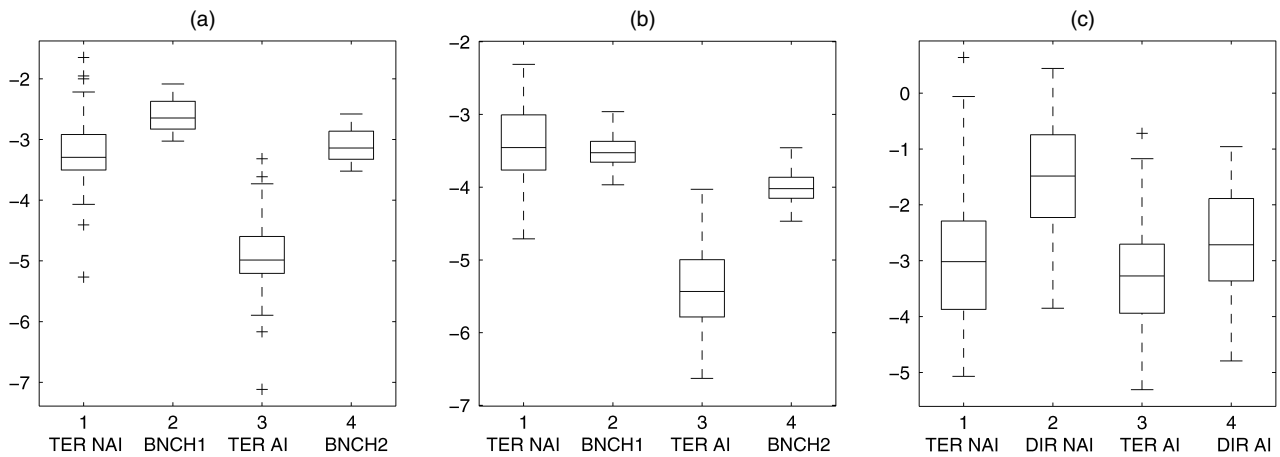


Figure 2. Boxplots of Log Maxima Over Space, Time, and Species of Absolute Value Prediction Errors at the 100 Prediction Points. Each panel illustrates the performance of the TER (NAI and AI) against its competing methods. (a) PDE two-species model; (b) PDE five-species model; (c) ODE two-species model (DIR, direct method).

direct modeling is now slower than that for the TER likelihood by a factor of only $O(10^1)$ and can be optimized relatively easily. The estimate of the parameter appearing in C_M (generated using a Gaussian correlation function) for the direct method without covariates is $\hat{\phi} = 8.3251$, suggesting a strong temporal correlation for the output data. Had we chosen to fit a temporal correlation for the truncation errors data (in which case the TER and direct likelihoods are about equally computationally expensive), the corresponding parameter estimate would be $\hat{\phi} = 82.2117$ (i.e., weak temporal correlation), supporting the assumption of temporal independence for the TER method.

The prediction results for the TER (with temporal independence) and direct methods are summarized in Table 2. The RMSE measures for the TER method are smaller than those of the direct method, especially in the NAI case. The actual prediction interval coverage for the TER method is close to the nominal 90% coverage, whereas the direct method prediction intervals undercover the fine-grid solution. Figure 2(c) shows similar boxplots as in the other two panels. With the exception of a single prediction point (where in fact the TER prediction interval is wider and covers the true value, whereas the direct method does not), overall TER performs better than the direct method. The TER method is more accurate than the direct method because the nonlinear mechanism of the ODE is intrinsically built into the TER predictor.

To better demonstrate the nonlinear/nonstationary feature of the TER model and the qualitative differences between the direct and TER methods, Figure 3 displays point and interval predictions along with the true fine-grid time series at prediction input $(n_1, n_2) = (.0731, .0693)$. The first two columns of Figure 3 correspond to the first species; the last two columns, to the second species. The direct method results are shown in the upper row; the TER results, in the lower row. It is clear that

the code nonlinearity inherited by the TER model helps better predict the outcome of competition. When no covariates are used [Figs. 3(a), 3(c), 3(e), and 3(g)], the direct method appears “unsure” about the binary nature of the outcome of competition, whereas the TER prediction is correct. When covariates are used [Figs. 3(b), 3(d), 3(f), and 3(h)] the direct method improves due to the intermediate numerical solution used as a covariate, although output at the middle of the time segment is still poorly predicted. The TER method with covariates results in better prediction than any of the other methods discussed.

6. CONCLUSION

Many computer codes of scientific interest are implementations of finite-difference schemes for computing approximate solutions to systems of differential equations. As demonstrated herein, relatively simple stochastic processes can be used to model the behavior of the local truncation errors associated with such numerical approximations. These models can serve as the basis for effective nonstationary models of outputs, preserving much of the intrinsic nonlinearity of the underlying system. Further, it is apparent that, at least in some cases, spatiotemporal correlation is not needed in the local truncation error model, whereas it would be absolutely necessary in the corresponding direct model, leading to substantial computational advantages for the truncation error approach.

In general, the accuracy of the truncation error method and the computational resources that it requires depend on the particular code being examined. However, in the examples described herein, models based on truncation errors lead to better output predictions than numerical solutions requiring comparable computational resources. Because local truncation errors often display weak correlation and relatively stationary behavior, even when the output functions do not, this approach can be expected to lead to improved surrogates in many applications.

Table 2. Prediction Accuracy Measures for the ODE Competition Model

	TER NAI	Direct NAI	TER AI	Direct AI
RMSE	.1091	.2747	.0383	.0577
COVER	.8912	.6110	.9241	.7603

APPENDIX: PROOF OF MEAN SQUARED CONVERGENCE FOR THE TER STATISTICAL PREDICTORS

The convergence of the numerical solution to the analytical solution of a differential equation as the grid increments converge to 0 is a

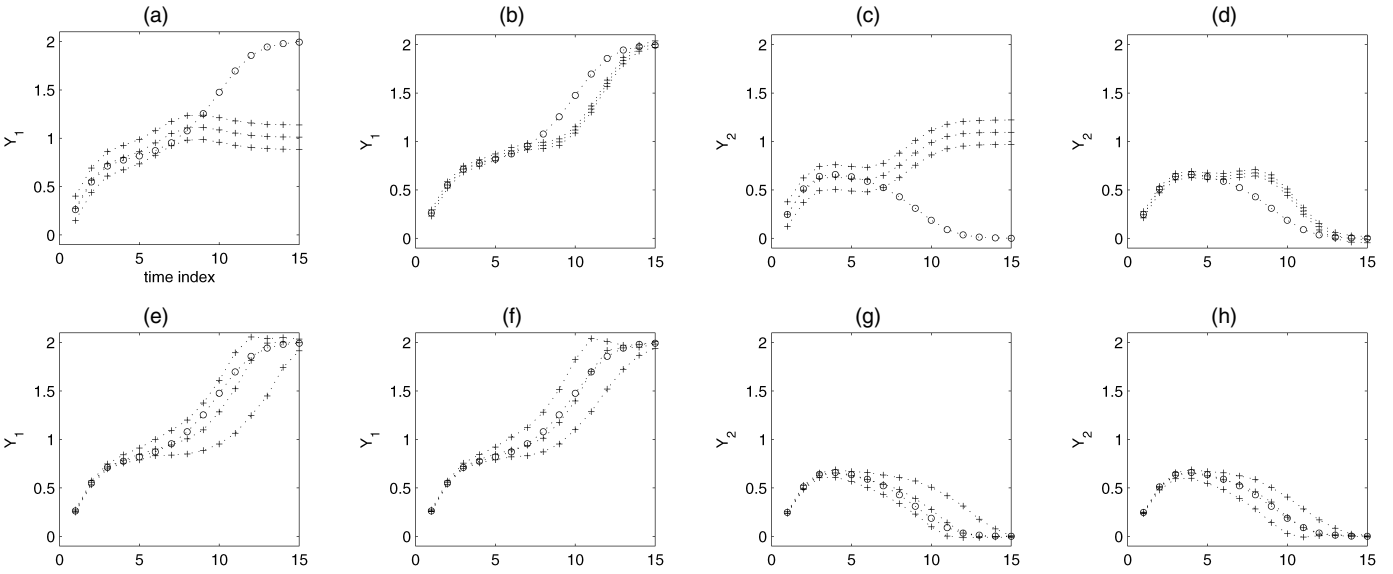


Figure 3. Point and 90% Interval Prediction [PRED, UB 90, LB 90 (---)] Along With Fine-Grid Time Series [TRUE (∘⋯∘)] at Prediction Input $(n_1, n_2) = (.0731, .0693)$ for the ODE Example. Plots (a), (b), (e), and (f) correspond to the first species, (c), (d), (g), and (h), to the second species. Plots (a)–(d) give results for the direct method, and (e)–(h) give the TER results, for models without covariates [(a), (c), (e), and (g)] and with covariates [(b), (d), (f), and (h)].

fundamental property of the finite-difference schemes that must be satisfied. The main concern comes from the fact that the approximation errors made at each step of the iteration could accumulate in such a way that convergence cannot occur. We prove here that a form of convergence for the TER statistical predictor (of numerical solutions) to the analytical solution occurs under conditions similar to those for the convergence of the numerical solution in the deterministic case. First, we introduce the following definitions from Isaacson and Keller (1966, pp. 516, 519). To be closer to their notation, all the terms of the finite-difference scheme (2) are moved to the left side, and then (2) becomes $\mathcal{F}_{\Delta_C}(\mathbf{Y}) = 0$.

Definition A.1. Let $\mathbf{T}^{\mathcal{Y}} = \mathcal{F}_{\Delta_C}(\mathcal{Y})$ be the local truncation errors. The finite-difference scheme is *consistent* if $\|\mathbf{T}^{\mathcal{Y}}\| \rightarrow 0$ as $\|\Delta_C\| \rightarrow 0$, where $\Delta_C = (\Delta_{t_C}, \Delta_{s_C})$.

In this definition $\mathbf{T}^{\mathcal{Y}}$ are the scalar local truncation errors reshaped in vector format. In general, when the increments do not vary across space time (the method of choice in many applications), the components of $\mathbf{T}^{\mathcal{Y}}$ are of the same order of magnitude, denoted generically by $O(T)$ here.

Definition A.2. The finite difference scheme is *stable* if there exists a positive constant K independent of the grid increments, such that $\|U - V\| \leq K\|\mathcal{F}_{\Delta_C}(U) - \mathcal{F}_{\Delta_C}(V)\|$ for any pair of functions U and V defined on the grid.

The main result of this appendix is as follows.

Theorem A.1. If the finite difference is consistent and stable, then the TER predictor of the numerical solution at any new input converges in mean square to the analytical solution of the differential equation as the coarse-grid increments converge to 0.

This result parallels the fundamental result in finite-difference methods, which states that if a scheme is consistent and stable, then the numerical solution converges in norm to the analytical solution of the differential equation as the grid increments converge to 0 (Isaacson and Keller 1966, p. 521). The following auxiliary result is helpful in proving the main result.

Lemma A.1. If the approximate local truncation errors are assumed independent except for input correlation, then the prediction distribution of scalar approximate local truncation errors at new inputs is normal with mean $O(T)$ and variance $O(T^2)$.

Proof. (a) No auxiliary information (NAI). The MLE of μ is $\hat{\mu} = (\mathbf{1}'\Sigma^{-1}\mathbf{T}^{\mathcal{Y}})/(\mathbf{1}'\Sigma^{-1}\mathbf{1}) = (\sum_{i,j=1}^D (C_D)^{-1}_{i,j}(\mathbf{1}'\mathbf{T}^{\mathcal{Y}}_{:,j}))/(\sum_{i,j=1}^D (C_D)^{-1}_{i,j}(\mathbf{1}'\mathbf{1})) = O(T)$, because each component of $\mathbf{T}^{\mathcal{Y}}$ is $O(T)$ as $\|\Delta_C\| \rightarrow 0$. (Here $\mathbf{T}^{\mathcal{Y}}_{:,j}$ is the vector $\mathbf{T}^{\mathcal{Y}}$ reshaped in a matrix so as to separate its inputs dimension from the other dimensions.) The MLE of σ^2 is $\hat{\sigma}^2 = \sum_{i,j=1}^D (C_D)^{-1}_{i,j}(\mathbf{T}^{\mathcal{Y}}_{:,i} - \hat{\mu}\mathbf{1})' \times (\mathbf{T}^{\mathcal{Y}}_{:,j} - \hat{\mu}\mathbf{1})/(M_C S_C E D) = O(T^2)$. The posterior mean at \mathbf{d}_0 is $\tilde{\mu} := \hat{\mu}\mathbf{1} + [(C_{\mathbf{d}_0,D} C_D^{-1}) \otimes \mathbf{I}_{M_C S_C E}](\mathbf{T}^{\mathcal{Y}} - \hat{\mu}\mathbf{1}) = \hat{\mu}\mathbf{1} + \sum_{j=1}^D (C_{\mathbf{d}_0,D} \times C_D^{-1})_j(\mathbf{T}^{\mathcal{Y}}_{:,j} - \hat{\mu}\mathbf{1}) = O(T)\mathbf{1}$, and the posterior variance is $\tilde{\sigma}^2 := \hat{\sigma}^2(1 - C_{\mathbf{d}_0,D} C_D^{-1} C'_{\mathbf{d}_0,D}) = O(T^2)$.

(b) Auxiliary information (AI) used. The MLE of $\mu = [\mu_0, \mu_1]$ is $\hat{\mu} = ((\mathbf{1}, \mathbf{T}^{\mathcal{X}})' \Sigma^{-1} [\mathbf{1}, \mathbf{T}^{\mathcal{X}}])^{-1} ((\mathbf{1}, \mathbf{T}^{\mathcal{X}})' \Sigma^{-1} \mathbf{T}^{\mathcal{Y}})$, assuming that the inverse matrix exists. With the notation $v_{11} = \sum_{i,j=1}^D (C_D)^{-1}_{i,j}(\mathbf{1}'\mathbf{1})$, $v_{12} = \sum_{i,j=1}^D (C_D)^{-1}_{i,j}(\mathbf{1}'\mathbf{T}^{\mathcal{X}}_{:,j})$, $v_{22} = \sum_{i,j=1}^D (C_D)^{-1}_{i,j}(\mathbf{T}^{\mathcal{X}}_{:,i} \mathbf{T}^{\mathcal{X}}_{:,j})$, $u_1 = \sum_{i,j=1}^D (C_D)^{-1}_{i,j}(\mathbf{1}'\mathbf{T}^{\mathcal{Y}}_{:,j})$, and $u_2 = \sum_{i,j=1}^D (C_D)^{-1}_{i,j}(\mathbf{T}^{\mathcal{X}}_{:,i} \mathbf{T}^{\mathcal{Y}}_{:,j})$, it follows that $\hat{\mu}_0 = (v_{22}u_1 - v_{12}u_2)/(v_{11}v_{22} - v_{12}^2)$ and $\hat{\mu}_1 = (-v_{12}u_1 + v_{11}u_2)/(v_{11}v_{22} - v_{12}^2)$. When $\|\Delta_C\| \rightarrow 0$, $\mathbf{T}^{\mathcal{Y}} \approx \mathbf{T}^{\mathcal{X}}$, and thus $u_1 \approx v_{12}$, $u_2 \approx v_{22}$. This implies that $\hat{\mu}_1 = O(1)$. Note that $v_{11}v_{22}u_1 - v_{11}v_{12}u_2 = (v_{11}v_{22} - v_{12}^2)u_1 + (v_{12}u_1 - v_{11}u_2)v_{12}$, and, therefore, $\hat{\mu}_0 = (1/v_{11})(v_{11}v_{22}u_1 - v_{11}v_{12}u_2)/(v_{11}v_{22} - v_{12}^2) = O(1)(u_1/v_{11} - v_{12}/v_{11}) = O(T)$. Finally, the MLE of σ^2 is $\hat{\sigma}^2 = \sum_{i,j=1}^D (C_D)^{-1}_{i,j}(\mathbf{T}^{\mathcal{Y}}_{:,i} - \hat{\mu}_0\mathbf{1} - \hat{\mu}_1\mathbf{T}^{\mathcal{X}}_{:,i})'(\mathbf{T}^{\mathcal{Y}}_{:,j} - \hat{\mu}_0\mathbf{1} - \hat{\mu}_1\mathbf{T}^{\mathcal{X}}_{:,j})/(M_C \times S_C E D) = O(T^2)$. Similar arguments as in (a) demonstrate that the posterior mean is $O(T)\mathbf{1}$ and the posterior variance is $O(T^2)$.

Proof of Theorem A.1. $\|\mathcal{Y} - \tilde{\mathcal{Y}}\| \leq K\|\mathcal{F}_{\Delta_C}(\mathcal{Y}) - \mathcal{F}_{\Delta_C}(\tilde{\mathcal{Y}})\| = K\|\mathbf{T}^{\mathcal{Y}} - \tilde{\mu} - \tilde{\sigma}\| = O(T)\|\mathbf{m} + \boldsymbol{\varepsilon}\|$, where $\boldsymbol{\varepsilon}$ is a vector with iid $N(0, 1)$ components and \mathbf{m} is a vector with $O(1)$ components. Therefore, $E\|\frac{\mathcal{Y} - \tilde{\mathcal{Y}}}{M_C S_C E}\|^2 = O(T^2) \rightarrow 0$ as $\|\Delta_C\| \rightarrow 0$.

Table A.1. Convergence Study for the TER Method,
Two-Species Model

	$(\Delta_{t_C}, \Delta_{s_C})$		
	$(.0250, 1)$	$(.0125, .5)$	$(.0083, .3333)$
RMSE (NAI)	.0126	.0050	.0024
RMSE (AI)	.0025	.0004	.0001

We close this appendix with a discussion of the examples presented in this article. Ames (1965, p. 323) discussed conditions for the stability and consistency of a class of PDEs that includes the species PDE presented here. It can be shown that the scalar local truncation errors for the species PDE example are $O(\Delta_{t_C}) + O(\Delta_{s_C}^2)$, which actually defines $O(T)$. As a demonstration, Table A.1 provides empirical evidence of convergence in mean square for the two-species PDE model, where the RMSE, $\sqrt{E(\|\mathbf{Y} - \tilde{\mathbf{Y}}\|^2)/(M_C S_C E)}$, decreases toward 0 as $\|\Delta_C\|$ decreases toward 0.

[Received March 2004. Revised May 2006.]

REFERENCES

- Ames, W. F. (1965), *Nonlinear Partial Differential Equations in Engineering*, New York: Academic Press.
- Curran, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991), "Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments," *Journal of the American Statistical Association*, 86, 953–963.
- Isaacson, E., and Keller, H. B. (1966), *Analysis of Numerical Methods*, New York: Wiley.
- Kennedy, M. C., and O'Hagan, A. (2000), "Predicting the Output From a Complex Computer Code When Fast Approximations Are Available," *Biometrika*, 87, 1–13.
- (2001), "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society, Ser. B*, 63, 425–450.
- Li, R., and Sudjianto, A. (2005), "Analysis of Computer Experiments Using Penalized Likelihood in Gaussian Kriging Models," *Technometrics*, 47, 111–120.
- Morris, M. D., Mitchell, T. J., and Ylvisaker, D. (1993), "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," *Technometrics*, 35, 243–255.
- Murray, J. D. (2003), *Mathematical Biology II: Spatial Models and Biomedical Applications*, New York: Springer-Verlag.
- Ramsay, J. O., and Silverman, B. W. (1997), *Functional Data Analysis*, New York: Springer-Verlag.
- Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–423.
- Wynn, H. P. (2001), Discussion of "Bayesian Calibration of Computer Models," by M. C. Kennedy and A. O'Hagan, *Journal of the Royal Statistical Society, Ser. B*, 63, 450–451.