Statistical analysis of multivariate computer output

by

Dorin Drignei

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Statistics

Program of Study Committee:
Max D. Morris, Major Professor
William Q. Meeker, Jr.
Michael W. Smiley
Stephen B. Vardeman
Huaiqing Wu

Iowa State University

Ames, Iowa

2004

Graduate College
Iowa State University

This is to certify that the doctoral dissertation of

Dorin Drignei

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Committee Member

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

TABLE OF CONTENTS

LIST OF TABLES vi

LIST OF FIGURES vii

CHAPTER 1. REVIEW OF SCALAR STATISTICAL ANALYSIS FOR COMPUTER EXPERIMENTS 1

CHAPTER 2. STATISTICAL ANALYSIS OF COMPUTER EXPERIMENTS FOR FINITE DIFFERENCE CODES: SMALL TEMPORAL DIMENSION 9

2.1 Introduction 9

2.2 Example: Competing Species Model 12

2.2.1 The Mathematical Model 12

2.2.2 Finite Difference Approximations 13

2.2.3 Local truncation errors 15

2.2.4 Design Selection 16

2.3 Statistical Methodology for Computer Experiments 17

2.3.1 Direct Extension from the Scalar to the Multivariate Case 18

2.3.2 Using Black-Box Information 22

2.4 Coarse Numerical Solutions Used as Auxiliary Information 25

2.5 Results 27

2.6 Conclusion 31

CHAPTER 3. STATISTICAL ANALYSIS OF COMPUTER EXPERIMENTS FOR FINITE DIFFERENCE CODES: LARGE TEMPORAL DIMENSION 43

3.1 Introduction 43

3.2 Output data 44

3.3 Direct approach to statistical modeling of the numerical solutions 53

3.3.1 Modeling the $l(\Lambda_1) \times N_0^x \times N_0^y \times D$ array of numerical solution data . . 53

3.3.2 Modeling the output data at other time points 55

3.3.3 Prediction at a new site γ_0 57

3.4 A statistical model incorporating code information 58

3.4.1 Modeling the $l(\Lambda_2) \times N_0^x \times N_0^y \times D$ array of numerical solution data . . 59

3.4.2 Modeling the intermediary numerical local truncation error data 60

3.4.3 Prediction at a new site γ_0 60

3.5 Coarse numerical solutions as auxiliary information 62

3.5.1 The direct approach 63

3.5.2 Method incorporating code information 64

3.6 Results 65

3.7 Conclusion 69

CHAPTER 4. TWO ADDITIONAL CASES OF MULTIVARIATE COMPUTER OUTPUT STATISTICAL ANALYSES: INTRODUCTORY IDEAS 74

4.1 A statistical pilot study for characterizing the uncertainty in scaling AOGCM results 74

4.2 Stochastic closure for multi-scale simulations 79

APPENDIX A. PROOF OF ALMOST SURE CONVERGENCE FOR THE TER STATISTICAL MODEL 83

A.1 No auxiliary information 83

A.2 Auxiliary information used 85

APPENDIX B. DERIVATION OF THE DISCRETE-TIME BROWNIAN BRIDGE RELATIONSHIP 89

BIBLIOGRAPHY 91

ACKNOWLEDGMENTS 94

LIST OF TABLES

Table 2.1 RMSE measures 29

Table 3.1 Coverage, RMSE and SRMSE measures 68

Table 4.1 RMSE measures 82

Table A.1 Asymptotic study of the noise standard error 88

Table A.2 Asymptotic study of convergence 88

LIST OF FIGURES

Figure 2.1 Sensitivity of species size functions and experimental region 14

Figure 2.2 Predicted and true values at $t = t_8$, no auxiliary information 33

Figure 2.3 Predicted and true values at $t = t_{15}$, no auxiliary information 34

Figure 2.4 Predicted and true values at $t = t_8$, auxiliary information used 35

Figure 2.5 Predicted and true values at $t = t_{15}$, auxiliary information used 36

Figure 2.6 Mean coverage for nominal 90% prediction intervals 37

Figure 2.7 Prediction and true values at testing site A=(0.0463,0.0760) 38

Figure 2.8 Prediction and true values at testing site A=(0.0655,0.0942) 39

Figure 2.9 Prediction and true values at testing site A=(0.0731,0.0693) 40

Figure 2.10 Prediction and true values at testing site A=(0.0069,0.0108) 41

Figure 2.11 Simulations and true values at testing site A=(0.0069,0.0108) 42

Figure 3.1 Design and testing sites used in the computational experiment 49

Figure 3.2 Contour plots of the numerical solutions at the final time point 50

Figure 3.3 Ten simulated realizations of a DBB 56

Figure 3.4 Predicted and true h time series at spatial location $(i, j) = (3, 15)$ . . . 71

Figure 3.5 Predicted and true u time series at spatial location $(i, j) = (4, 15)$ . . . 72

Figure 3.6 Predicted and true v time series at spatial location $(i, j) = (4, 9)$ 73

Figure 4.1 Estimated scale pattern, standard errors and prediction 77

CHAPTER 1. REVIEW OF SCALAR STATISTICAL ANALYSIS FOR COMPUTER EXPERIMENTS

Many scientific investigations rely on computer models for simulating plausible real situations. This is especially useful when physical experimentation is too expensive or even impossible (e.g. climate modeling). In trying to describe the complexities of reality, some computer models are themselves very complex and are therefore expensive to run, in terms of computational resources and time. For example, a single run of a climate model may take weeks on a high performance computer. At the end of the run, the storage and analysis requirements of the output data raise additional difficulties. In response to some of these issues, a recent approach proposes to use statistical models as less computationally demanding surrogates of such complex computer models. More precisely, based on a number of selected runs of the computer model, one builds a statistical model to predict the output of the computer model for untried runs. These statistical surrogates do not exactly match the computer model output in a new situation, but they have the capability to describe the associated uncertainty. Ideally, the completed statistical model would not require as many computational resources as the original computer model. In this chapter we provide a brief survey of the literature dealing with the statistical analysis of computer experiments producing univariate output. We do not discuss the design aspect of the methodology. The notation throughout this chapter corresponds to that used in each paper.

Sacks et al. (1989) was among the first papers to give a detailed account of the motivations for, as well as the issues associated with the design and statistical analysis of computer experiments. In this paper the computer model can be viewed as a deterministic function $y(x)$, in which the output y depends on an input vector x. The input is sometimes called a

site, as this methodology parallels spatial statistics methods. The lack of random error (and associated absence of any need for replication) leads to important distinctions when compared with physical experimentation. For example, the bias is the only component of the adequacy of a response-surface model fitted to the output data. However, the authors point out that statistical components are nevertheless present in this context. Indeed, the selection of the inputs x at which to run the computer model can be viewed as an experimental design problem. Also, the uncertainty characterization of the output predictions resulting from the fitted models is a statistical problem. This paper treats $y(x)$ as a realization of a stochastic process $Y(x)$. Furthermore, it is assumed that Y is, in fact, a regression model

$$Y(x) = \sum_{j=1}^{k} \beta_j f_j(x) + Z(x)$$

where the random process $Z(.)$ has mean zero and covariance $V(x_1, x_2) = \sigma^2 R(x_1, x_2)$. Given a set of inputs $S = (x_1, ..., x_n)$ chosen according to a planned experiment, the authors suggest a kriging approach to statistical analysis and prediction at a new input x. Namely, the best linear unbiased predictor (BLUP) is

$$\hat{y}(x) = f'(x)\hat{\beta} + r'(x)R^{-1}(Y_S - F\hat{\beta})$$

and its mean squared error is

$$MSE[\hat{y}(x)] = E[y(x) - \hat{y}(x)]^2 = \sigma^2[1 - cC^{-1}c']$$

where

$$f(x) = [f_1(x), ..., f_k(x)], F = [f'(x_1), ..., f'(x_k)], R = R(x_i, x_j), r(x) = [R(x_1, x), ..., R(x_k, x)]$$

and $c = (f(x), r(x)), C = \begin{pmatrix} 0 & F' \\ F & R \end{pmatrix}$. The statistical parameters are estimated by maximum likelihood.

In his discussion to the above paper, Morris (1989) points out that the interpretation of the deterministic function $y(x)$ as a realization of a stochastic process might not be appealing to some statisticians who associate this setting with frequentist replication. In a later paper,

Currin et al. (1991) took a Bayesian viewpoint and $y(x)$ was treated as a highly dimensional unknown parameter on which one can put a highly dimensional prior distribution. Thus, the prior knowledge about $y(x)$ is represented by a Gaussian process $Y(x)$. Given a set of inputs $S = (x_1, ..., x_n)$, then Y_S is assumed to be normal with vector mean $E(Y_S) = \mu_S$ and covariance matrix $cov(Y_S, Y_S) = \sigma_{SS}$. Then the prediction at a new site x is given by the posterior mean

$$\hat{y}(x) = \mu_{x|S} = \mu_x + \sigma_{xS}\sigma_{SS}^{-1}(Y_S - \mu_S)$$

and the prediction error variance is given by the posterior variance

$$\sigma_{x|S} = \sigma_{xx} - \sigma_{xS}\sigma_{SS}^{-1}\sigma_{Sx}.$$

The parameters are estimated by maximum likelihood. Under the assumption of normality, the kriging and Bayesian methods give the same point predictions at each x . Notice, however, that the Bayesian formula for the prediction error does not incorporate the prior mean whereas the kriging MSE formula incorporates the prior mean. While a Bayesian approach to the analysis of error-free data can be traced back at least to O'Hagan (1978) and the implementation in Currin et al. (1991) is not fully Bayesian, this paper is an important reference for recent papers implementing fully Bayesian computer output models (to be discussed later). The paper also makes important contributions to the design aspect of the methodology, but we do not discuss them here.

Maximum likelihood estimation is usually the method of choice for fitting stochastic process parameters in both approaches discussed above. A common choice of correlation between the responses at two input vectors x and x' in the above papers is the Gaussian correlation:

$$R(x, x') = \prod_{i=1}^{d} \exp(-\theta_i |x_i - x_i'|^{p_i})$$

with d the input dimension. While a small to moderate number of parameters θ is not a problem for optimization algorithms, some difficulties are noticed when a large number of parameters need to be estimated. Welch et al. (1992) report that an input dimension of $d = 20$ or larger is difficult to handle by optimization algorithms, but suggest that in practical problems only a few dimensions among the d are "active" and require their own correlation

parameter values. The rest of "inactive" dimensions share a common value of θ, which usually is close to zero. They propose a likelihood optimization algorithm that starts with common values for θ and p parameters. At the next stage, according to the "leave-one-out" principle, each dimension is given, by rotation, its own parameters θ and p while the rest of dimensions receive a common set of parameters θ and p. The dimension which leads to the largest increase in likelihood is "separated" from the rest of dimensions and it is entitled to keep its own set of parameters θ and p. The procedure continues until the likelihood does not increase significantly. At that point, the inactive dimensions will share a common, small value of θ. The authors also propose a variation of this algorithm due to the difficulty of handling both parameters θ and p at once. This new algorithm focuses more on the θ parameters as they appear to be more important. Thus, the dimension leading to the largest increase in the likelihood is discovered by allowing only the θ parameters to be different, while the p parameters are kept the same, for computational convenience. In the analysis of computer experiments, the "leave-one-out" principle has also been used in cross-validation in order to assess the fit of the model. Alternatively, one can run the expensive code at new inputs and compare the output against the predictions of the model, provided one can afford to make new runs.

In an application paper illustrating this methodology, Mitchell and Morris (1992) considered two examples. The first example involves a system of differential equations modeling the combustion of methane. The output considered, the ignition delay time in a combustion reaction, is a scalar. The input is 7-dimensional, with each component corresponding to a particular reaction rate of interest. A total of 50 runs were made and the authors considered a product of cubic correlations, one factor corresponding to each input component, to fit these output data. The univariate cubic correlation is given by

$$R(z) = 1 - 6\left(\frac{z}{\theta}\right)^2 + 6\left(\frac{|z|}{\theta}\right)^3 \qquad |z| < \frac{\theta}{2}$$

$$R(z) = 2\left(1 - \frac{|z|}{\theta}\right)^3 \qquad \frac{\theta}{2} \leq |z| < \theta$$

$$R(z) = 0 \qquad |z| \geq \theta,$$

where z is the difference between the values of a common component of two input vectors

and θ is a parameter associated with that input component. The second example involves a finite element simulation model of the compression molding of an automobile hood. The authors considered four inputs to the code, which are physical characteristics of the process, and 15 code runs were made. The output is multivariate in nature and consists of 5 filling times for each of the 469 nodes of the finite element discretization of the mold surface. For reasons of computational efficiency, each component of the multivariate output was modeled as a univariate Gaussian process with component-dependent mean, but the parameters of the cubic correlation were assumed to be common for all components.

The paper by Morris et al. (1993) represents, a departure from the mainstream literature dealing with statistical analysis of computer experiments. The authors consider computer models that produce, at little additional cost, derivatives of the output with respect to the components of inputs. Therefore each run of the model produces a vector of data, composed of the scalar output itself and its d derivatives with respect to the d components of the input vector. From this prospective, one can see it as one of the first papers considering statistical analysis of multivariate computer output. The correlation matrix is composed of block matrices corresponding to correlations between response and derivatives acquired at a site. Another novelty in this paper is that it uses information other than the output itself, which is the information provided by the derivatives. Thus, the authors hint at considering the code as a "gray box" rather than a "black box", having a lot of information to offer besides the output itself. Moreover, the derivative data augments the output data and the authors use this new set of data in covariance, or second-order moments. The most straightforward use of auxiliary information is in the first-order moment, as explanatory variables in a regression model. Due to the non-similarities (in a loose sense) between a function and its derivatives, it is unlikely that this auxiliary information would be of much use as explanatory variable in a regression model.

In an application to an inverse problem, Morris and Solomon (1995) use design and analysis of computer experiments techniques to predict unknown contaminant concentrations, based on measurements of the same concentration function at a finite set of spatio-temporal points.

More precisely, the concentration function $c(x,t)$, which can also be studied theoretically as the solution of a partial differential equation of advection-dispersion on the set $(x,t) \in [0,1] \times [0,1]$, is measured at the end of a one-dimensional tube and a set of data $c(1,t_1),...,c(1,t_n)$ is obtained. The task is to construct a statistical model to predict the initial conditions $c(x,0)$, $x \in [0,1]$, at the beginning of the temporal process. The authors demonstrate the feasibility of the method by selecting six test functions as initial conditions to be predicted.

Aslett et al. (1998) suggest a sequential methodology for optimization problems involving the design and analysis of computer experiments. In this paper, the goal is to optimize a computer output that depends on a set of inputs. The methodology proposes to construct a predictor at the early stages based on an initial design and analysis using methods referenced above. After an assessment of its accuracy, the predictor is used for preliminary optimization. Then new subregion(s) of interest in the input space are identified. The sequential methodology is applied again to this new, smaller input space until no further improvement in the optimization algorithm is noticed. This sequential methodology is demonstrated in an application using a circuit simulator.

McMillan et al. (1999) give an application of the Gaussian correlation R to a protein activity example. The effects of some chemicals on protein activity are investigated and the authors propose the semiparametric model $Y(x) = f^T(x)\beta + Z(x) + \epsilon$, where $Y(x)$ is the observed protein activity and x is a vector of eight categorical explanatory variables. The stochastic terms are $Z \sim N(0, \sigma_Z^2 R)$ and $\epsilon \sim N(0, \sigma_\epsilon^2 I)$. Notice the presence of the random-error term ϵ, which does not appear in the methodology of computer code statistical analysis. This error term implies that the the physical observations Y are not interpolated exactly by this model; this is the usual statistical framework for the analysis of physical observations. It is assumed, however, that an additional "layer" Z of uncertainty with inter-site correlation R is present. Another difference here is that the components of the input are categorical, and the authors propose a modification of the Gaussian correlation. Specifically, if x_i is the i^{th} component of x and has k levels, then the factor $\exp(-\theta_i|x_i - x_i'|^{p_i})$ in the Gaussian correlation is replaced by $\prod_{j=1}^{k} \exp[-\theta_{ij}|I(x_i = j) - I(x_i' = j)|]$, where $I(x_i = j)$ is 1 if x_i takes level j and

zero otherwise.

A few recent papers have described implementations of a fully Bayesian approach to computer experiments. In one of these papers, Kennedy and O'Hagan (2000) show that it is possible to use faster approximations of a slow running computer code as auxiliary information in regression-type models. The authors demonstrate that there are situations where approximations at various levels of accuracy can be used in a more sophisticated model. More precisely, let $z_t(x)$ denote a computer code output of accuracy level t and input x. In their example, z was the output of a hydrocarbon reservoir finite element code with porosity and permeability as inputs and t represented the resolution of the code: the bigger the value of t, the more accurate the code. The output of such a code consists of several time series (various well measurements), but they selected as an output for their example a scalar given by the pressure readings from a single well at a single point in time. The autoregressive model suggested was $z_t(x) = \rho_{t-1} z_{t-1}(x) + \delta_t(x)$, with $\delta(.)$ independent of $z_{t-1}(.), ..., z_1(.)$. The process $\delta(.)$ has a stationary Gaussian distribution generating infinitely differentiable trajectories to reflect the smoothness of the code output. The implementation is fully Bayesian as the authors imposed priors on the parameters associated with the first and second order moments of $z_t(x)$.

Kennedy and O'Hagan (2001) used the statistical models for computer codes presented earlier as ingredients in a Bayesian methodology for calibrating the computer models. In their context, calibration is the operation of choosing the parameters of a mathematical model that best fit a set of physical observations. However, the authors consider the calibration to be a preliminary operation to be addressed before considering other statistical issues, such as prediction of other physical observations. The observations z_i are written as $z_i = \xi(i) + e_i = \rho\eta(x_i, \theta) + \delta(x_i) + e_i$, where e_i is the observation error, $\xi(.)$ is the "true" process, $\eta(.,.)$ is the computer model output, ρ is a regression parameter and $\delta(.)$ is a model inadequacy function. The authors also distinguish between the *calibration* inputs θ (that one wishes to learn about) and the rest of *variable* inputs x having known values for each of the observations used for calibration (e.g. rainfall levels in a hydrological model). The computer code $\eta(.,.)$ has a prior stationary Gaussian distribution, separable with respect to the inputs, and independent of the

stationary Gaussian distributions of $\delta(.)$ and e. The covariance of $\eta(.,.)$ produces infinitely differentiable trajectories, reflecting the smoothness of the code output. Prior distributions are attached to the parameters of the first two moments of $\eta(.,.)$, $\delta(.)$ and e, and a fully Bayesian approach is adopted.

In another application involving a hydrocarbon reservoir, Craig et al. (2001) used physical well pressure data at the bottom of the hole and at various past times from six different wells in order to predict bottom hole well pressure at different future times. The procedure implemented is fully Bayesian. They also have available a computer model of the reservoir running at a user-specified resolution and having various inputs such as the reservoir's geometry, permeability and porosity. The inputs are gathered in a vector x. From various outputs available the authors choose the bottom hole well pressure at various past and future times. A set of n runs $x_1, ..., x_n$ of the computer code is also available. The connection between the set of n output computer model runs and the physical data is made through the assumption that there exists a unique and unknown input x_0 at which the computer model provides the best fit for those data. A prior distribution is associated with x_0. A direct link between the data and the set of n output computer model runs is established by integrating x_0 out of the total likelihood. Finally, a predictive distribution for the bottom hole well pressure is derived, conditioned (among other variables) on the output data collected from n runs of the computer model output at future times.

The next chapters of the dissertation outline methodologies for statistically analyzing multivariate computer output. The examples considered are based on finite difference solvers of differential equations. The motivation comes from the interest in some applications to evaluate the whole spatio-temporal output at various input configurations. Chapter 2 proposes a method aimed at preserving the nonlinearity of such differential equations. It deals specifically with situations in which the output can be easily saved and analyzed. Chapter 3 suggests ways to cope with situations where this is not feasible, especially due to a large temporal dimension of the output. Chapter 4 discusses preliminary work on two additional applications, each based on alternative methodologies.

CHAPTER 2. STATISTICAL ANALYSIS OF COMPUTER EXPERIMENTS FOR FINITE DIFFERENCE CODES: SMALL TEMPORAL DIMENSION

Dorin Drignei and Max D. Morris

2.1 Introduction

Many complex phenomena being subject to scientific inquiries are better understood through the development and use of computer simulation models, or "codes". Over time, the computer codes used for simulations become more and more complex reflecting a continuous improvement in the understanding of these phenomena. However, the increasing complexity of the computer codes is sometimes of limited value due to the accompanying increase in demand on available computational resources. A recent approach to this problem is to develop faster running surrogates for slow running computer codes. These faster (or "cheaper") surrogates are approximations of the original codes, however, and so do not exactly match their outputs. Therefore the error introduced by using a surrogate also needs to be characterized.

Statistical models seem to be a natural choice as surrogates since one can achieve both point-wise prediction and uncertainty (error) characterization. The statistical approach to this problem is outlined in Sacks et al. (1989) and Currin et al. (1991), which are among the earliest works in this area. These papers discuss modeling of computer codes for which inputs can be of very high dimension, but focus on a single scalar-valued output. Input vector values are sometimes called "sites" or "locations" in the input space, following an analogy with geostatistics where related "kriging" methods are used. A computer experiment consists of a number of code runs at various input configurations. Since these runs are sometimes expensive,

it is necessary to choose the inputs carefully in the experimental region to render the statistical surrogates as efficient as possible. This is the design aspect of the statistical methodology. Once the design inputs have been chosen, the computer code output will be obtained and modeled statistically. In many cases the statistical model is described in a Bayesian context where a stationary Gaussian process is prescribed as a diffuse prior for the computer code output at specific inputs. The posterior distribution of the output at a new input site, i.e. one not included in the design, will then act as a surrogate for the slow-running computer code. In other words, one can obtain statistical predictions of the slow-running code output at untried inputs in the experimental region from which the design inputs were selected. The hope is that these statistical predictions are of good quality and at the same time substantially less computationally demanding than a single new run of the computer code. We shall use this Bayesian approach throughout this study.

The statistical methodology described above may not be satisfactory in some situations, however, due to a lack of enough data or violations of various assumptions. Since the data generating mechanism is known, although complicated, a natural place to look for additional information to strengthen the methodology is the computer code itself. This has been referred to in the literature as "opening up the black box" (e.g. Kennedy and O'Hagan 2001). However, due to the wide variety of computer codes available, there is no unified framework for extracting useful information or assessing how much information is enough. Therefore, only isolated attempts to exploit computer code information have been recorded in the literature. For example, an earlier work by Morris at al. (1993) suggested that some codes describing functions produce derivatives of those functions with respect to the components of the input, at little or no additional cost. The authors augmented the output function data with a set of derivative function data, thus increasing the accuracy of predictions. The diffuse prior stationary Gaussian process was used as the statistical surrogate model in this methodology. The question of whether or how to exploit black box information has appeared constantly throughout the literature. Sacks et al. (1989) in their rejoinder pose the question 'Black box or gray box?' and agree with a discussant that subject-matter expertise needs to be reflected in the

statistical methodology whenever possible. More recently, Kennedy and O'Hagan (2001) note that their Bayes approach treats the computer code as simply a black box but acknowledge that exploiting black box information could be a powerful approach. In his discussion to the paper by Kennedy and O'Hagan (2001), Wynn (2001) argues that "it is useful to know what is going on inside the black box", alluding to the need to exploit information about the computer code, especially for differential equation solvers. The differential equation models express relationships that often reflect physical laws (e.g. conservation laws) and involve derivatives of functions of interest with respect to some arguments such as time and/or space. These are deterministic models in the sense that once the relationship and starting values are provided, then the functions of interest (or solutions) can, in principle, be known without error at any spatio/temporal point in the domain of interest. However, in practice, their solutions provide guidance rather than perfect prediction of the observations since any model only approximates the reality. In this study we statistically model virtual observations which are numerical output of such differential equations, as opposed to the physical observations. The solutions of the differential equations, in particular in the nonlinear case, exhibit sophisticated dynamics for which some of the assumptions of the statistical methodology referenced above may fail.

The computer codes considered in this paper are finite difference numerical solvers of differential equations. In general, the input vector for such a computer code is a set of initial conditions and/or a set of parameters and the output is multivariate, such as time series or surfaces. The computer experiments consist of evaluating the multivariate output, i.e. executing the code, for several input vectors and assessing the influence of the inputs on the output. We shall use computer code information and suggest second-order nonstationary statistical models that will mimic more realistically the dynamics of these computer codes. This is particularly useful for nonlinear differential equations since the statistical simulator proposed here inherits the code's nonlinear behavior through the nonstationary structure derived from output data. To illustrate the methodology we shall use a competing species differential system from mathematical ecology. It will be shown that a nonlinear statistical model derived from the structure of the computer code is a good representation of the mathematical model whereas the statisti-

cal model based on a stationary Gaussian prior process for the output fails to represent basic properties. The findings of the analysis are strengthen by a proof that the statistical method proposed here has good asymptotic statistical properties.

2.2 Example: Competing Species Model

2.2.1 The Mathematical Model

Computer models of scientific interest are often based on large systems of nonlinear equations involving many variables. The system we describe here is much simpler, yet it has many of the features of important computer codes and will allow an extensive comparison of the statistical methods to be developed later in this chapter. This example comes from mathematical ecology and is known as the Lotka-Volterra competition model. It is a theoretical description of two species that compete for the same food and are allowed to diminish each other's growth by direct interaction. The dynamics are formulated mathematically as a nonlinear system of two ordinary differential equations which holds for any time t in an interval $[0, L]$:

(2.1a) $$N_1'(t) = \frac{r_1}{K_1} N_1(t)(K_1 - N_1(t)) - \frac{r_1 \alpha_{12}}{K_1} N_1(t) N_2(t)$$

(2.1b) $$N_2'(t) = \frac{r_2}{K_2} N_2(t)(K_2 - N_2(t)) - \frac{r_2 \alpha_{21}}{K_2} N_2(t) N_1(t)$$

The system (1) can be written in a more compact vector form as

(2.1') $$\underline{N}'(t) = \underline{f}(t, \underline{N}(t))$$

with $\underline{f}$ being the bivariate function appearing in the right hand side of (2.1). If initial conditions $N_1(0)$ and $N_2(0)$ are provided then this system will have a unique pair of solutions $(N_1(t), N_2(t))$. These represent the two species sizes N_1 and N_2 at a point t in time and are continuously differentiable. As determined by (2.1), the species size temporal evolutions are influenced by a logistical term of growth rate r for each species and interaction terms between the two species characterized by the parameters α. Here K_1 and K_2 are called the saturation levels of the two species and are positive constants. The model has a rich set of dynamics depending on the relationship between the various parameters appearing in the system (2.1). Kot (2001) provides a full analysis of these dynamics. Of particular interest to us is the case $\alpha_{12} > \frac{K_1}{K_2}$ and $\alpha_{21} > \frac{K_2}{K_1}$ when one species eventually becomes extinct and the other converges

to the corresponding saturation level K, depending on the initial conditions. More precisely, if $N_1(0) > N_2(0)$ then $N_1(t)$ converges asymptotically to K_1 whereas $N_2(t)$ converges to zero. Likewise, if $N_1(0) < N_2(0)$ then $N_2(t)$ converges to K_2 whereas $N_1(t)$ converges to zero. If $N_1(0) = N_2(0)$ then no species wins the contest and they will share the object of competition.

In this study we will conduct a computer experiment to investigate the influence of the initial conditions $N_1(0)$ and $N_2(0)$. We fix the parameters in model (2.1) to some values that correspond to the above scenario: $K_1 = K_2 = 2, \alpha_{12} = \alpha_{21} = 1.75$. Also, we fix $r_1 = r_2 = 8$ and limit our study to the time interval $[0, L] = [0, 3]$. The experimental values $(N_1(0), N_2(0))$ will be selected from the square $[0.005, 0.1] \times [0.005, 0.1]$. We have chosen not to include the case $(N_1(0), N_2(0)) = (0, 0)$ in order to avoid the resulting trivial solution. Figure 2.1a-c illustrates the three typical cases mentioned above. However, depending on the specific values of the initial conditions, the species sizes N_1 and N_2 will approach their limiting values faster or slower. For example, if $N_1(0)$ is much bigger than $N_2(0)$ then the first species wins the competition very quickly. On the other hand, if $N_1(0) > N_2(0)$ but their difference is small then species 1 still wins the competition but in a longer time. Figure 1d shows the region of initial conditions considered and a sample of $D = 20$ design points in this region. The problem in this study is to find statistical models that predict species size functions N_1, N_2 corresponding to new pairs of initial values in this region without actually solving the system (2.1), given that the system has been solved for the set of $D = 20$ pairs of initial values appearing in Figure 2.1d. (Over-plotted is a set of $P = 100$ testing sites.)

2.2.2 Finite Difference Approximations

Nonlinear systems of differential equations can rarely be solved analytically, even though solutions exist. In these situations one relies on numerical approximations of the true solution. A common practice is to approximate the derivatives involved by scaled differences of the corresponding function evaluated at neighboring points on a grid. The resulting procedure is called the finite differences method. In our case, the time interval $[0, L]$ is divided into M subintervals of equal size $h = \frac{L}{M}$. Denote by $t_0, t_1, ..., t_M$ the end points of these subintervals.

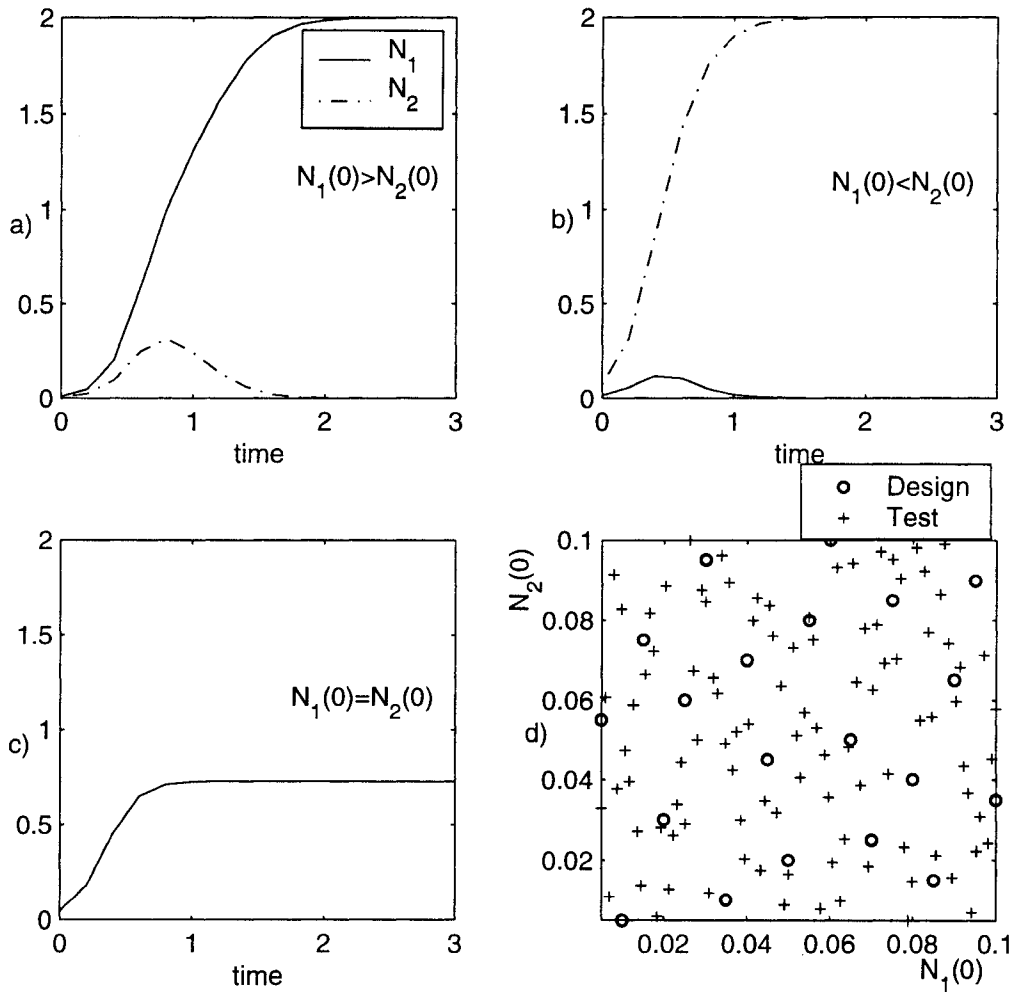

Figure 2.1 Sensitivity of species size functions and experimental region

The system (2.1) can be rewritten in the numerical form

(2.2a) $\frac{\tilde{N}_1(t_{i+1})-\tilde{N}_1(t_i)}{h} = \frac{r_1}{K_1}\tilde{N}_1(t_i)(K_1 - \tilde{N}_1(t_i)) - \frac{r_1\alpha_{12}}{K_1}\tilde{N}_1(t_i)\tilde{N}_2(t_i)$

(2.2b) $\frac{\tilde{N}_2(t_{i+1})-\tilde{N}_2(t_i)}{h} = \frac{r_2}{K_2}\tilde{N}_2(t_i)(K_2 - \tilde{N}_2(t_i)) - \frac{r_2\alpha_{21}}{K_2}\tilde{N}_2(t_i)\tilde{N}_1(t_i)$

for $i = 0, 1, ..., M-1$, or equivalently

(2.2a') $\tilde{N}_1(t_{i+1}) = \tilde{N}_1(t_i) + h[\frac{r_1}{K_1}\tilde{N}_1(t_i)(K_1 - \tilde{N}_1(t_i)) - \frac{r_1\alpha_{12}}{K_1}\tilde{N}_1(t_i)\tilde{N}_2(t_i)]$

(2.2b') $\tilde{N}_2(t_{i+1}) = \tilde{N}_2(t_i) + h[\frac{r_2}{K_2}\tilde{N}_2(t_i)(K_2 - \tilde{N}_2(t_i)) - \frac{r_2\alpha_{21}}{K_2}\tilde{N}_2(t_i)\tilde{N}_1(t_i)]$.

For sufficiently small h, the sequence $(\tilde{N}_1, \tilde{N}_2)$ approximates (N_1, N_2) at $t_0, t_1, ..., t_M$. The time step h needs to be small in order to guarantee that accumulated errors in (2.2') are bounded. The numerical scheme (2.2) is called the Euler's method which is the simplest method based on finite differences. We mention that there are many other finite difference methods, some of them better than Euler's (see Lambert 1991), but we choose the later because its simplicity is helpful in introducing the ideas of this study.

2.2.3 Local truncation errors

The system (2.2') can be rewritten as

(2.3a) $0=\tilde{N}_1(t_{i+1}) - [\tilde{N}_1(t_i) + h(\frac{r_1}{K_1}\tilde{N}_1(t_i)(K_1 - \tilde{N}_1(t_i)) - \frac{r_1\alpha_{12}}{K_1}\tilde{N}_1(t_i)\tilde{N}_2(t_i))]$

(2.3b) $0=\tilde{N}_2(t_{i+1}) - [\tilde{N}_2(t_i) + h(\frac{r_2}{K_2}\tilde{N}_2(t_i)(K_2 - \tilde{N}_2(t_i)) - \frac{r_2\alpha_{21}}{K_2}\tilde{N}_2(t_i)\tilde{N}_1(t_i))]$.

The local truncation errors are defined as the (generally non-zero) quantities obtained by introducing the true solution (N_1, N_2) of the differential system into the right side of these equations, that is

(2.4a) $T_1(t_{i+1}) = N_1(t_{i+1}) - [N_1(t_i) + h(\frac{r_1}{K_1}N_1(t_i)(K_1 - N_1(t_i)) - \frac{r_1\alpha_{12}}{K_1}N_1(t_i)N_2(t_i))]$

(2.4b) $T_2(t_{i+1}) = N_2(t_{i+1}) - [N_2(t_i) + h(\frac{r_2}{K_2}N_2(t_i)(K_2 - N_2(t_i)) - \frac{r_2\alpha_{21}}{K_2}N_2(t_i)N_1(t_i))]$.

Taking into account (2.1) and Taylor series arguments, the system (2.4) can be rewritten as

(2.5a) $T_1(t_{i+1}) = N_1(t_{i+1}) - [N_1(t_i) + hN_1'(t_i)] = \frac{h^2}{2}N_1''(t_i) + R_1$

(2.5b) $T_2(t_{i+1}) = N_2(t_{i+1}) - [N_2(t_i) + hN_2'(t_i)] = \frac{h^2}{2}N_2''(t_i) + R_2$.

Assuming that N_1 and N_2 have finite second order derivatives at any point in $[0, L]$, one obtains $T_1(t_{i+1}) = O(h^2)$ and $T_2(t_{i+1}) = O(h^2)$. The reminders R_1 and R_2 contain h of order 3 or higher and are considered negligible in this study. Hence the Euler's finite difference method

used here is said to be of order 2.

The local truncation error can be thought of as a residual quantity and provides a measure of accuracy of the finite difference scheme. To draw a parallel with statistics where the residuals play a key role, let's consider $\underline{Y}$ to be a vector of data and $\underline{X}$ be an "explanatory" vector of $\underline{Y}$ in a simple linear regression model. If $\underline{\hat{Y}}$ denotes the predictor of $\underline{Y}$ then $0 = \underline{\hat{Y}} - (\beta_0 + \beta_1 \underline{X})$ for some parameters β_0, β_1. However, if one introduces the data vector into this prediction relationship, then one obtains a vector of residuals $\underline{\epsilon} = \underline{Y} - (\beta_0 + \beta_1 \underline{X})$ which represent a measure of accuracy of the regression prediction. Clearly there are similarities between the two situations: the data vector is analogous to the true solution of the differential equation, the regression predictor is similar to the numerical solution of the finite difference equation, the regression relationship is similar to the finite difference scheme and finally the regression residuals are akin to the local truncation errors.

As we shall see later, output predictions based on a statistical model of local truncation errors can be more accurate than statistical predictors derived directly from observed output. The main gain is the preservation of the nonlinearity of the system, resulting in more realistic statistical predictions.

2.2.4 Design Selection

The design aspect of the problem is concerned with the choice of the input vectors, in our case with the pair of initial values $(N_1(0), N_2(0))$ within the region of interest $[0.005, 0.1] \times [0.005, 0.1]$. One approach is to choose the input vectors to cover the space thoroughly in some sense. Let $d_1, d_2, ..., d_D$ be the D input vectors to be chosen. A Latin Hypercube Design (LHD) is defined as follows. Denote by π an arbitrary permutation of $1, 2, ..., D$. Divide each axis of the region into $D-1$ intervals of equal length and denote by $i_1, i_2, ..., i_D$ the end points. Then define $d_k = (i_k, i_{\pi(k)})$ for $k = 1, 2, ..., D$. While an LHD has good one-dimensional projections, it is not always an attractive design; for example, a design that puts all the points on the main diagonal of the design region is an LHD. The LHD can be more useful when it is constructed in conjunction with other criteria that quantify other desirable design properties. A maximin

distance design of minimum index (or simply a maximin) is a set of points $d_1, d_2, ..., d_D$ that: a) maximizes the minimum intersite distance $\underline{d}(D) = \min_{i,j}[\text{dist}(d_i, d_j)]$, and b) minimizes the index $J(D)$, the number of pairs $(i, j), i < j$ for which $\text{dist}(d_i, d_j) = \underline{d}(D)$. Unrestricted maximin designs of moderate size generally contain a large proportion of points on the boundary of the design region and therefore do not have good one-dimensional properties. On the other hand, unrestricted LHDs are sometimes referred to as "space filling" designs. Morris et al. (1993) considered compromise designs and showed evidence that a maximin design within the class of LHDs can perform better than either an LHD constructed from randomly selected permutation π, or an unrestricted maximin design. The search for a maximin LHD was based on a point-exchange algorithm. Here we shall take a less sophisticated but practical approach, by simply selecting the maximin LHD within a relatively large list of randomly chosen LHDs. In Figure 2.1d is illustrated the experimental region for the pair of initial conditions along with a set of $D = 20$ design sites. This design was selected according to a maximin criterion from a list of 1,000,000 randomly generated LHDs, and had a minimum inter-point distance of $\underline{d}(D) = 0.0180$, and index $J(D) = 7$. While this empirical approach is perhaps easier to implement and more straightforward than the exchange-point algorithm, it has the main disadvantage of not searching for a (local) optimum in a coherent way and therefore not focusing all the resources in the "correct direction". Unless the number of input dimensions is relatively small so that a reasonably large list of randomly chosen LHDs can include many possible LHDs, it is perhaps better to use a search algorithm than this empirical approach.

2.3 Statistical Methodology for Computer Experiments

The aim of this methodology is to obtain statistical models that predict a fine-grid numerical solution for a new set of initial condition pairs, given that the fine-grid simulator has been run for a well-chosen set of initial conditions pairs. This problem originates in an effort to find less expensive surrogates for computationally expensive simulators. While direct application of Euler's method to the Lotka-Volterra competition model is not computationally demanding, it has the advantage that the statistical surrogates developed here can be tested against fine

grid validation solutions computed for a large set of new inputs. Empirical assessment of predictor performance cannot be based on the data used to fit the statistical model, since all the statistical predictors presented in this study will interpolate exactly the output data at the design sites.

2.3.1 Direct Extension from the Scalar to the Multivariate Case

2.3.1.1 Output Data

The Euler's finite difference method (2.3) was used to obtain numerical solutions on a fine grid of $M_f = 15,000$ time intervals for each of the $D = 20$ pairs of initial conditions plotted in Figure 2.1d. We have chosen such a large M_f due to the slow convergence of the Euler method. The numerical solutions were retained at time points corresponding to a coarser sub-grid of $M_c = 15$ intervals (e.g. every 1000^{th} point from the computed series). Note that this is not equivalent to running the code (2.3) for $h_c = L/M_c$. Qualitatively, the numerical solutions obtained on the fine grid for $h_f = L/M_f$ are closer to the exact solution of (2.1) than the numerical solutions that would be obtained on the coarse grid for $h_c = L/M_c$. In the sequel, M will be equal to M_c if not otherwise specified. Since there are two species size functions, the data to be analyzed is an $M \times 2 \times D$ array. We do not include the data at the starting time since we always know what the initial conditions are and there is no need to regard them as data to be modeled. As a numerical feature, all the data must be in the interval $(0, 2)$ since one cannot have species of negative size and the species size cannot grow beyond the saturation constants $K_1 = K_2 = 2$. Notice that the $M \times 2 \times D = 15 \times 2 \times 20$ array is relatively small which allows saving and analyzing the output data with a relatively small effort. In our case the number of time points (or temporal dimension) was relatively small. However, there are situations (see next chapter) where this number of time points cannot be chosen as small. Different techniques need to be applied to the resulting large set of output data.

Most of the current statistical methodology for computer experiments treats the output as a scalar. It has been suggested in the literature (Kennedy and O'Hagan, 2001, p.435) that instead of treating the output as multivariate, one can augment the set of input variables by

a number of variables equal to the number of output dimensions and then treat the output as scalar. For example, in our case, there are two input variables (the two initial conditions) and two output dimensions (species and time). The suggestion is to index the output by four inputs and apply the current methodology for scalar output. This amounts to $2DM$ design sites and leads to the specification of a $(2DM) \times (2DM)$ covariance matrix for all data, which is numerically unstable for practical values of D and M unless more assumptions are made. (One such assumption allowing decomposition of the variance matrix into a Kronecker product is discussed in this study.) At each input point there are two functions that represent the species size over time. Let $Y^i_{d,t}$ denote the output value produced by the fine grid solver for species $i = 1, 2$, input vector d (i.e. pair of initial conditions) and time point t. This is a shorter version of the more cumbersome, yet complete, notation $\tilde{N}$ used in section 2.2.2. We will consider a statistical model in which the two species sizes are correlated, since the mathematical model that produced them specifies interaction between species.

2.3.1.2 The likelihood function

The data are a three dimensional array of size $M \times 2 \times D$. This arrangement facilitates the specification of a covariance matrix as a Kronecker product. Since the two theoretical solutions of the system (2.1) are infinitely differentiable in time, one can assume a Gaussian correlation corresponding to the time dimension, i.e.

$$C_T(t_i, t_j) = \exp(-\theta_T(t_i - t_j)^2), \quad i, j = 1, ..., M.$$

This correlation produces infinitely differentiable trajectories. Next we will represent the fact that the two species size functions are correlated, with the correlation matrix

$$C_2 = \begin{pmatrix} 1 & p \\ p & 1 \end{pmatrix}.$$

Finally, we will assume again a Gaussian correlation corresponding to the two dimensional input of initial conditions. The reason for this choice is a theoretical result stating that the solutions of differential equations depend smoothly on the initial conditions as long as the function $\underline{f}$ in

(2.1') is smooth in all its variables (e.g. Hartman 1964). Thus, the input correlation will be the product correlation

$$C_D(d_i, d_j) = \exp(-\theta_{D_1}(d_i(1) - d_j(1))^2 - \theta_{D_2}(d_i(2) - d_j(2))^2), \quad i, j = 1, ..., D$$

and d_i is the i^{th} two dimensional input vector of initial conditions, $(N_1(0), N_2(0))$. Let $\underline{Y}$ denote the reshaped, vectorized form of the three dimensional array of data $\{Y^i_{d,t}\}$, with the time as the inner dimension and input as the most outer dimension. It will be assumed that

$$\underline{Y} = N(\mu\underline{1}, \sigma^2\Sigma), \ \Sigma = C_D \otimes (C_2 \otimes C_T),$$

where "$\otimes$" represents the Kronecker product. The likelihood of the multivariate normal distribution can be written as

$$L(\theta_T, p, \theta_{D_1}, \theta_{D_2}, \mu, \sigma^2|\underline{Y}) = (2\pi\sigma)^{-2MD/2}|\det(\Sigma)|^{-1/2}\exp[-\frac{1}{2\sigma^2}(\underline{Y} - \mu\underline{1})'[\Sigma]^{-1}(\underline{Y} - \mu\underline{1})].$$

Maximum likelihood estimates (MLEs) of the parameters involved $(\theta_T, p, \theta_{D_1}, \theta_{D_2}, \mu, \sigma^2)$ will be obtained by maximizing the likelihood function above. In manipulating the likelihood function, it is worth pointing out some algebraic simplifications associated with the Kronecker product, specifically $[\Sigma]^{-1} = C_D^{-1} \otimes (C_2^{-1} \otimes C_T^{-1})$ and $\det(\Sigma) = \det(C_D)^{2M}\det(C_2)^{MD}\det(C_T)^{2D}$. These are helpful in avoiding numerical instabilities. The MLE for μ is $\hat{\mu} = \frac{\underline{1}'[\Sigma]^{-1}\underline{Y}}{\underline{1}'[\Sigma]^{-1}\underline{1}}$ and the MLE for σ^2 is $\hat{\sigma}^2 = \frac{1}{2MD}(\underline{Y} - \underline{1}\mu)'[\Sigma]^{-1}(\underline{Y} - \underline{1}\mu)$. The other parameter estimates are obtained numerically by iterative maximization of the likelihood function, or equivalently by minimizing the objective function

$$l(\theta_T, p, \theta_{D_1}, \theta_{D_2}|\underline{Y}) = \log(\hat{\sigma}^2) + \frac{\log(\det(C_D))}{D} + \frac{\log(\det(C_2))}{2} + \frac{\log(\det(C_T))}{M}.$$

The numerical values of the MLEs in our demonstration exercise are: $\hat{\mu} = 0.6599$, $\hat{\sigma}^2 = 0.3956$, $\hat{\theta}_T = 8.3251, \hat{\theta}_{D_1} = 7.5660, \hat{\theta}_{D_2} = 6.1805, \hat{p} = -0.4564$.

The correlation between time series at each location is estimated to be negative. This is to be expected since, on approximately the last two thirds of the time interval $[0, L]$, the two functions behave completely differently depending on who wins the competition and on approximately the first third they behave more or less similarly. Therefore the correlation

parameter p is expected to be negative, although not close to -1. Because of this behavior, one could argue that it might be possible to improve our model by specifying a C_2 that varies with time t. However, this would require care since the overall correlation would no longer be of product form.

2.3.1.3 Prediction

Let $d_0 = (N_1(0), N_2(0))$ be a new input vector, that is a pair of initial conditions not included in the D-point experimental design. The goal of the statistical model is to predict the fine-grid solution at the M coarse-grid time points, without actually running the fine-scale simulator at d_0. (In fact, here we'll run the fine-scale simulator off-line at selected d_0 vectors, but only to assess the accuracy of our statistical predictors.) Let C_{D+} be the design covariance matrix C_D augmented by a row and column corresponding to the correlations between the output data at the new input vector d_0 and the D design input vectors, that is the correlation matrix corresponding to the augmented input set $D^+ = \{d_1, ..., d_D, d_0\}$. Then the prior covariance matrix corresponding to the $M \times 2 \times (D+1)$ augmented array of output data is $\Sigma^+ = C_{D+} \otimes (C_2 \otimes C_T)$. Denote by $\Sigma^+_{d_0,D}$ the matrix corresponding to the first $2MD$ elements of each of the last $2M$ rows of Σ^+ and, likewise, $C_{d_0,D}$ the vector of first D elements of the last row of C_{D+}. Notice that

$$\Sigma^+_{d_0,D}[\Sigma]^{-1} = (C_{d_0,D} \otimes (C_2 \otimes C_T))(C_D^{-1} \otimes (C_2^{-1} \otimes C_T^{-1})) =$$

$$(C_{d_0,D}C_D^{-1}) \otimes ((C_2 \otimes C_T)(C_2^{-1} \otimes C_T^{-1})) = (C_{d_0,D}C_D^{-1}) \otimes I_{2M}.$$

Then the posterior mean vector at the new site d_0, which is the usual point predictor suggested by minimization of expected squared-error loss, is

$$\hat{\underline{Y}}_{d_0} = \mu\underline{1} + \Sigma^+_{d_0,D}[\Sigma]^{-1}(\underline{Y} - \mu\underline{1}) = \mu\underline{1} + [(C_{d_0,D}C_D^{-1}) \otimes I_{2M}](\underline{Y} - \mu\underline{1}).$$

$\hat{\underline{Y}}_{d_0}$ may be "de-vectorized", that is reshaped in a $2 \times M$ array corresponding to the two species size functions at d_0. The last expression shows that the posterior mean involves only the correlation between the design sites, not the correlation matrices C_2 and C_T. Moreover, the posterior mean of a species at a time point t_i depends only on the output data for that

species at the same time point t_i. Denote by $\Sigma^+_{d_0}$ the matrix corresponding to the last $2M$ elements of each of the last $2M$ rows of Σ^+ and, likewise, C_{d_0} the last element of the last row of C_{D^+}. To derive a similar expression for the posterior covariance, notice that

$$\frac{1}{\sigma^2}COV(\underline{Y}_{d_0}, \underline{Y}) = \Sigma^+_{d_0} - \Sigma^+_{d_0,D}[\Sigma]^{-1}{\Sigma^+_{d_0,D}}' =$$

$$C_{d_0} \otimes (C_2 \otimes C_T) - (C_{d_0,D} \otimes (C_2 \otimes C_T))(C_D^{-1} \otimes (C_2^{-1} \otimes C_T^{-1}))(C'_{d_0,D} \otimes (C_2 \otimes C_T)) =$$

$$(C_{d_0} - C_{d_0,D}C_D^{-1}C'_{d_0,D})(C_2 \otimes C_T).$$

The posterior covariance at d_0 is

$$\sigma^2(\Sigma^+_{d_0} - \Sigma^+_{d_0,D}[\Sigma]^{-1}{\Sigma^+_{d_0,D}}') = \sigma^2(C_{d_0} - C_{d_0,D}C_D^{-1}C'_{d_0,D})(C_2 \otimes C_T).$$

Notice that the main diagonal of the posterior covariance at d_0 gives the marginal posterior variances at each $i = 1, 2$ and $t = 1, ..., M$. That is, the marginal posterior variances are all equal to $\sigma^2(C_{d_0} - C_{d_0,D}C_D^{-1}C'_{d_0,D})$, which again involves only the correlation between d_0 and the design set. The prediction standard errors are defined as the square root of the marginal posterior variances.

2.3.2 Using Black-Box Information

In section 2.2.3 we reviewed the concept of local truncation error as a measure of accuracy of the finite difference scheme. We drew a parallel with statistics and showed that the local truncation errors are similar in some ways to statistical residuals. In this section we will model the numerical local truncation errors as opposed to modeling the numerical solutions as described in section 2.3.1. By doing this, we will make use of numerical information about the accuracy of the computer code and therefore here it will no longer be treated as a black-box.

In section 2.3.1.1 we described the output data as a $M \times 2 \times D$ array containing the subset of the fine-grid numerical solution that corresponds to the coarse grid. Here we will obtain numerical local truncation errors by replacing the true solutions with their fine-grid approximations (denoted $Y^i_{d,t}$ in section 2.3.1) in relationship (2.4):

(2.6a) $\quad T^{1,Y}_{d,t_{i+1}} = Y^1_{d,t_{i+1}} - [Y^1_{d,t_i} + h_c(\frac{r_1}{K_1}Y^1_{d,t_i}(K_1 - Y^1_{d,t_i}) - \frac{r_1\alpha_{12}}{K_1}Y^1_{d,t_i}Y^2_{d,t_i})]$

(2.6b) $T^{2,Y}_{d,t_{i+1}} = Y^2_{d,t_{i+1}} - [Y^2_{d,t_i} + h_c(\frac{r_2}{K_2}Y^2_{d,t_i}(K_2 - Y^2_{d,t_i}) - \frac{r_2\alpha_{21}}{K_2}Y^2_{d,t_i}Y^1_{d,t_i})]$

for $i = 0, ..., M-1$ and $d = 1, .., D$.

Let $\underline{T}^Y$ denote the vectorized form of the numerical local truncation errors $T^{j,Y}_{d,t}$ $(j = 1, 2)$. Following the general arguments presented in section 2.3.1 for modeling $\underline{Y}$, we now use the same approach to model $\underline{T}^Y$ as a function of initial conditions, species and time. In particular, we specify a covariance matrix that is a scaled Kronecker product of three correlation matrices corresponding to its three dimensions prior to vectorization,

$$COV(\underline{T}^Y) = \tau^2\Sigma = \tau^2(C_D \otimes (C_2 \otimes C_T)).$$

We shall assume again

$$C_D(d_i, d_j) = \exp(-\theta_{D_1}(d_i(1) - d_j(1))^2 - \theta_{D_2}(d_i(2) - d_j(2))^2), \quad i, j = 1, ..., D$$

and

$$C_2 = \begin{pmatrix} 1 & p \\ p & 1 \end{pmatrix}.$$

Unlike in the case of direct modeling of the numerical solutions (section 2.3.1.), here time-independence will be assumed for the numerical local truncation errors, that is $C_T = I_M$. We shall prove in the Appendix A that if the numerical local truncation errors are assumed time-independent, the statistical predictor converges almost surely to the exact solution of the ODE system when the coarse time step $h_c = L/M_c$ converges to zero. In other words, the statistical predictor has good properties as long as the coarse time step $h_c = L/M_c$ is small enough. The numerical results show that time-independence of the numerical local truncation errors does not necessarily mean time-independence of the predicted numerical solutions. In fact, the point predictors of the numerical solutions will be quite smooth as functions of time. This is not to say that if one assumes time-dependence for the numerical local truncation errors the statistical method does not work. We fitted both linear and cubic correlations for numerical local truncation errors in the time dimension (not shown here) and the method seems to work well. However, we see the time-independence assumption as an important practical advantage since it eliminates the need to estimate parameters associated with C_T. Maximum likelihood

estimates (MLEs) of the parameters $(\mu, p, \theta_{D_1}, \theta_{D_2}, \tau^2)$ will be obtained by maximizing the likelihood function, which is similar to the one presented in section 2.3.1.2. The MLE of μ is $\hat{\mu} = \frac{\underline{1}'[\Sigma]^{-1}\underline{T}^Y}{\underline{1}'[\Sigma]^{-1}\underline{1}}$ and the MLE of τ^2 is $\hat{\tau}^2 = \frac{1}{2MD}(\underline{T}^Y - \underline{1}\mu)'[\Sigma]^{-1}(\underline{T}^Y - \underline{1}\mu)$. The MLEs of $(p, \theta_{D_1}, \theta_{D_2})$ are obtained numerically by minimizing the objective function

$$l(p, \theta_{D_1}, \theta_{D_2}|\underline{T}^Y) = \log(\hat{\tau}^2) + \frac{\log(\det(C_D))}{D} + \frac{\log(\det(C_2))}{2}.$$

The MLEs are: $\hat{\mu} = -0.0020, \hat{\tau}^2 = 0.0016, \hat{\theta}_{D_1} = 6.7505, \hat{\theta}_{D_2} = 5.8905, \hat{p} = -0.3445$. There seems to be a significant negative correlation between local truncation error time series, as was also the case in our direct modeling of the output series in section 2.3.1. It was expected that $\hat{\mu}$ is somewhat close to zero because the local truncation errors represent deviations from zero, both positive and negative.

In order to predict at a new input vector $d_0 = (N_1(0), N_2(0))$ we shall use a simulation approach. The algebraic formulae of section 2.3.1. also apply here for predicting local truncation errors. The posterior mean is

$$\underline{\hat{T}}^Y_{d_0} = \mu\underline{1} + [(C_{d_0,D}C_D^{-1}) \otimes I_{2M}](\underline{T}^Y - \mu\underline{1})$$

and the posterior covariance is

$$\tau^2(C_{d_0} - C_{d_0,D}C_D^{-1}C'_{d_0,D})(C_2 \otimes I_M).$$

The Choleski matrix of the posterior covariance is

$$\tau\sqrt{C_{d_0} - C_{d_0,D}C_D^{-1}C'_{d_0,D}}(CHOL(C_2)' \otimes I_M)$$

and it will be used in the simulation process. For output prediction at a given d_0, $R = 19$ simulated values of $\tilde{T}^Y$ are drawn from this posterior multivariate normal distribution. Each of these simulations has been reshaped in a $2 \times M$ array corresponding to the two local truncation error time series $\tilde{T}^{1,Y}$ and $\tilde{T}^{2,Y}$, corresponding to equations (2.6a) and (2.6b) respectively. Finally, the relationship (2.6) is used to obtain R simulations for $\tilde{Y}^1_{d,t_{i+1}}$ and $\tilde{Y}^2_{d,t_{i+1}}$:

(2.7a) $\tilde{Y}^1_{d,t_{i+1}} = \tilde{Y}^1_{d,t_i} + h_c(\frac{r_1}{K_1}\tilde{Y}^1_{d,t_i}(K_1 - \tilde{Y}^1_{d,t_i}) - \frac{r_1\alpha_{12}}{K_1}\tilde{Y}^1_{d,t_i}\tilde{Y}^2_{d,t_i}) + \tilde{T}^{1,Y}_{d,t_{i+1}}$

(2.7b) $\tilde{Y}^2_{d,t_{i+1}} = \tilde{Y}^2_{d,t_i} + h_c(\frac{r_2}{K_2}\tilde{Y}^2_{d,t_i}(K_2 - \tilde{Y}^2_{d,t_i}) - \frac{r_2\alpha_{21}}{K_2}\tilde{Y}^2_{d,t_i}\tilde{Y}^1_{d,t_i}) + \tilde{T}^{2,Y}_{d,t_{i+1}}$

for $i = 0, ..., M-1$ and $d = d_0$. It will be shown in the Appendix A that $\tilde{Y}$ at $d = d_0$ converges strongly to the exact solution of (2.1) as h_c converges to 0. The solution of equations (2.7) can be thought of as a coarse-grid Euler solution "corrected" by a statistical prediction of the local truncation error. The median and percentile-based prediction intervals have been used to summarize the prediction.

2.4 Coarse Numerical Solutions Used as Auxiliary Information

In some situations, numerical solutions corresponding to a grid of an intermediary size are available at little computational cost. One can incorporate them as auxiliary information in the statistical model in order to improve it. Moreover, there is more than one way one can make use of this information. A direct way is to use these values as covariates in a regression-type model. This is in the spirit of Kennedy and O'Hagan (2000), who used low-level accuracy computer output data as auxiliary information in order to predict high-level accuracy computer output.

We consider the use of auxiliary information in the form of a numerical solution on a grid of M_C time intervals. Here M_C is a positive integer such that $M_f > M_C > M_c$ and $h_C = \frac{L}{M_C}$. That is, the grid to be used in calculating the auxiliary information is of resolution between that of "fine" and "coarse" grids discussed above. The closer is M_C to M_c, the less computationally expensive (but also less accurate) is the numerical solution to (2.3). In our numerical results we will use $M_C = 2M_c = 30$. Denote by $X^i_{d,t}$ the array of output corresponding to h_C, for species $i = 1, 2$, input vector d (i.e. pair of initial conditions) and time point t, again at time points corresponding to only the coarsest sub-grid $M_c = 15$. A modeling strategy generalizing the approach presented in section 2.3.1. will be presented first. The prior assumption is that $\underline{Y} = N(\mu_0 \underline{1} + \mu_1 \underline{X}, \sigma^2 \Sigma)$, with $\Sigma = C_D \otimes (C_2 \otimes C_T)$, where

$$C_D(d_i, d_j) = \exp(-\theta_{D_1}(d_i(1) - d_j(1))^2 - \theta_{D_2}(d_i(2) - d_j(2))^2), i, j = 0, 1, ..., D,$$

$$C_T(t_i, t_j) = \exp(-\theta_T(t_i - t_j)^2), i, j = 1, ..., M$$

and

$$C_2 = \begin{pmatrix} 1 & p \\ p & 1 \end{pmatrix}.$$

The regression parameters and variance estimates are obtained analytically as

$$[\hat{\mu}_0, \hat{\mu}_1] = ([\underline{1}, \underline{X}]'[\Sigma]^{-1}[\underline{1}, \underline{X}])^{-1}([\underline{1}, \underline{X}]'[\Sigma]^{-1}\underline{Y})$$

and

$$\hat{\sigma}^2 = \frac{1}{2MD}(\underline{Y} - \mu_0\underline{1} - \mu_1\underline{X})'[\Sigma]^{-1}(\underline{Y} - \mu_0\underline{1} - \mu_1\underline{X}),$$

whereas the rest of the estimates are obtained by numerical optimization of the likelihood. The numerical values of these estimates are: $\hat{\mu}_0 = 0.0056$, $\hat{\mu}_1 = 1.0135$, $\hat{\sigma}^2 = 0.0068$, $\hat{\theta}_T = 46.0022, \hat{\theta}_{D_1} = 12.6430, \hat{\theta}_{D_2} = 8.4192, \hat{p} = -0.0867$. Denote by $\underline{X}_{d_0}$ the vector of auxiliary data at a new site d_0 (restricted to the coarse grid only). Then the posterior mean vector is

$$\underline{\hat{Y}}_{d_0} = \mu_0\underline{1} + \mu_1\underline{X}_{d_0} + \Sigma^+_{d_0,D}[\Sigma]^{-1}(\underline{Y} - \mu_0\underline{1} - \mu_1\underline{X}) =$$

$$\mu_0\underline{1} + \mu_1\underline{X}_{d_0} + [(C_{d_0,D}C_D^{-1}) \otimes I_{2M}](\underline{Y} - \mu_0\underline{1} - \mu_1\underline{X}).$$

The marginal posterior variances are equal to $\sigma^2(C_{d_0} - C_{d_0,D}C_D^{-1}C'_{d_0,D})$.

Apart from this direct way of using the auxiliary information, there is another possibility which follows the lines of section 2.3.2. Denote by $T^{j,X}_{d,t}$ $(j = 1, 2)$ the coarse numerical truncation errors obtained as in (2.6) for the numerical solution X and $\underline{T}^X$ denote their vectorized form. We shall assume again that

$$\underline{T}^Y = N(\mu_0\underline{1} + \mu_1\underline{T}^X, \tau^2(C_D \otimes (C_2 \otimes C_T))),$$

with

$$C_D(d_i, d_j) = \exp(-\theta_{D_1}(d_i(1) - d_j(1))^2 - \theta_{D_2}(d_i(2) - d_j(2))^2), i, j = 1, ..., D,$$

$$C_2 = \begin{pmatrix} 1 & p \\ p & 1 \end{pmatrix}$$

and $C_T = I_M$ (time-independence). The MLEs of the parameters involved are: $\hat{\theta}_{D_1} = 9.9764, \hat{\theta}_{D_2} = 8.2074, \hat{p} = -0.1368$, $\hat{\tau}^2 = 0.0005$, $\hat{\mu}_0 = -0.0008$ and $\hat{\mu}_1 = 1.2694$. Similarly, the regression parameters and variance estimates were expressed analytically and the

rest of the estimates are obtained by numerical optimization. If $\underline{T}^X_{d_0}$ denotes the vector of coarse numerical truncation errors at a new testing site d_0, then the posterior vector mean is

$$\hat{\underline{T}}^Y_{d_0} = \mu_0\underline{1} + \mu_1\underline{T}^X_{d_0} + [(C_{d_0,D}C_D^{-1}) \otimes I_{2M}](\underline{T}^Y - \mu_0\underline{1} - \mu_1\underline{T}^X)$$

and the posterior covariance is $\tau^2(C_{d_0} - C_{d_0,D}C_D^{-1}C'_{d_0,D})(C_2 \otimes I_M)$. Next, R simulations are obtained from this posterior multivariate normal distribution, and each of them is reshaped in a $2 \times M$ array corresponding to the two numerical local truncation error time series $\tilde{T}^{1,Y}$ and $\tilde{T}^{2,Y}$. As in section 2.3.2, the relationship (2.7) is used to obtain R simulations $\tilde{Y}$ for the numerical solutions. Again, the median and percentile-based prediction intervals are used to summarize the prediction. It is proved in the Appendix A that $\tilde{Y}$ at $d = d_0$ converges almost surely to the exact solution of (2.1) as h_c converges to zero. This ensures good asymptotic properties of the statistical model.

2.5 Results

In addition to the $D = 20$ design points, Figure 2.1d displays a set of $P = 100$ prediction (test) sites. This test set was also selected according to a maximin criterion from a list of 1,000,000 random LHDs. The minimum inter-point distance for this set is $\underline{d}(P) = 0.0030$, and the index of this testing set was $J(P) = 4$. The points are widely spread and ensure a reasonably good coverage of the experimental region.

Recall that the time interval of study is $[0, L] = [0, 3]$. Data have been saved only at $M_c = 15$ equally spaced points $t = t_1, ..., t_{15}$. Figures 2.2a-b present species size $N_1(t_8)$ in the middle of the time interval at t_8, point predictions and 90% prediction intervals across the 100 test sites for both approaches presented in section 2.3.1 (direct) and section 2.3.2 (local truncation error based, abbreviated TER in the sequel) with $R = 19$ simulations for the later method. Similarly, Figures 2.2c-d present species size $N_2(t_8)$, point predictions and 90% prediction intervals across the $P = 100$ test sites at time t_8 for both approaches. One can notice a better actual probability coverage for the TER method. Moreover, the direct method predicts species size values outside the interval $(0, 2)$ which is not in agreement with the mathematical model. An even sharper distinction between the two methods can be noticed in Figure 2.3, which shows

plots analogous to those in Figure 2.2, but at the end of the time interval, t_{15}. TER method seems to better represent the mathematical properties of the phenomenon. For example, TER correctly predicts that at time t_{15} one of the species has almost won the competition for most of the testing sites. The direct method presented in section 2.3.1 predicts confidently that at time t_{15} the species size functions can take any value in $(0, 2)$ and beyond!

Figures 2.4 and 2.5 show plots similar to those in Figures 2.2 and 2.3, but here auxiliary information has been incorporated into the statistical models according to the methodology presented in section 2.4. Auxiliary information has been especially useful for the direct method since the coarser approximation correctly reflects the general shape of the species size time series. For example, Figure 2.5 shows that the direct method successfully predicted the correct winner of the competition at t_{15}. However, the auxiliary information has not been as helpful in predicting the species sizes in the middle of the time interval at t_8, as it is shown in Figure 2.4. Also notice in Figure 2.4 that the prediction intervals are unrealistically narrow for the direct method in the cases where the competition has not been resolved by t_8. This is due to the stationary nature of the statistical model used in the direct method.

Figure 2.6 shows, for each of the $M_c = 15$ time points, the proportion of $P = 100$ nominal 90% prediction intervals that actually include their corresponding fine-scale solution (or "coverage"). One can notice that the direct method has low coverage of the true values in the second half of the time interval, whereas the TER method has a more realistic coverage with an overall mean coverage closer to the 90% target. When auxiliary information has been used, the direct method improved toward the end of the time interval, but mean coverage at intermediate times is only about 40% and the overall mean coverage is about 79%, still far from the target. We should also note that the uncertainty associated with parameter estimates is not accounted for in this study. Rather, the parameters have simply been replaced by their MLE values in the predictions. (As an aside, notice that the horizontal axis of the plots in Figures 2.6 through 2.11 represents the index of the set of $M_c = 15$ time points. While the time interval of study is continuous, here we want to emphasize the discrete nature of the recorded output data.)

Table 2.1 RMSE measures

N_1	Direct (NAI)	TER (NAI)	Direct (AI)	TER (AI)
RMSE	0.2738	0.1069	0.0610	0.0347
SRMSE	0.2851	0.1163	0.0634	0.0574
N_2	Direct (NAI)	TER (NAI)	Direct (AI)	TER (AI)
RMSE	0.2757	0.1342	0.0543	0.0509
SRMSE	0.2869	0.1355	0.0570	0.0625

Table 2.1 shows a summary of the overall Euclidean distance between the true values and predicted values at each testing site d and time point t when no auxiliary information (NAI) was available and when auxiliary information (AI) has been used. This distance based measure is sometimes called root mean square error (RMSE) and is defined as

$$RMSE = \sqrt{\frac{1}{M_c P}\sum_{d=1}^{P}\sum_{t=1}^{M_c}(Y_{d,t} - \hat{Y}_{d,t})^2}.$$

Table 2.1 also shows the statistical root mean square error (SRMSE), which incorporates the prediction error. For the direct method, this is defined as

$$SRMSE = \sqrt{\frac{1}{M_c P}\sum_{d=1}^{P}\sum_{t=1}^{M_c}[(Y_{d,t} - \hat{Y}_{d,t})^2 + \sigma_d^2]},$$

where σ_d^2 is the posterior variance at each testing site. For the TER method a simulation-based SRMSE has been used, which is defined as

$$SRMSE = \sqrt{\frac{1}{M_c P}\sum_{d=1}^{P}\sum_{t=1}^{M_c}[\frac{1}{R}\sum_{i=1}^{R}(Y_{d,t} - \tilde{Y}_{d,t,i})^2]},$$

with R the number of simulations ($R = 19$ here). When comparing the Direct and TER methods, TER does a better job when no auxiliary information has been used. When auxiliary information was available, RMSE statistics for TER are still smaller, although the SRMSE for N_2 is larger for the TER than for the direct method. However, the SRMSE measure for the direct method incorporates posterior variances which are unrealistically small, as the plot of actual probability mean coverages in Figure 2.6c shows. Thus, the direct method SRMSE for N_2 is an overly optimistic measure. While the coarse grid output data used as auxiliary information had a bigger impact on the direct method, it is worth noting that reliable coarse

data are not inexpensive for all applications (see next chapter) and therefore one should balance the cost versus improvement of the results.

Figures 2.7-2.10 show time series for prediction and true values at four different test sites among the $P = 100$. They could be considered qualitatively representative for all $P = 100$ sites. The $D = 20$ design sites plotted in Figure 2.1d show that few correspond to cases where $N_1(0) \approx N_2(0)$; most of the design sites correspond to situations where there is a clear winner of the competition during the time interval considered. When $N_1(0) \approx N_2(0)$, there will eventually be a winner, but not during the time interval considered here. This region of the input space could be challenging for the statistical predictors considered here because the available information from the training set of design points might not correctly reflect the characteristics of species size functions. Therefore special attention will be paid to this region. Figure 2.7 corresponds to the testing site $A = (0.0463, 0.0760)$ which is inside the experimental region and reasonably far from $N_1(0) \approx N_2(0)$. It shows a situation in which both the direct and TER methods predict correctly, with or without auxiliary information being used. Figure 2.8 corresponds to the testing site $A = (0.0655, 0.0942)$ which is near the boundary of the experimental region, but far from $N_1(0) \approx N_2(0)$. The direct method without auxiliary information predicts negative values for the species size N_1 and values beyond the species saturation level (equal to 2 here) for the species size N_2. This clearly violates the restrictions of the mathematical model considered here. However, the quality of the predictions is greatly improved for the direct method when auxiliary information is used. The TER method accurately predicts the true values, with or without auxiliary information. Figure 2.9 corresponds to the testing site $A = (0.0731, 0.0693)$ which is inside the experimental region and close to $N_1(0) \approx N_2(0)$. The direct method without auxiliary information tends to predict a no-winner situation (which is the main diagonal's property), although the pair of initial values considered is not on the main diagonal and species N_1 wins the competition at about t_{15}. This error at t_{15} is fixed, however, when auxiliary information is available, although the second half of the time interval is still badly covered by the 90% prediction intervals. The TER method predicts the true values accurately although the prediction bounds are wider in

the second half of the time interval, reflecting a greater uncertainty about the dynamics of the species size functions. Finally, Figure 2.10 corresponds to the testing site $A = (0.0069, 0.0108)$ which is both near the boundary of the experimental region and close to $N_1(0) \approx N_2(0)$. The direct method without auxiliary information confidently predicts the wrong winner of the competition, that is the first species has a better chance of winning when in fact the second species wins. The TER method is completely unsure about which species will win the competition, as indicated by the prediction interval of maximum logical size at t_{15}, that is the interval $(0, 2)$. This clearly makes the TER prediction interval useless as a predictor for this testing site at t_{15}, but not misleading. Panels $e - h$ show that auxiliary information improves the qualitative performance of both the direct and TER methods. Prediction intervals are incorrect for both methods at intermediate times, but are broader for the TER method which gives a more accurate indication of predictive uncertainty. To see more clearly that the auxiliary information was helpful for the TER method at $A = (0.0069, 0.0108)$, in Figure 2.11 are plotted the prediction summaries shown in Figure 2.10 along with the $R = 19$ simulations. When no auxiliary information has been used, about half of the simulations tend to choose the first species as a winner, while the other half tends to choose the second species. Adding the auxiliary information greatly improves the predictions in the sense that all simulations choose the correct winning species.

2.6 Conclusion

This study showed evidence that code information can be used to improve the fidelity of statistical predictions of model output. The example presented in this study illustrates a methodology exploiting code information that could be useful in situations involving slow-running finite difference codes. This example is simple and has pedagogical value in the sense that it has properties, such as nonlinearity, underlying more complex models, although it is not demanding in terms of computational resources. The fine grid numerical solution takes only about 2 seconds per run in MATLAB on a 1GB RAM, 400MHz 64-bit processor workstation. The TER (AI) method with $R = 19$ takes about the same time as a numerical solution on a

grid with intervals of size $h_c/4$. The RMSE measures for a numerical solution computed on a grid of this size, and restricted to grid of size h_c for comparison, are RMSE(N_1)=0.0456 and RMSE(N_2)=0.0458 which are comparable to those of the TER (AI) method. Additionally, the statistical methods provide prediction bounds, which are especially accurate for the TER method. In general, the accuracy of the TER method and the computational resources that it requires depend on the particular problem studied. While this chapter focused primarily on introducing a statistical method (TER), the next chapter will present a more computationally demanding example where all the statistical simulators are more accurate than numerical solutions requiring comparable computational resources.

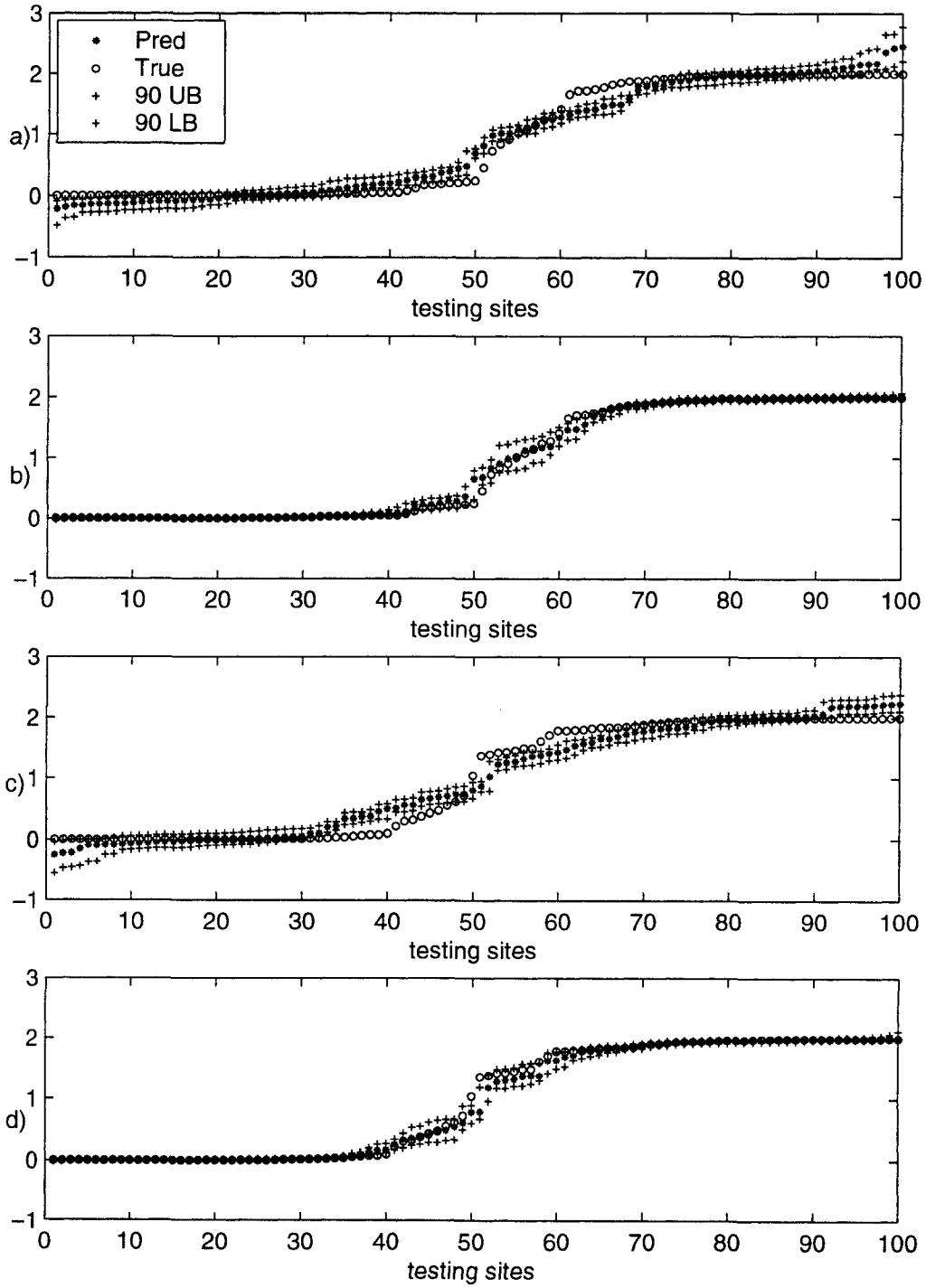

Figure 2.2 Predicted and true values at $t = t_8$, no auxiliary information

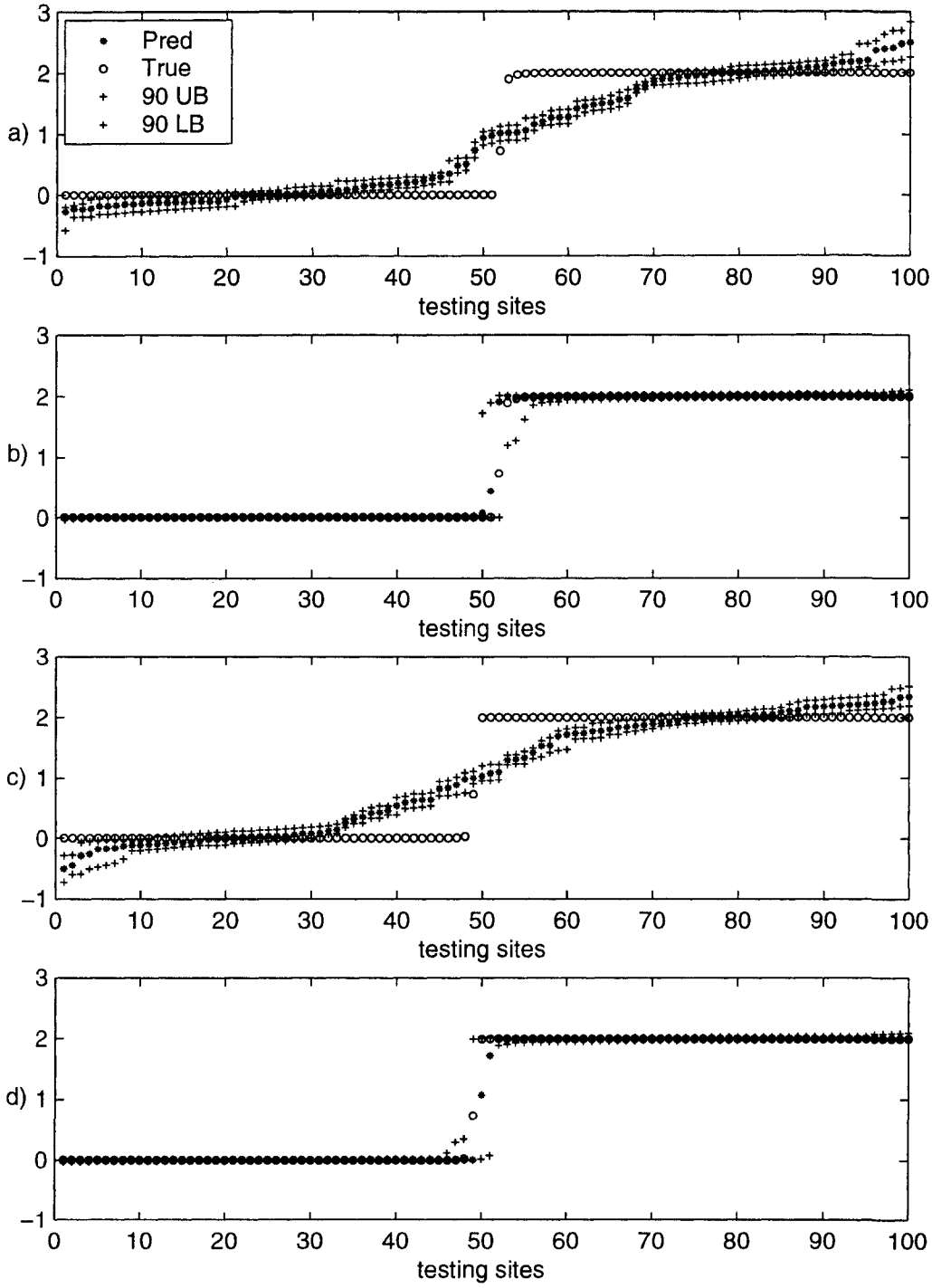

Figure 2.3 Predicted and true values at $t = t_{15}$, no auxiliary information

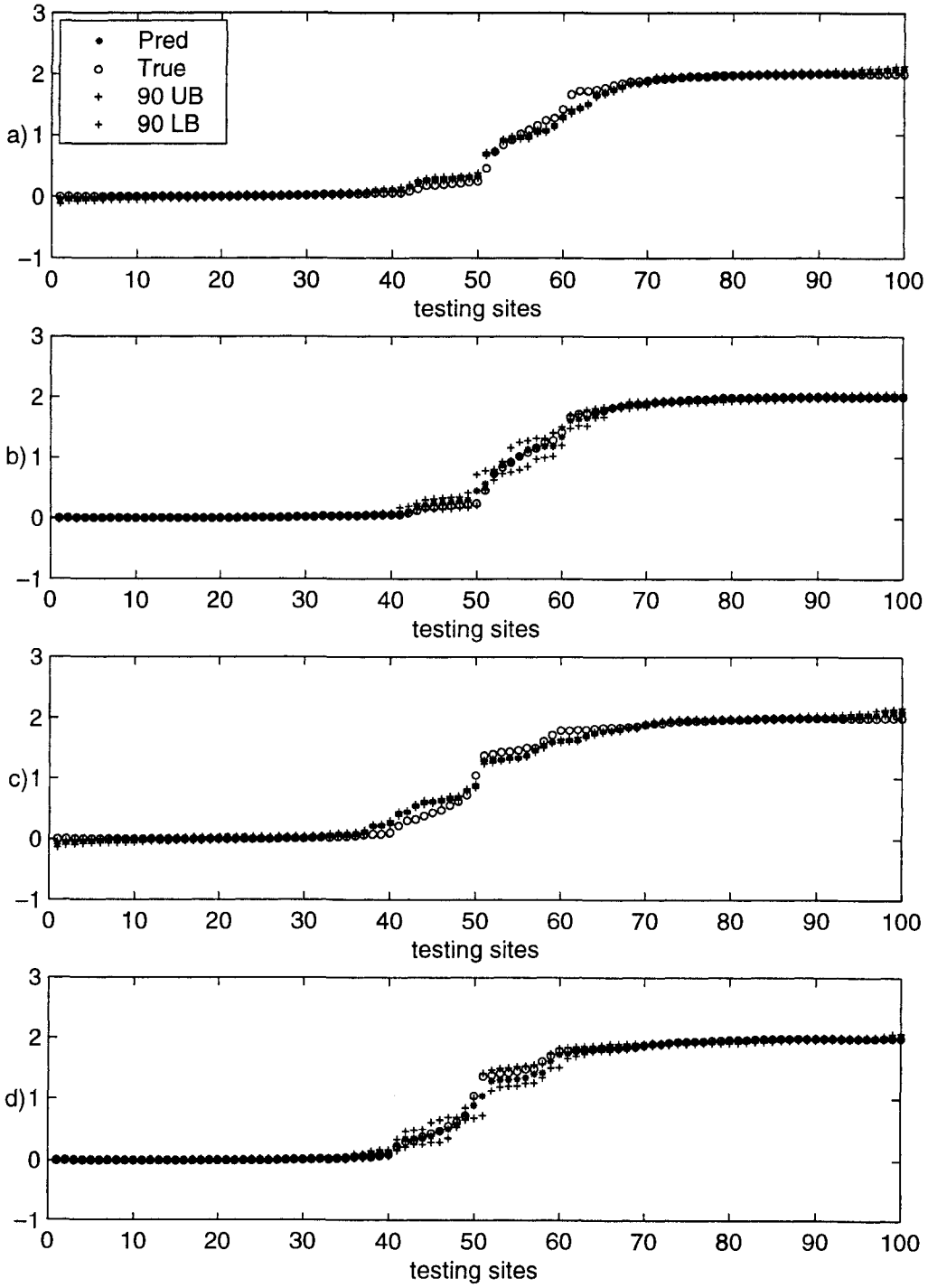

Figure 2.4 Predicted and true values at $t = t_8$, auxiliary information used

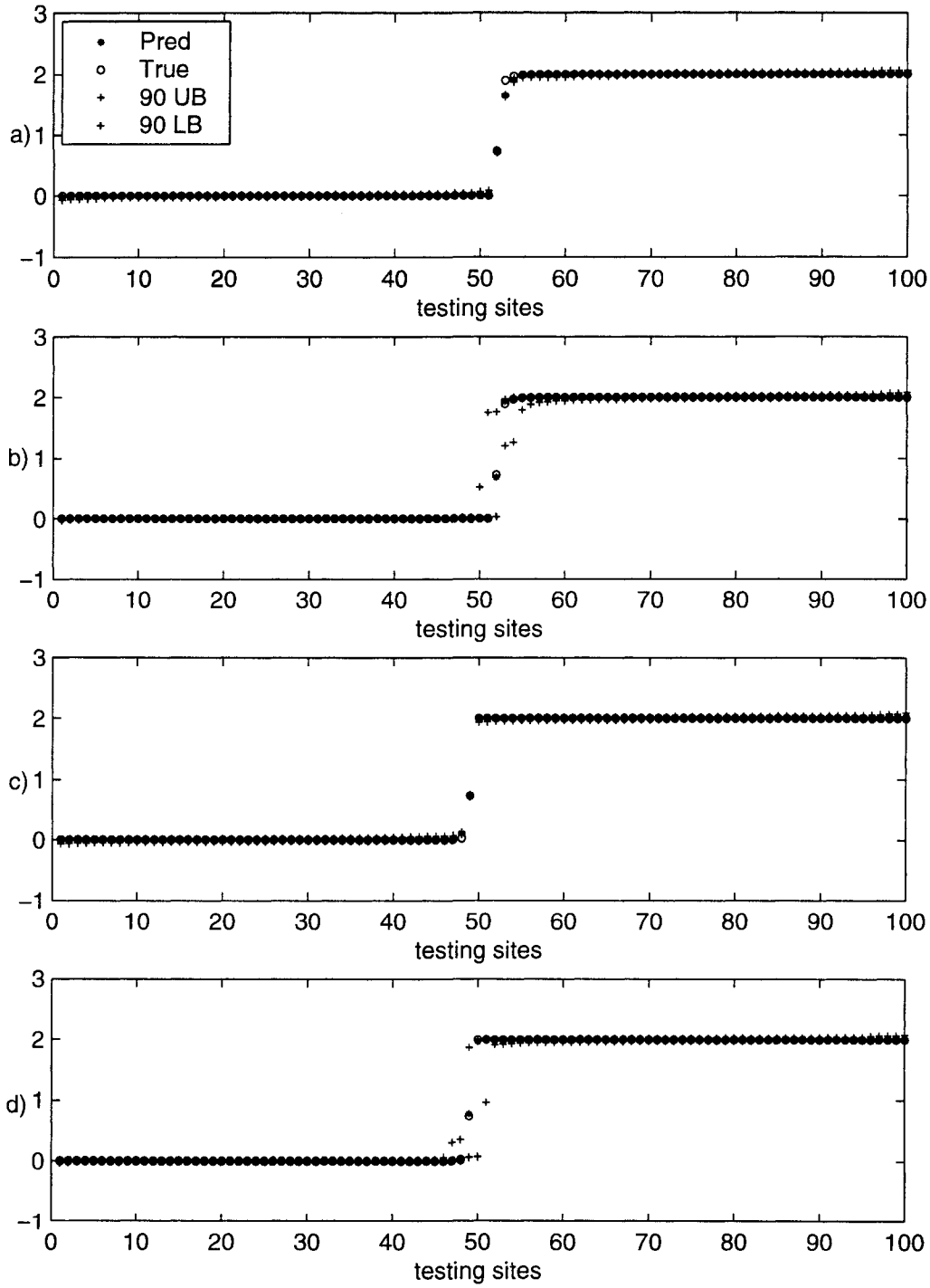

Figure 2.5 Predicted and true values at $t = t_{15}$, auxiliary information used

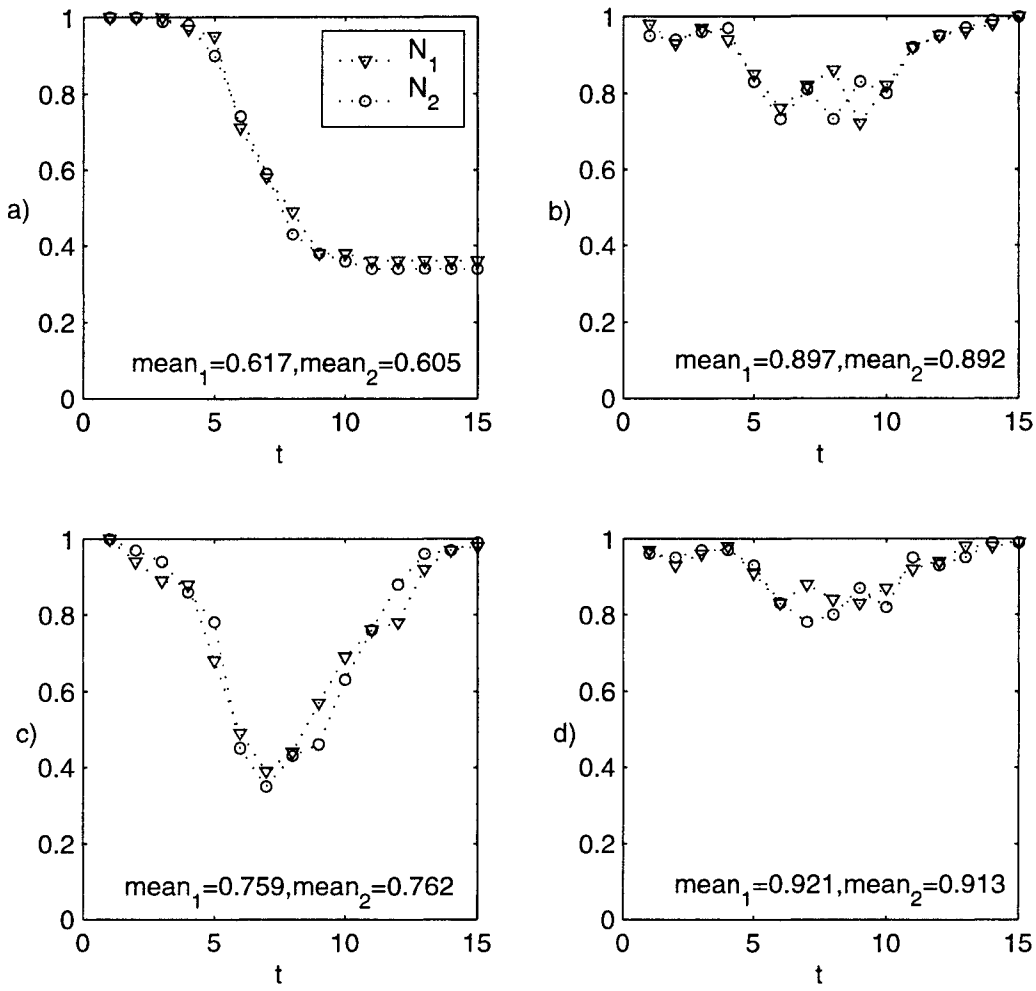

Figure 2.6 Mean coverage for nominal 90% prediction intervals

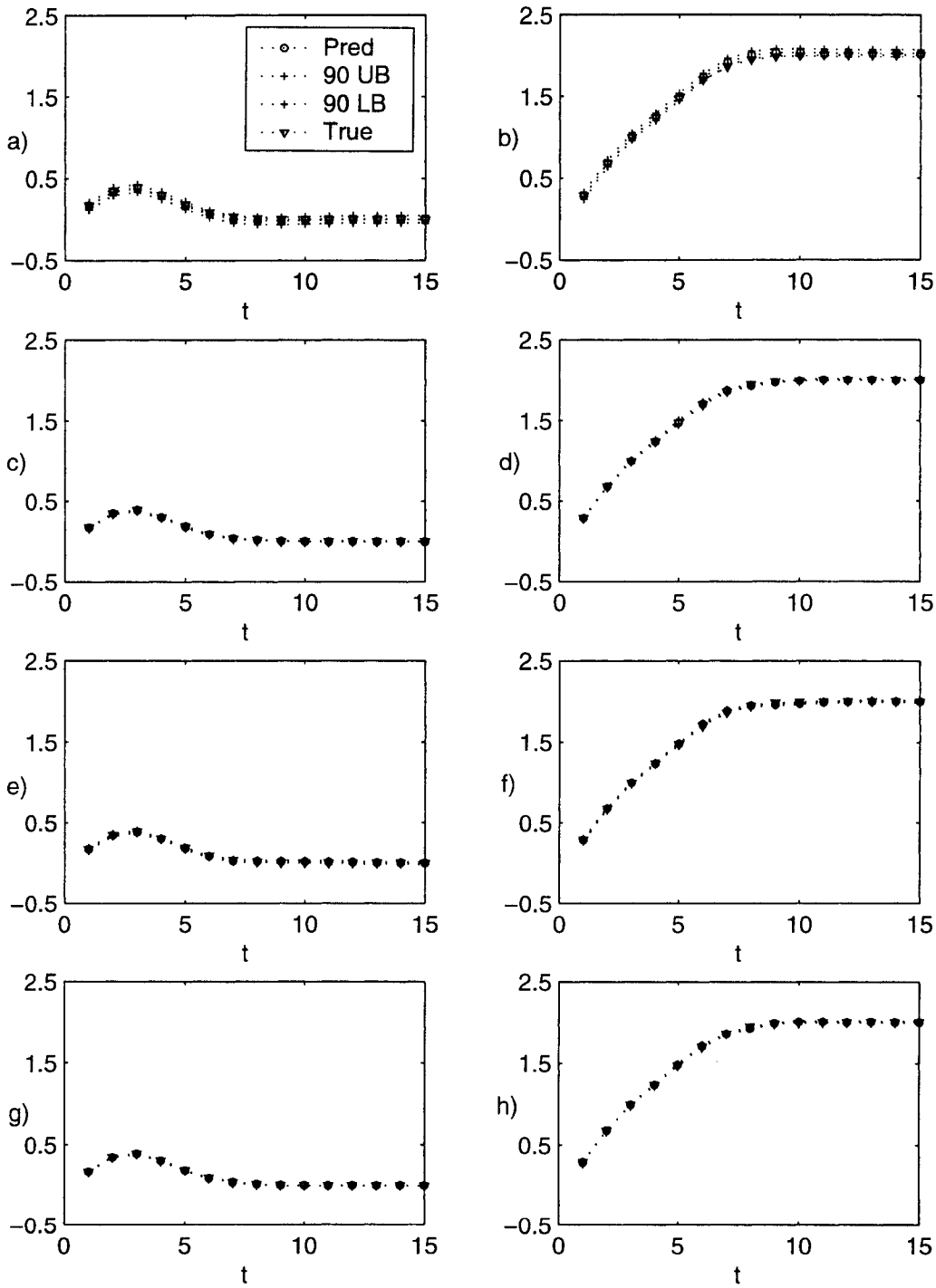

Figure 2.7 Prediction and true values at testing site A=(0.0463,0.0760)

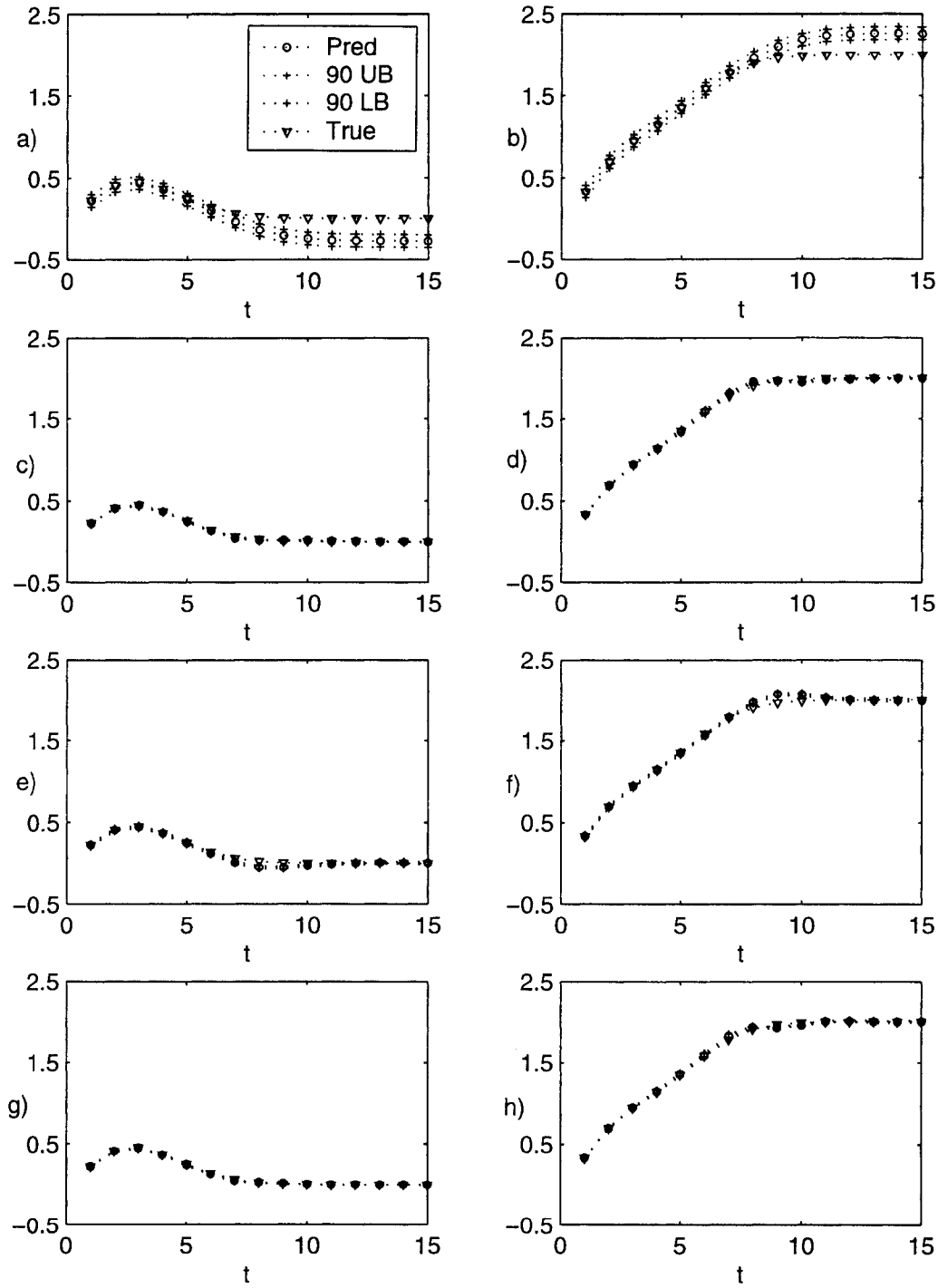

Figure 2.8 Prediction and true values at testing site A=(0.0655,0.0942)

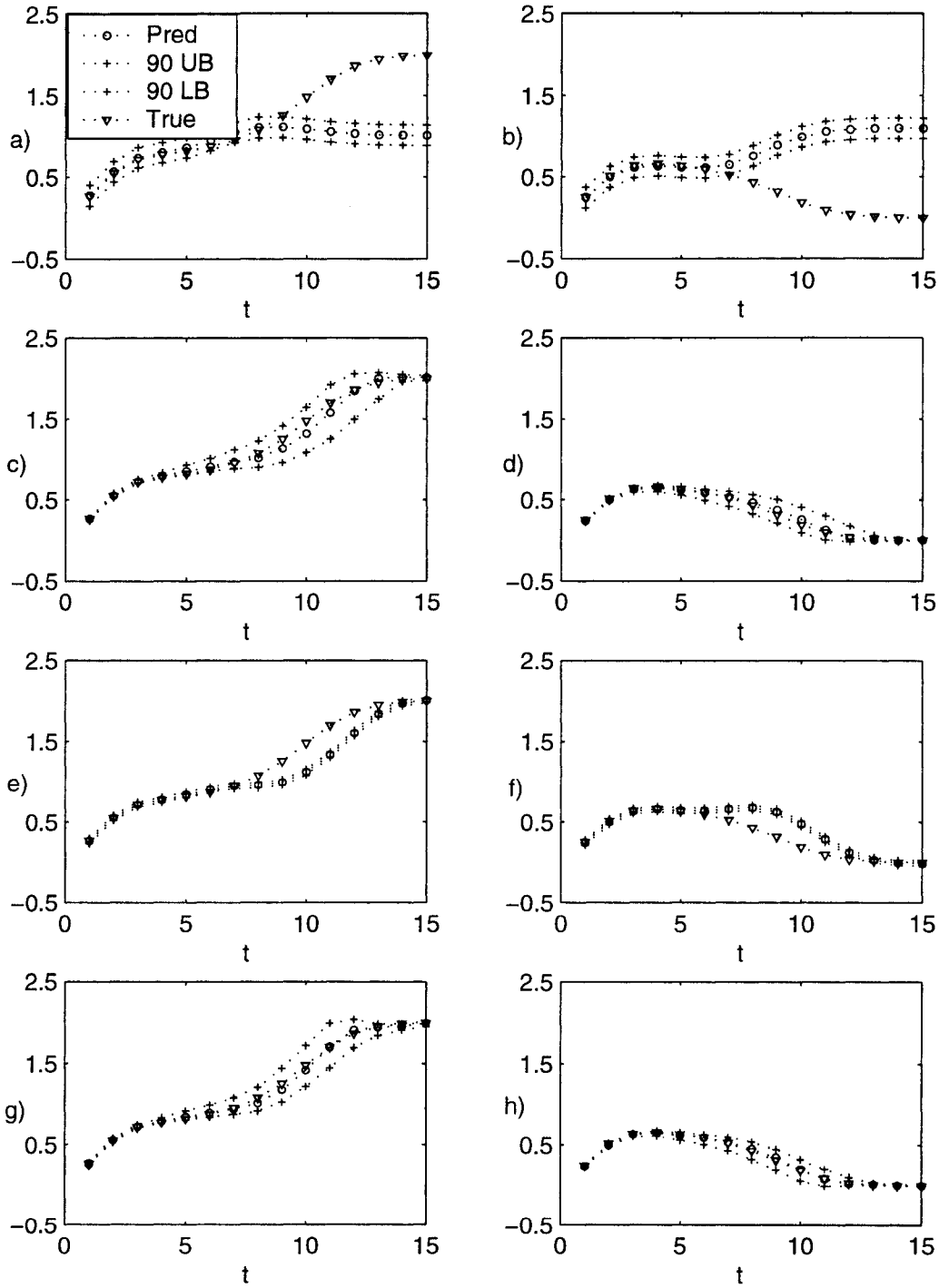

Figure 2.9 Prediction and true values at testing site A=(0.0731,0.0693)

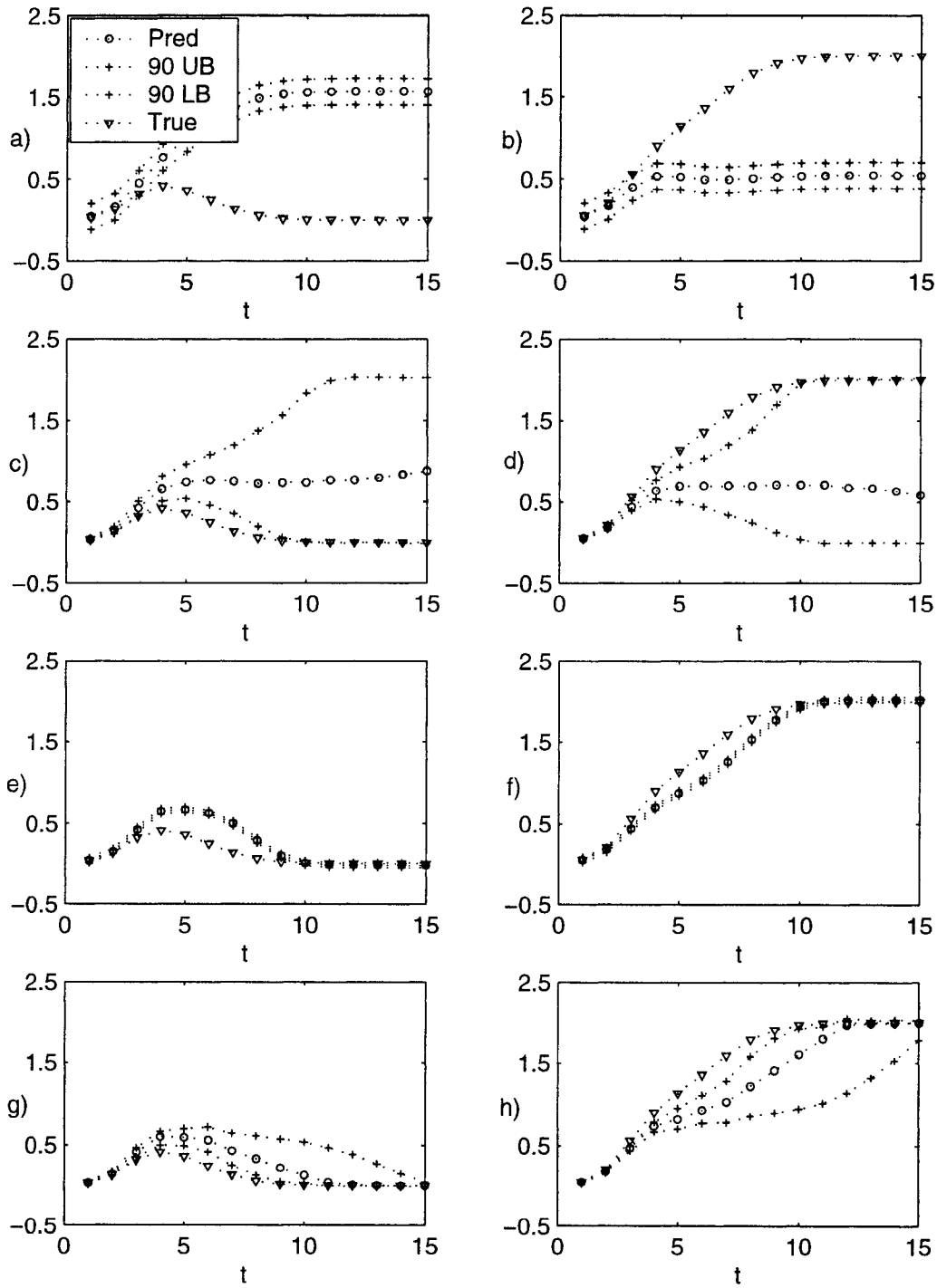

Figure 2.10 Prediction and true values at testing site A=(0.0069,0.0108)

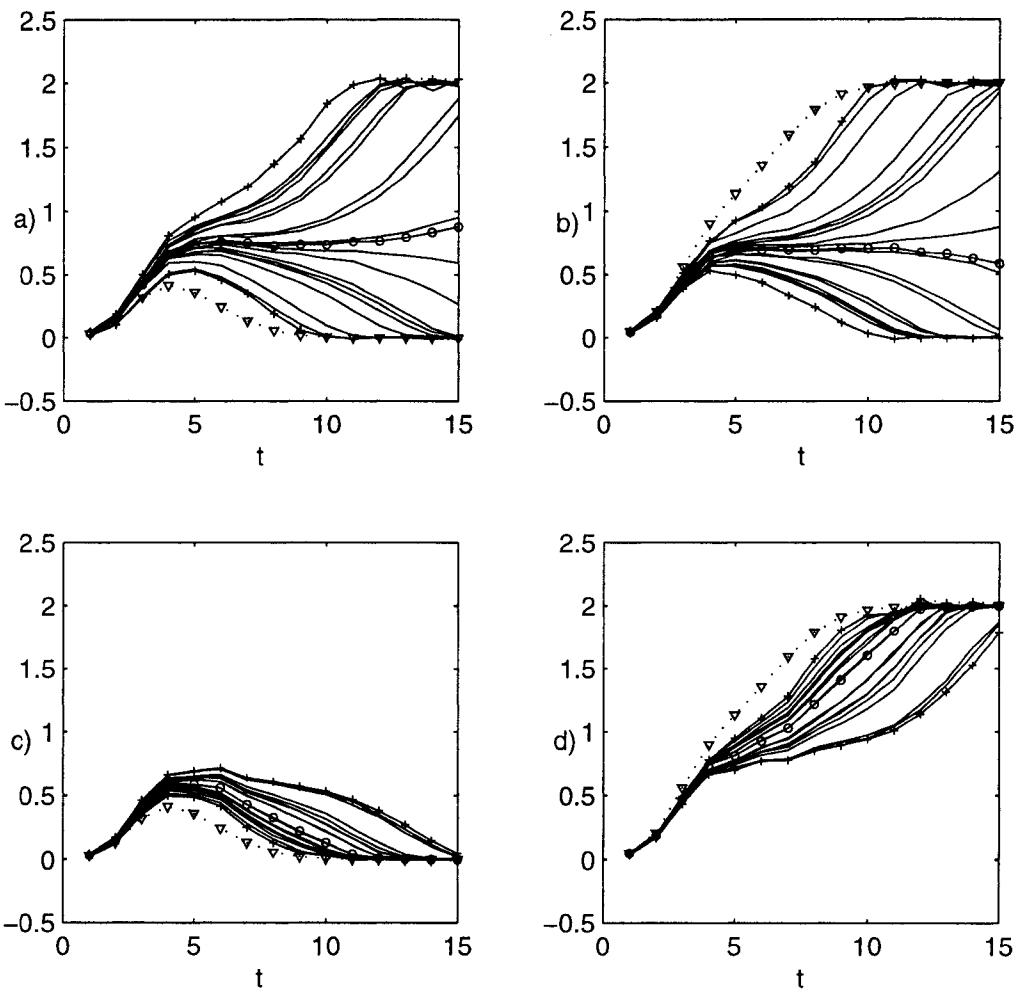

Figure 2.11 Simulations and true values at testing site A=(0.0069,0.0108)

CHAPTER 3. STATISTICAL ANALYSIS OF COMPUTER EXPERIMENTS FOR FINITE DIFFERENCE CODES: LARGE TEMPORAL DIMENSION

Dorin Drignei and Max D. Morris

3.1 Introduction

In chapter 2 we analyzed statistically a computer experiment of a finite difference solver of differential equations. We suggested methods for constructing statistical predictors as surrogates for output of the code at a fine resolution. These methods involve multivariate output and can be considered, in some sense, generalizations of the univariate output methods presented in Sacks et al. (1989) and Currin et al. (1991). The first method was a direct extension of the existing methodology. In a demonstration exercise this method was not especially successful in representing important details of the mathematical model (section 2.3.1), but it improved when intermediary grid resolution numerical solutions were used as auxiliary information in regression type models (section 2.4). We then suggested a second method that exploits computer code information derived from numerical local truncation errors (section 2.3.2) and demonstrated that the resulting statistical predictions can be more accurate surrogates in the exercise. When this method is used, the fine grid output data is saved on a coarser grid with increments small enough for the statistical method based on local truncation errors to work. The output data set is saved and analyzed without significant computational effort because the resulting coarse grid set of points is relatively small. While this approach could be used for a wide class of differential equations models, there are cases where it is not applicable because even a coarse grid, in order to be reliable for our purposes, must have increments that

are small relative to the total time interval of the simulation . This results in a large set of coarse grid points, and saving and manipulating such large amounts of output data becomes a problem. The climate models are important examples for which this situation may occur.

This chapter will discuss the limitations of the methods proposed in chapter 2 and suggest alternative ways to analyze such large data sets. Here we deal with a computer experiment of a finite difference code solver of a partial differential equation (PDE) system, in a situation where the output data set has a large temporal dimension, e.g. a long simulation interval relative to characteristic dynamics of the modeled system. We shall modify the two methods presented in chapter 2 so that these data will be analyzed in two stages and only a fraction of them will be saved, at a small number of time points dividing the temporal space into intervals of equal length. The first stage in both methods proposes analyzing these data by conventional methods developed for the statistical analysis of computer experiments. The second stage in the first method consists of modeling the output data at the intermediary time points by discrete-time Brownian bridge (DBB) processes, which are autoregressive processes constrained to pass through the data saved at the equally spaced time points above. This is a fast method but the predictor has the shape of a linear interpolator and is therefore not of high quality. To overcome this weakness, the second stage in the second method consists of modeling the numerical local truncation errors at the intermediary time points by DBB processes. This method is slower, but the predictors of the numerical solutions are more curvilinear and of higher quality. Intermediate grid resolution output data can be used as auxiliary information for both methods. This option, however, adds a non-negligible amount of computational effort into the statistical prediction scheme. We will show that all the statistical predictors presented for this example are more accurate than the numerical solutions requiring comparable computational time.

3.2 Output data

To demonstrate methods, we shall use output data from a numerical solver of PDEs describing a two-dimensional model of a thin fluid layer such as shallow water. These equations

are (Gill 1982, but also Jiang et al. 1995, Poje et al. 1996, Jones et al. 1997):

(3.1a) $\frac{\partial h}{\partial t} + \frac{\partial (uh)}{\partial x} + \frac{\partial (vh)}{\partial y} = 0$

(3.1b) $\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -g\frac{\partial h}{\partial x} + (f_0 + \beta y)v + \nu\nabla^2 u + F^u$

(3.1c) $\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -g\frac{\partial h}{\partial y} - (f_0 + \beta y)u + \nu\nabla^2 v + F^v$

A brief discussion of (3.1) will follow in order to describe the data and the experimental space. The system of PDEs (3.1) has been used to describe the large scale motions of the oceans surface. According to Jiang et al. (1995), in this model it is assumed that a dynamically active and thin upper layer of water with density ρ lies above an infinitely deep and motionless layer with a different density. Here, $h(x, y, t)$ is the elevation above the bottom of the upper layer of a point situated at the surface of the water with horizontal coordinates (x, y) at time t. The vector $(u(x, y, t), v(x, y, t))$ is the velocity of this surficial point along the two horizontal dimensions. Equation (3.1a) describes mass conservation, and is a statement about the mass remaining constant in a unit volume of a moving fluid. Equations (3.1b) and (3.1c) need to be considered together and represent momentum conservation. Essentially they state that the acceleration of a moving particle is the sum of forces acting per unit mass. Indeed, if $\underline{\omega} = (u, v)$, the left part of (3.1b-c) is the acceleration, the time derivative of velocity, obtained by the chain rule of calculus:

$$\underline{a} = \frac{d}{dt}\underline{\omega} = \frac{\partial \underline{\omega}}{\partial t}\frac{dt}{dt} + \frac{\partial \underline{\omega}}{\partial x}\frac{dx}{dt} + \frac{\partial \underline{\omega}}{\partial y}\frac{dy}{dt} = \frac{\partial \underline{\omega}}{\partial t} + \frac{\partial \underline{\omega}}{\partial x}u + \frac{\partial \underline{\omega}}{\partial y}v = (\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}, \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}).$$

The right hand side of (3.1b-c) is the sum of the acting vector forces such as the pressure force $-g(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y})$, the Coriolis force $((f_0 + \beta y)v, -(f_0 + \beta y)u)$ resulted from the Earth's daily spinning, the fluid viscous force $\nu\nabla^2(u, v) = \nu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2})$ and the wind forcing (F^u, F^v) with $F^v = 0$ and $F^u(y) = -\frac{\tau}{\rho h_0} cos(2\pi\tilde{y}) * (1 - 4\alpha(\tilde{y} - .5))$, where $\tilde{y}$ is the latitude coordinate normalized to the interval $[0, 1]$. These forces are discussed in more detail by Apel (1987). The functional form of the wind forcing is similar to that used by McCalpin (1995) who introduced an asymmetry factor in the expression of the sinusoidal shaped wind forcing appearing in other papers such as Jiang et al. (1995) and Jones et al. (1997). The forces presented above depend on parameters of which some are known accurately and others are known only to be within some ranges. Jiang et al. (1995), Jones et al. (1997) among others

list the known parameters and conduct numerical studies aimed at clarifying the dependence of the PDE solutions on the less well-known parameters. These studies included visual, informal inspection of the output data for various parameter configurations. For example, Jiang et al. (1995) were interested in whether the plots reveal time periodicity of the numerical solutions, whether the contour plots of the function h at various time points are symmetric, etc. For the purpose of this chapter we shall consider that all parameters are accurately known except for three: ν the fluid viscosity, τ the wind stress and α the wind asymmetry. In order for the system (3.1) to have a unique solution (h, u, v) one needs to specify initial and boundary conditions. The PDE system (3.1) is intended to describe the surface motion of a mid-latitude ocean such as the North Atlantic and it is specified on a rectangular space domain Ω=900 km $\times$ 1,800 km. This system is written in Cartesian coordinates here, although it can also be written in spherical coordinates which is a more realistic but more complicated form. At the initial time $t = 0$ the fluid is considered at rest, that is $\underline{\omega}(x, y) = 0$ for any (x, y) in Ω. The water surface has elevation $h(x, y) = h_0 = constant$ at $t = 0$ for any (x, y) in Ω. For the boundary conditions, the water is considered at rest on the boundary of Ω and the water surface has elevation $h \equiv h_0$ on the boundary of Ω at any time t. The initial elevation has been set at $h_0 = 500$ m, as in Jiang et al. (1995). The PDE system (3.1) runs on a finite time interval $[0, T]$. In fact, physically realistic solutions are considered only on an interval $[T_0, T]$, $T_0 > 0$, since the model undergoes a spin-up (or burn-in) period $[0, T_0]$. Depending on applications, T_0 could be months or years and T could be years, decades or even centuries (on high-performance computers). Due to limited computational resources, here we shall consider approximately T_0=0.5 years and T=1.5 years.

The PDE system (3.1) is nonlinear and must be solved numerically since, in general, no analytical solution is available. One common approach is to approximate the continuous time PDE solutions on a discrete grid embedded in $[T_0, T] \times \Omega$, by approximating the derivatives appearing in (3.1) with scaled finite differences. Since there is no unique way to approximate these derivatives, there is a variety of finite difference schemes for the same problem. Here we have chosen a finite difference scheme described in Poje et al. (1996) in which derivatives are

approximated over three contiguous time values:

(3.2a) $\frac{h_{i,j}^{n+1}-h_{i,j}^{n-1}}{2\Delta t} = -(\frac{\delta_x \bar{u}_{i,j}^n h_{i,j}^n}{2\Delta x} + \frac{\delta_y \bar{v}_{i,j}^n h_{i,j}^n}{2\Delta y})$

(3.2b) $\frac{u_{i,j}^{n+1}-u_{i,j}^{n-1}}{2\Delta t} = -(u_{i,j}^n \frac{\delta_x u_{i,j}^n}{2\Delta x} + v_{i,j}^n \frac{\delta_y u_{i,j}^n}{2\Delta y}) - g\frac{\delta_x \bar{h}_{i,j}^n}{2\Delta x} + (f_0 + \beta y_j)v_{i,j}^n + \nu(\frac{\delta_x^2}{\Delta x^2} + \frac{\delta_y^2}{\Delta y^2})u_{i,j}^n + F_{i,j}^u$

(3.2c) $\frac{v_{i,j}^{n+1}-v_{i,j}^{n-1}}{2\Delta t} = -(u_{i,j}^n \frac{\delta_x v_{i,j}^n}{2\Delta x} + v_{i,j}^n \frac{\delta_y v_{i,j}^n}{2\Delta y}) - g\frac{\delta_y \bar{h}_{i,j}^n}{2\Delta y} - (f_0 + \beta y_j)u_{i,j}^n + \nu(\frac{\delta_x^2}{\Delta x^2} + \frac{\delta_y^2}{\Delta y^2})v_{i,j}^n.$

Notation used in these equations is defined as

$$\bar{h}_{i,j} = h_{i+1,j+1} + h_{i+1,j} - h_{i,j+1} - h_{i,j}, \bar{u}_{i,j}^n = \frac{u_{i,j}^n + u_{i,j-1}^n}{2}, \bar{v}_{i,j}^n = \frac{v_{i,j}^n + v_{i-1,j}^n}{2}, \delta_x u_{i,j}^n = u_{i+1,j}^n - u_{i-1,j}^n,$$

$$\delta_x^2 u_{i,j}^n = u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n, \delta_y u_{i,j}^n = u_{i,j+1}^n - u_{i,j-1}^n, \delta_y^2 u_{i,j}^n = u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n.$$

The approximation scheme (3.2) can be rewritten as:

(3.3a) $h_{i,j}^{n+1} = h_{i,j}^{n-1} + 2\Delta t[-(\frac{\delta_x \bar{u}_{i,j}^n h_{i,j}^n}{2\Delta x} + \frac{\delta_y \bar{v}_{i,j}^n h_{i,j}^n}{2\Delta y})]$

(3.3b) $u_{i,j}^{n+1} = u_{i,j}^{n-1} + 2\Delta t[-(u_{i,j}^n \frac{\delta_x u_{i,j}^n}{2\Delta x} + v_{i,j}^n \frac{\delta_y u_{i,j}^n}{2\Delta y}) - g\frac{\delta_x h_{i,j}^n}{2\Delta x} + (f_0 + \beta y_j)v_{i,j}^n + \nu(\frac{\delta_x^2}{\Delta x^2} + \frac{\delta_y^2}{\Delta y^2})u_{i,j}^n + F_{i,j}^u]$

(3.3c) $v_{i,j}^{n+1} = v_{i,j}^{n-1} + 2\Delta t[-(u_{i,j}^n \frac{\delta_x v_{i,j}^n}{2\Delta x} + v_{i,j}^n \frac{\delta_y v_{i,j}^n}{2\Delta y}) - g\frac{\delta_y h_{i,j}^n}{2\Delta y} - (f_0 + \beta y_j)u_{i,j}^n + \nu(\frac{\delta_x^2}{\Delta x^2} + \frac{\delta_y^2}{\Delta y^2})v_{i,j}^n]$

The scheme (3.3) iteratively computes the approximate solutions starting with the initial values and inherently introduces an approximation error at each time step. For this error to be small enough to prevent unacceptable accumulation, the increments Δt, Δx and Δy, which appear in the approximation of derivatives, need to be small. From the papers mentioned above and from our own assessment it appears that triplets of grid sizes such as $(\Delta t, \Delta x, \Delta y) = (20min, 20km, 20km)$ yield good numerical solutions. In fact, in our analyses we shall consider a wider range of grid size triplets, as will be described below. Similar wide ranges of grid size triplets have been considered in Poje et al. (1996) and Jones et al. (1997) in their studies. It is important to point out that the finer the grid, the more accurate the numerical solutions but also the more computationally intensive the solution will be.

Next we introduce the notion of numerical local truncation errors which will play an important role in the development of the statistical methodology. Let $(\Delta t_c, \Delta x_c, \Delta y_c)$ be a triplet grid size and $(\Delta t_1, \Delta x_1, \Delta y_1)$ be another, finer triplet grid size. Here we require $\Delta t_1 = 2^{-k_1}\Delta t_c$, $\Delta x_1 = 2^{-k_2}\Delta x_0$, $\Delta y_1 = 2^{-k_3}\Delta y_c$ for some non-negative integers k_1, k_2, k_3 (at least one must be positive) so that the points on the original coarse grid form a subset of those on any finer grid. If h, u and v are the solution of (3.3) with $(\Delta t, \Delta x, \Delta y) = (\Delta t_1, \Delta x_1, \Delta y_1)$ and i, j, n are indices corresponding to the coarser grid then

(3.4a) $E(h)_{i,j}^{n-1} = h_{i,j}^{n+1} - \{h_{i,j}^{n-1} + 2\Delta t_c[-(\frac{\delta_x \bar{u}_{i,j}^n h_{i,j}^n}{2\Delta x_c} + \frac{\delta_y \bar{v}_{i,j}^n h_{i,j}^n}{2\Delta y_c})]\}$

(3.4b) $E(u)_{i,j}^{n-1} = u_{i,j}^{n+1} - \{u_{i,j}^{n-1} + 2\Delta t_c[-(u_{i,j}^n \frac{\delta_x u_{i,j}^n}{2\Delta x_c} + v_{i,j}^n \frac{\delta_y u_{i,j}^n}{2\Delta y_c}) - g\frac{\delta_x h_{i,j}^n}{2\Delta x_c} + (f_0 + \beta y_j)v_{i,j}^n + \nu(\frac{\delta_x^2}{\Delta x_c{}^2} + \frac{\delta_y^2}{\Delta y_c{}^2})u_{i,j}^n + F_{i,j}^u]\}$

(3.4c) $E(v)_{i,j}^{n-1} = v_{i,j}^{n+1} - \{v_{i,j}^{n-1} + 2\Delta t_c[-(u_{i,j}^n \frac{\delta_x v_{i,j}^n}{2\Delta x_c} + v_{i,j}^n \frac{\delta_y v_{i,j}^n}{2\Delta y_c}) - g\frac{\delta_y h_{i,j}^n}{2\Delta y_c} - (f_0 + \beta y_j)u_{i,j}^n + \nu(\frac{\delta_x^2}{\Delta x_c{}^2} + \frac{\delta_y^2}{\Delta y_c{}^2})v_{i,j}^n]\}$

are the numerical local truncation errors. Thus, they are obtained by introducing a fine grid numerical solution into a coarser finite difference scheme. Chapter 2 provided more details on these quantities which are in some respects analogous to residuals in a statistical analysis. We shall use them in the next sections to introduce statistical models incorporating code information.

We now turn to discussing the form of the data used in the statistical analyses. The goal of this chapter is to conduct and analyze statistically a computer experiment on the effects of the three unknown parameters ν, τ and α, gathered in a vector $\gamma = (\nu, \tau, \alpha)$. The experiment consists of selecting a small number of three dimensional parameter vectors and computing the corresponding fine grid numerical solution using system (3.3). Using these calculated values as data, the analysis will produce statistical predictors of fine grid numerical solutions at untried parameter vectors. The design region and the ranges of parameters considered are plotted in Figure 3.1. The ranges of these parameters are chosen so that the solutions are numerically stable (given the grid size triplets considered) and physically realistic. A number $D = 20$ of parameter vectors γ were selected according to a maximin latin hypercube design (LHD) within a list of 1,000,000 randomly generated LHDs. This selected set of points provides good coverage of the design region. This design is probably not the globally optimum design within the set of all possible LHDs, however our emphasis here is on analysis rather than experimental design. To see qualitatively how much the behavior of this model varies over the design region, we plotted in Figure 3.2 contour plots of finite difference h, u and v solutions at the end of the experimental time for parameters situated in the corners of the box denoted 'A' and 'B' in Figure 3.1. There are substantial qualitative differences between these solutions, ranging from almost symmetric (corner 'B') to asymmetric, spatially complex dynamics (corner 'A'). The

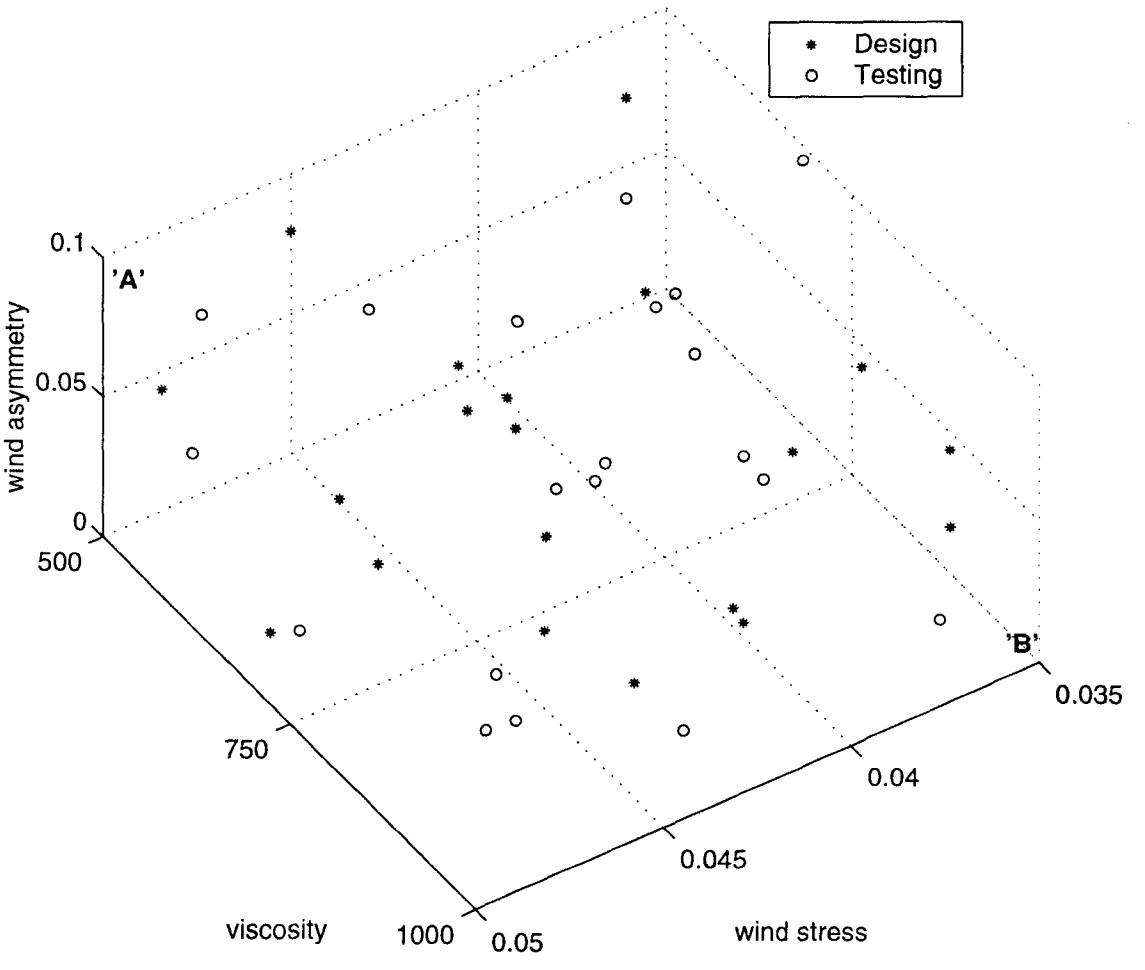

Figure 3.1 Design and testing sites used in the computational experiment

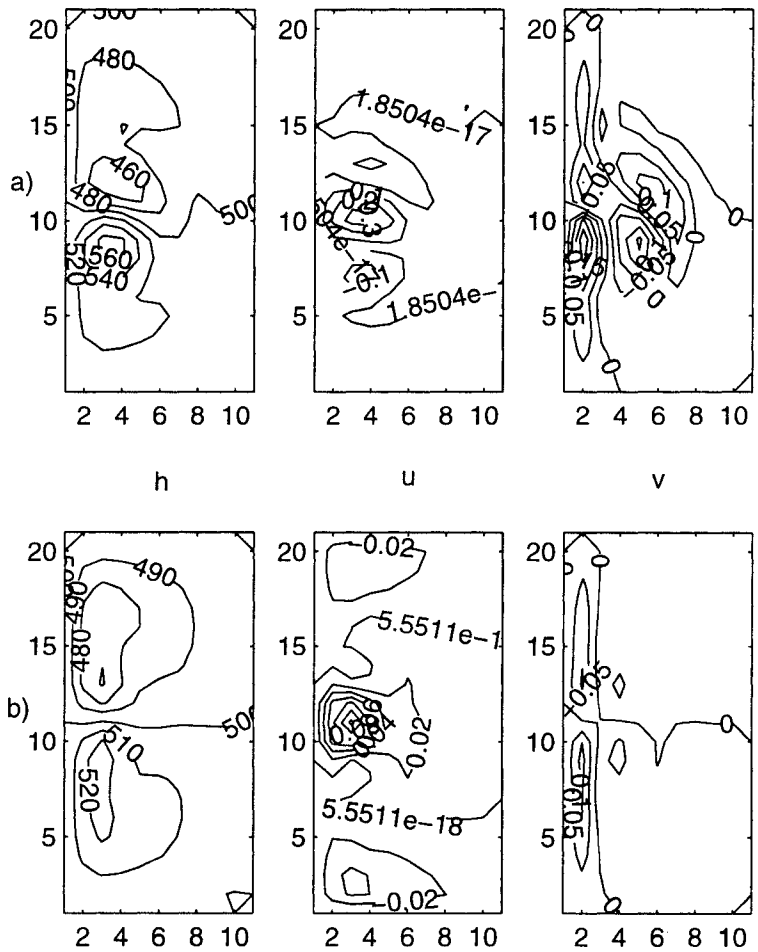

Figure 3.2 Contour plots of the numerical solutions at the final time point

grid on which computed output data will be saved is $(\Delta t_c, \Delta x_c, \Delta y_c) = (50min, 90km, 90km)$. We found by trial and error that numerical solutions corresponding to a grid coarser than this are unbounded, numerically unstable for the PDE parameters considered here. The instability is usually associated with explicit schemes in time, where the solution values at a time step are calculated directly from the values at previous time steps. Scheme (3.2) is an example of an explicit scheme. There are also implicit schemes, where the computation of the solution value at a time step requires solving a set of simultaneous equations. The explicit schemes are easier to implement than the implicit ones, but the implicit schemes usually are stable for any grid increments. We did not investigate the later type of schemes here. While the numerical solutions generated by the $(\Delta t_c, \Delta x_c, \Delta y_c)$ grid size triplet appear to be numerically stable and have the expected qualitative properties on the time interval considered, more accurate solutions will be computed at the $D = 20$ parameter values using the fine grid solver with $(\Delta t_f, \Delta x_f, \Delta y_f) = (12.5min, 15km, 15km)$. Then only the subset of fine grid output data corresponding to the coarser grid of size $(\Delta t_c, \Delta x_c, \Delta y_c)$ is retained in the analysis. This will generate a four-dimensional array of data of size $(N_0^t+2) \times N_0^x \times N_0^y \times D = 10,003 \times 10 \times 20 \times 20$ which, even though it is the coarse grid, still requires approximately 350MB computer memory.

The size of this data set makes it difficult to save and manipulate all the values. Since specifying a general global covariance model for these data is practically impossible because of the large time dimension, we will suggest two-step methods to handle these large data sets. We shall take advantage of the fact that each of the $N_0^x \times N_0^y \times D$ time series of length N_0^t+2 is very smooth and therefore locally approximately linear. More precisely, we break the above four-dimensional array of data into m smaller arrays of size $N_m \times N_0^x \times N_0^y \times D$, with $N_m = [N_0^t/m]$ the largest integer smaller than N_0^t/m, so that each of the $m \times N_0^x \times N_0^y \times D$ time series of length N_m is approximately linear. Informal inspection of several time series suggests that m could be about 8 or larger in our study. We shall take $m = 8$ which implies $N_m = 1,250$. The methods we introduce in sections 3.3 and 3.4 require saving only a small portion of the complete data set. Specifically we define a relatively small subset Λ of time index values, and will need to retain the entire output array at only these times for each run, resulting in a data

array of size $l(\Lambda) \times N_0^x \times N_0^y \times D$ where l is the size of Λ. The method presented in section 3.3 requires $\Lambda = \Lambda_1 = \{1, N_m + 1, 2N_m + 1, ..., mN_m + 1\}$ of size $m + 1$ and the method presented in section 3.4 requires $\Lambda = \Lambda_2 = \{1, 2, 3, N_m + 1, N_m + 2, N_m + 3, 2N_m + 1, 2N_m + 2, 2N_m + 3, ..., mN_m + 1, mN_m + 2, mN_m + 3\}$ of size $3(m+1)$. The first step in each method is to specify a statistical model for data at these end points. In the second step, conditional on the data at the end points, very simple statistical models for the output data at other time points will be suggested. However, the two methods differ in that the statistical model developed in section 3.3 is applied directly to model output, while that in section 3.4 is applied to the numerical local truncation errors and, as a result, better mimics the dynamics of the system.

Optionally, one can use output data at an intermediate level of accuracy as auxiliary information in regression-type models. Let $(\Delta t_i, \Delta x_i, \Delta y_i) = (50min, 45km, 45km)$ be an intermediate grid size triplet and the corresponding output data equally spatio-temporally sampled at the coarse grid of size $(\Delta t_c, \Delta x_c, \Delta y_c)$. Notice that this intermediary grid is finer that the coarse grid because, although the time points of the two grids coincide, the spatial points of the intermediary grid include as a subset the spatial points of the coarse grid. As above, one can break this new four-dimensional array of coarser output data into m sub-arrays and save the end points. Also, one can use formula (3.4) to obtain coarser grid numerical local truncation errors. In section 3.5 we shall develop regression-like models using these coarser data as an explanatory variable.

We close this section with a remark about choosing the coarse grid. From our empirical assessment for this example, a coarser grid with increments larger than $(\Delta t_c, \Delta x_c, \Delta y_c) = (50min, 90km, 90km)$ leads to an unbounded accumulation of the statistical predictions of the numerical solutions, given the chosen experimental region of the inputs. Therefore a coarser grid would be useless from the standpoint of statistical prediction method involving numerical local truncation errors. Unfortunately, we do not have a clear rule for choosing such a working coarse grid with maximal increments. As a rule of thumb, the coarsest grid for which a bounded numerical solution is obtained can be used as a starting grid. This information is available from theoretical considerations in some situations or empirical assessments, as it was the case in this

example where the grid $(\Delta t_c, \Delta x_c, \Delta y_c) = (50min, 90km, 90km)$ was the coarsest grid that we found to provide a valid, bounded numerical solution over the entire simulation time interval. If this grid is too coarse and the statistical predictions of numerical local truncation errors accumulate unboundedly, then one should refine gradually the coarse grid until acceptable grid increments are found. For the example presented here no refinement of the above coarse grid was necessary. We should caution at this point that it is possible for other numerical schemes to allow choosing a coarser grid (e.g. implicit methods). However, if the application requires, maybe it is better not to work with the coarsest grid allowed by the numerical scheme. As we pointed out earlier, Jiang et al. (1995), among others, conducted numerical experiments to study the dependence of the solutions on the unknown parameters. Some of the features studied, such as the symmetry and periodicity of the solutions, are not visible unless the set of output data used in such studies is reasonably large. Hypothetically speaking, had we choose a numerical method that allows working with only, say, a 4×8 spatial grid instead of a 10×20 spatial grid, we probably wouldn't be able to see as clearly the two spatial gyres appearing in the contour plot of function h in Figure 3.2. Then we would be required to choose a larger set of spatial grid points to meet the goals of the study, even if the numerical scheme allows us to work with a very coarse grid.

3.3 Direct approach to statistical modeling of the numerical solutions

This method is based on a two-step strategy. The first step is to model the sampled $l(\Lambda_1) \times N_0^x \times N_0^y \times D$ array of numerical solution data. Then a DBB model conditioned on values at times in Λ_1 will be used to simulate data at the rest of the time points.

3.3.1 Modeling the $l(\Lambda_1) \times N_0^x \times N_0^y \times D$ array of numerical solution data

Let $Y^t_{\gamma,i,j}$ denote generically any of the fine grid output data $h^t_{i,j}$,$u^t_{i,j}$ or $v^t_{i,j}$ at time $t \in 1 : N_0^t$, spatial indices $i \in 1 : N_0^x$, $j \in 1 : N_0^y$ and design site $\gamma \in 1 : D$. Here, for example, $1 : N_0^t$ denotes the set $\{1, 2, ..., N_0^t - 1, N_0^t\}$ of size $N_0^t = 10,001$. Let $\{t_k\}_{k=1,...,m+1}$ be the subset of time indices t that belong to Λ_1. Then $Y^{t_k}_{\gamma,i,j}$ is the data saved at the $\Lambda_1 \times (1 : N_0^x) \times (1 :$

$N_0^y) \times (1 : D)$ sub-array of indices. Note, however, that only the time index has changed in this sub-array of indices. Statistical models of the same form will be used for the h, u and v output data, however they will be modeled separately of each other. We shall model this subset of data $Y^{t_k}_{\gamma,i,j}$ as a realization of a Gaussian process allowing the statistical parameters to have different values for h, u or v. Due to the rectangular structure of the data array, it is natural to model the correlation matrix of the normal distribution as a Kronecker product of correlations corresponding to time, space and inputs. However, because this would require a great computational effort (there are two spatial dimensions in addition to the time and inputs dimensions considered in the example of chapter 2), the approach we have chosen is to assume that the elements of array Y are spatially uncorrelated. To account for spatial non-homogeneity a location-dependent mean is assumed. Thus, if $\underline{Y}_{i,j}$ is the $(m+1) \times D$ array of data $Y^{t_k}_{\gamma,i,j}$ in vector format at location (i,j), then $\underline{Y}_{i,j}$ is modeled as $N(\mu_{i,j}\underline{1}, \sigma^2\Gamma_{\underline{\gamma},\underline{\gamma}})$, with $\Gamma_{\underline{\gamma},\underline{\gamma}} = C_{\underline{\gamma},\underline{\gamma}} \otimes C_t$ the correlation matrix not dependent on spatial locations. It is assumed that $C_{\underline{\gamma},\underline{\gamma}}(\gamma_i, \gamma_j) = \exp[-\theta_\tau(t_i^\tau - t_j^\tau)^2 - \theta_\alpha(t_i^\alpha - t_j^\alpha)^2 - \theta_\nu(t_i^\nu - t_j^\nu)^2]$ with $i,j = 1, ..., D$ and t^τ, t^α, t^ν the linearly transformed coordinates of the PDE parameters to [0,1], and $C_t(s_i, s_j) = \exp[-\theta_t(s_i - s_j)^2]$ with $i,j = 1, ..., m+1$ and s the vector of $m+1$ equally spaced points dividing [0,1] into m subintervals. These correlations generate infinitely differentiable trajectories and their choice is based on out knowledge of the smoothness of the exact solutions of the PDEs and their and smooth dependence of the system on the parameters involved. A similar statistical model, consisting in different means and a common covariance matrix has been also considered by Mitchell and Morris (1992) for a compression molding code. The likelihood can be written as

$$L(\theta_\tau, \theta_\nu, \theta_\alpha, \theta_t, \mu_{i,j}, \sigma^2; Y) =$$

$$\prod_{i,j} (\frac{1}{\sqrt{2\pi\sigma^2}})^{D(m+1)} \frac{1}{\sqrt{\det(\Gamma_{\underline{\gamma},\underline{\gamma}})}} \exp[-\frac{1}{2\sigma^2}(\underline{Y}_{i,j} - \mu_{i,j}\underline{1})'\Gamma^{-1}_{\underline{\gamma},\underline{\gamma}}(\underline{Y}_{i,j} - \mu_{i,j}\underline{1})].$$

In the computational implementation, the formula $\Gamma^{-1}_{\underline{\gamma},\underline{\gamma}} = C^{-1}_{\underline{\gamma},\underline{\gamma}} \otimes C_t^{-1}$ has been used to avoid numerical instabilities. The statistical parameters are obtained by likelihood maximization. The means and variance estimators are $\hat{\mu}_{i,j} = \frac{1'\Gamma^{-1}_{\underline{\gamma},\underline{\gamma}}\underline{Y}_{i,j}}{1'\Gamma^{-1}_{\underline{\gamma},\underline{\gamma}}1}$ and

$$\hat{\sigma}^2 = \frac{1}{DmN_0^x N_0^y} \sum_{i,j} (\underline{Y}_{i,j} - \mu_{i,j}\underline{1})'\Gamma^{-1}_{\underline{\gamma},\underline{\gamma}}(\underline{Y}_{i,j} - \mu_{i,j}\underline{1}).$$

The MLEs of θ are obtained iteratively by minimizing

$$-2\log(L) \propto \log(\hat{\sigma}^2) + \frac{\log(\det(C_t))}{m+1} + \frac{\log(\det(C_{\underline{\gamma},\underline{\gamma}}))}{D}.$$

The estimates of θ's and σ are: $[\hat{\theta}_\tau, \hat{\theta}_\alpha, \hat{\theta}_\nu, \hat{\theta}_t, \hat{\sigma}]_h = [0.5483, 0.0263, 1.1559, 2.1229, 5.3990]$ for the h data, $[\hat{\theta}_\tau, \hat{\theta}_\alpha, \hat{\theta}_\nu, \hat{\theta}_t, \hat{\sigma}]_u = [0.8045, 0.0551, 2.1459, 2.8327, 0.0185]$ for the u data and $[\hat{\theta}_\tau, \hat{\theta}_\alpha, \hat{\theta}_\nu, \hat{\theta}_t, \hat{\sigma}]_v = [0.8534, 0.0358, 1.5159, 4.3171, 0.0142]$ for the v data.

3.3.2 Modeling the output data at other time points

In section 3.3.1 we presented a model for only a fraction of the output data. This smaller amount of data corresponds to time points at the ends of intervals over which the output time series were considered to be approximately linear. The remaining data in each output time series will be modeled as discrete-time Brownian bridge (DBB) stochastic processes, constrained to pass through the small subset of points considered in section 3.3.1. For simplicity, the time series modeled as DBBs will be considered independent conditionally on their end points. The Appendix B contains a derivation for the DBB constrained to pass through two prescribed end points. The iterative relationship between elements in such a time series is simple, making it very easy to simulate realizations. In Figure 3.3 are plotted ten simulations from such an DBB process. It will also be proved that the marginal expectations of the DBB lie on the straight line connecting the two end points and that the marginal variances are quadratic in time, being zero at the extremes and attaining their maximum in the middle. The simulations appear to confirm these theoretical findings. Therefore, an important underlying assumption is that each complete output time series is approximately locally linear and its subsample of size $m+1$, if linearly interpolated, is a good approximation of the complete time series itself. Let $Y^{t,k}_{\gamma,i,j}$ denote $Y^t_{\gamma,i,j}$, with $k = 1, ..., m$ and t an integer between t_k and t_{k+1}. To recapitulate the notation used in this section, the data $Y^t_{\gamma,i,j}$ has been partitioned in two sub-sets: $Y^{t_k}_{\gamma,i,j}$ represents the data $Y^t_{\gamma,i,j}$ at time indices in Λ_1, whereas $Y^{t,k}_{\gamma,i,j}$ denotes $Y^t_{\gamma,i,j}$ at the rest of time indices, with t belonging to the k^{th} time interval (t_k, t_{k+1}) among the m intervals. An estimate of the DBB variance based on the pooled $Y^{t,k}_{\gamma,i,j}$ time series, under the assumption that these pooled time series are independent conditional on their end points $Y^{t_k}_{\gamma,i,j}$, is

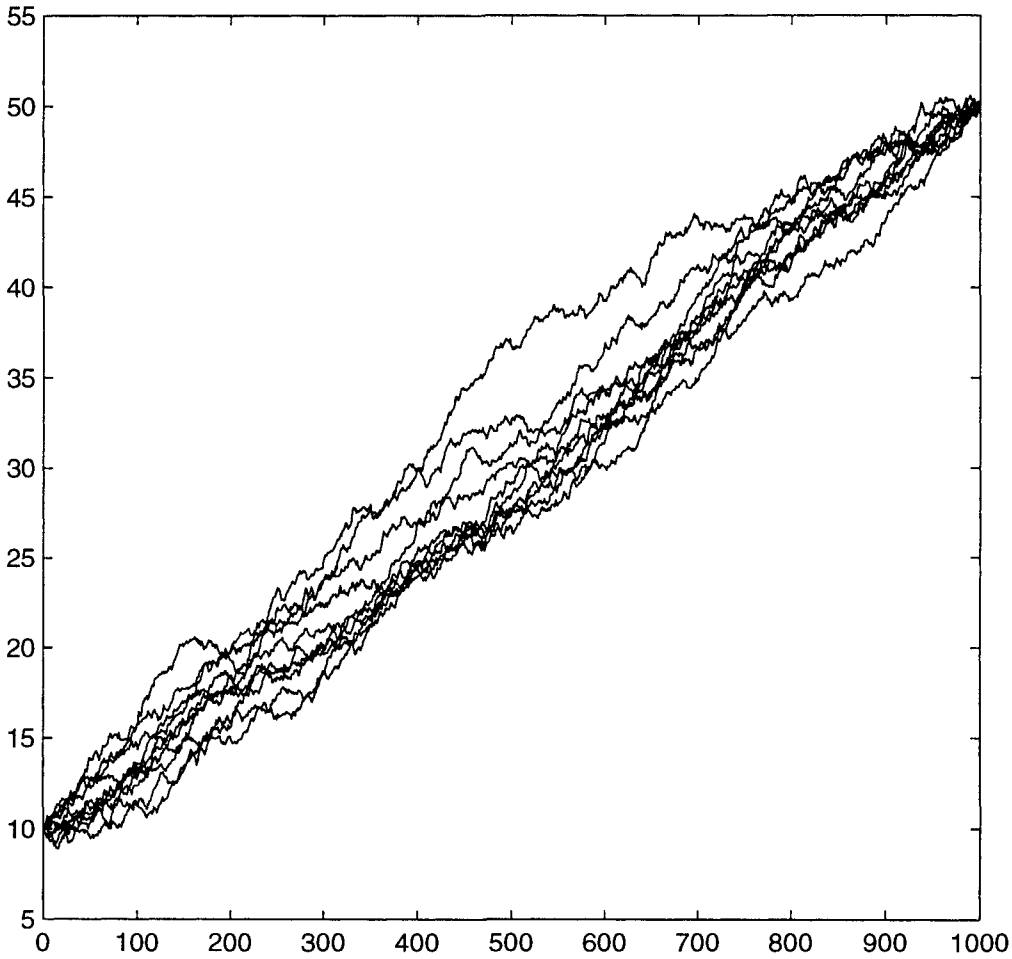

Figure 3.3 Ten simulated realizations of a DBB

$\hat{\sigma}^2_{DBB} = \frac{1}{DmN_0^x N_0^y} \sum_{\gamma,i,j,k} \{ \frac{1}{N_m - 1} \sum_{t=1}^{N_m - 1} [\frac{N_m - t + 1}{N_m - t} (Y^{t+1,k}_{\gamma,i,j} - \frac{1}{N_m - t + 1} Y^{t_{k+1}}_{\gamma,i,j} - \frac{N_m - t}{N_m - t + 1} Y^{t,k}_{\gamma,i,j})^2] \}$. The results are: $[\hat{\sigma}_{DBB}(h); \hat{\sigma}_{DBB}(u); \hat{\sigma}_{DBB}(v)] = [0.10514; 0.00057; 0.00061] \times 10^{-3}$. Here $N_m = 1,250$. It is important to notice that this estimator involves all the output data $Y^t_{\gamma,i,j}$, but the required sums can be accumulated as the numerical solution is computed, at little additional cost in memory or arithmetic, and in particular does not require storage of the complete output. The DBB process requires estimating only one parameter, the variance, and its simplicity is particularly useful here.

As a postscript, we note that the success of this approach depends on the local linearity of the time series, and that an interpolated model of the data at Λ_1 explains most of the variability in the entire data set. to illustrate this, we observe that treating the pooled time series $Y^{t,k}_{\gamma,i,j}$ as traditional autoregression processes in t (that is, unconditional on the end points) for the purpose of estimating the pooled noise variance results in an over-estimate because the constraints specific to DBB are not taken into account. Indeed, had these time series been treated as usual autoregressions, unconditional on their end points, the results would be: $[\hat{\sigma}_{AR}(h); \hat{\sigma}_{AR}(u); \hat{\sigma}_{AR}(v)] = [0.72998; 0.00344; 0.00317] \times 10^{-3}$, which are considerably larger than the conditional variance estimates reported above.

3.3.3 Prediction at a new site γ_0

The two steps below summarize the prediction of the complete output time series at a new testing site.

3.3.3.1 Simulation from the posterior distribution

The posterior distribution at the new site based on the model in section 3.3.1 will be used to obtain predicted values at time points in Λ_1 and at each location (i, j). For example, if m=8 and the output data correspond to h, these are in fact the predicted values for $\{h^{t_k}_{i,j} | t_k \in \Lambda_1\}$ at γ_0, with $\Lambda_1 = \{1, 1251, 2501, 3751, 5001, 6251, 7501, 8751, 10001\}$. The mean vector of this posterior distribution is

$$M_{i,j} = \mu_{i,j}\underline{1} + \Gamma_{\gamma_0,\underline{\gamma}}\Gamma^{-1}_{\underline{\gamma},\underline{\gamma}}(\underline{Y}_{i,j} - \mu_{i,j}\underline{1}) = \mu_{i,j}\underline{1} + [(C_{\gamma_0,\underline{\gamma}} \otimes C_t) * (C^{-1}_{\underline{\gamma},\underline{\gamma}} \otimes C_t^{-1})](\underline{Y}_{i,j} - \mu_{i,j}\underline{1}) =$$

$$\mu_{i,j}\underline{1} + [(C_{\gamma_0,\underline{\gamma}}C_{\underline{\gamma},\underline{\gamma}}^{-1}) \otimes I_{m+1}](\underline{Y}_{i,j} - \mu_{i,j}\underline{1}),$$

where I_{m+1} is the identity matrix. The square root (Choleski) of the posterior covariance matrix, which will be used in the actual simulation, is

$$CHOL[\sigma^2(\Sigma_{\gamma_0,\gamma_0} - \Gamma_{\gamma_0,\underline{\gamma}}\Gamma_{\underline{\gamma},\underline{\gamma}}^{-1}\Gamma'_{\gamma_0,\underline{\gamma}})] = \sigma\sqrt{C_{\gamma_0,\gamma_0} - C_{\gamma_0,\underline{\gamma}}C_{\underline{\gamma},\underline{\gamma}}^{-1}C'_{\gamma_0,\underline{\gamma}}} * CHOL(C_t).$$

We shall obtain R simulations from this posterior distribution, where the statistical parameters will be replaced by their estimates.

3.3.3.2 DBB simulation

Next we shall "connect" the above predicted values through DBBs to obtain predicted values at intermediate time points. Consider, for a simpler explanation, that the function of interest is again h and $m = 8$. Let $h_{i,j}^1(r)$ be the r^{th} simulated value for $h_{i,j}^1$ and $h_{i,j}^{1251}(r)$ be the r^{th} simulated value for $h_{i,j}^{1251}$. Simulated values at time points between 1 and 1251 will be obtained from a simulated DBB of standard deviations as estimated in section 3.3.2 and constrained to pass through these two simulated values. One conditional simulation is executed corresponding to each unconditional simulation in step 3.3.3.1. Thus, R simulations for the first temporal segment among m segments of the output time series at location (i, j) have been completed. The rest of $m - 1$ segments are completed similarly. Simulation summaries such as median and percentile-based prediction intervals will be reported at each of the $N_0^t = 10001$ time points and each spatial location (i, j). It is important to notice that there are two distinct sources of uncertainty here: one that corresponds to the model described in section 3.3.3.1 and the second that comes from the model described in section 3.3.3.2. The simulation method presented provides a way to simultaneously handle both sources of uncertainty.

3.4 A statistical model incorporating code information

In this section we shall use a model similar to the one presented section 3.3.1 as a starting point and successively add information extracted from the finite difference code. The intent is to correct for the linear approximation in a meaningful, more dynamically correct way.

3.4.1 Modeling the $l(\Lambda_2) \times N_0^x \times N_0^y \times D$ array of numerical solution data

As in section 3.3.1 we first model a subset of the output data. Here, groups of three temporally consecutive data are sampled, as opposed to section 3.3.1 where each group consisted of data at only one time point. Again let $Y_{\gamma,i,j}^{t_k}$ denote generically any of the fine grid output data h, u or v at design site γ, with (t_k, i, j, γ) in the $\Lambda_2 \times (1 : N_0^x) \times (1 : N_0^y) \times (1 : D)$ array of indices. The reason for sampling groups of three temporally consecutive data is that each such group allows calculation of numerical local truncation errors at one time point, according to the relationship (3.4). We shall again model $Y_{\gamma,i,j}^{t_k}$, with t_k in Λ_2, using a Gaussian process. If $\underline{Y}_{i,j}$ is the $(m+1) \times D$ array of data $Y_{\gamma,i,j}^{t_k}$ in vector format at location (i,j), then $\underline{Y}_{i,j}$ is modeled as $N(\mu_{i,j}\underline{1}, \sigma^2\Gamma_{\underline{\gamma},\underline{\gamma}})$, with $\Gamma_{\underline{\gamma},\underline{\gamma}} = C_{\underline{\gamma},\underline{\gamma}} \otimes C_t$, where $C_{\underline{\gamma},\underline{\gamma}}$ is defined as in section 3.3.1 and C_t corresponds to Λ_2 (independence among spatial locations is assumed). The formulae for the MLEs of $\mu_{i,j}$ and σ^2 are similar to those presented in section 3.3.1 and the MLEs of the parameters θ associated with the covariance matrix are obtained numerically. These MLEs for θ's and σ's are: $[\hat{\theta}_\tau, \hat{\theta}_\alpha, \hat{\theta}_\nu, \hat{\theta}_t, \hat{\sigma}]_h = [0.5488, 0.0290, 1.2233, 2.9983, 2.7707]$ for the h data, $[\hat{\theta}_\tau, \hat{\theta}_\alpha, \hat{\theta}_\nu, \hat{\theta}_t, \hat{\sigma}]_u = [0.7482, 0.0441, 2.4283, 3.7532, 0.0118]$ for the u data and $[\hat{\theta}_\tau, \hat{\theta}_\alpha, \hat{\theta}_\nu, \hat{\theta}_t, \hat{\sigma}]_v = [0.7502, 0.0286, 1.6999, 5.3258, 0.0101]$ for the v data. In this section C_t has dimension $3(m+1)$ since more time points have been sampled. We should caution that as m increases there is a possibility that C_t will become numerically unstable. This instability is due to an increase in the the dimension of C_t as well as to large correlations within groups of three consecutive time points. Since the distance between the time points within each cluster of 3 is so small and C_t is based on the distance between these time points, the rows (or columns) of C_t corresponding to the clustered points are almost equal, which contributes to its numerical instability. Should this situation occur the effects of numerical instability may be minimized by writing C_t as yet another Kronecker product $C_t = C_{t_a} \otimes C_{t_b}$ where C_{t_a} has dimension $m+1$ and C_{t_b} has dimension 3. C_{t_b} is the "within-groups" correlation matrix and C_{t_a} is the "between-groups" correlation matrix. This, however, adds a new factor in the Kronecker product of the overall correlation matrix and adds somewhat to the time required by our implementation of the likelihood maximization algorithm because the Kronecker function "kron" in MATLAB is

called one more time at each iteration in the optimization process.

3.4.2 Modeling the intermediary numerical local truncation error data

In section 3.3.2 it was shown that along with the numerical solutions one can obtain also the numerical local truncation errors E. Here we shall assume that the numerical truncation error time series are approximately locally linear and can be modeled within relatively short intervals as DBBs just as the numerical solutions were modeled in section 3.3.2. Thus, it will be assumed that a complete fine grid approximate local truncation error time series of length $N_0^t = 10001$ can be partitioned in m subsampled time series which are approximately linear. As was the case in modeling output directly, informal inspection of several such numerical local truncation error time series reveals that $m = 8$ would be a good choice. An estimator of the DBB variance is similar to the one presented in section 3.3.2 with numerical truncation error data E instead of numerical solution output data Y. The three DBB variance estimates corresponding to h, u and v are: $[\hat{\sigma}_{DBB}(h); \hat{\sigma}_{DBB}(u); \hat{\sigma}_{DBB}(v)] = [0.80005; 0.01166; 0.00971] \times 10^{-5}$. These estimates are smaller than those in the previous section because for this example the numerical local truncation errors are smaller in absolute value than the numerical solutions.

3.4.3 Prediction at a new site γ_0

The following three steps summarize the prediction at a new site.

3.4.3.1 Simulation from the posterior distribution

The model developed in section 3.4.1 will be used to obtain R simulated output values at the $3(m+1)$ time points indexed by Λ_2. If m=8 and the output data to be predicted is h, these are simulated values for $\{h_{i,j}^{t_k} | t_k \in \Lambda_2\}$ at γ_0, with $\Lambda_2 = \{1, 2, 3, 1251, 1252, 1253, 2501, 2502, 2503, ..., 10001, 10002, 10003\}$. These simulated values will be obtained from the posterior distribution at the new testing site γ_0 of the model in section 3.4.1. This posterior distribution is multivariate normal of dimension $3(m + 1)$, with mean vector and covariance matrix similar to those presented in section 3.3.3.1 for modeling Y, with formulae modified to accommodate

time index values from Λ_2 rather than Λ_1.

3.4.3.2 Local numerical truncation error simulation

Each simulated group of three consecutive output data will be used to obtain a simulated numerical local truncation error. Consider again the example $m = 8$. Let $r = 1, ..., R$ and $(h^1_{i,j}(r), h^2_{i,j}(r), h^3_{i,j}(r))$ be simulated values for $(h^1_{i,j}, h^2_{i,j}, h^3_{i,j})$, $(u^1_{i,j}(r), u^2_{i,j}(r), u^3_{i,j}(r))$ be simulated values for $(u^1_{i,j}, u^2_{i,j}, u^3_{i,j})$ and $(v^1_{i,j}(r), v^2_{i,j}(r), v^3_{i,j}(r))$ be simulated values for $(v^1_{i,j}, v^2_{i,j}, v^3_{i,j})$, respectively, obtained in section 3.4.3.1. For each $r = 1, ..., R$, these simulated values will be used in formula (3.4) to obtain R simulated local truncation errors for $E(h)^1_{i,j}$, $E(u)^1_{i,j}$ and $E(v)^1_{i,j}$. This process will be applied to obtain R simulated values for $E(h)^t_{i,j}$, $E(u)^t_{i,j}$ and $E(v)^t_{i,j}$, at all t in Λ_1. Notice that output numerical solutions saved at time points in Λ_2 in fact produce numerical truncation errors only at time points in Λ_1, according to the relationship (3.4).

3.4.3.3 DBB simulation

Let $E(h)^1_{i,j}(r)$ and $E(h)^{1251}_{i,j}(r)$ be simulated values for $E(h)^1_{i,j}$ and $E(h)^{1251}_{i,j}$, with $r = 1, ..., R$. Simulated values for the local truncation errors at time points between 1 and 1251 will be obtained by simulating DBBs connecting these two simulated values and variance estimated is section 3.4.2. Therefore we have R simulated values for $E(h)^t_{i,j}$ at each spatial location (i, j), with t between 1 and 1251. Similarly, R simulated values for each $E(u)^t_{i,j}$ and $E(v)^t_{i,j}$ will be obtained at each spatial location (i, j) and t between 1 and 1251. Next, the relationship (3.4) will be used iteratively to obtain R simulated values for the numerical solutions h, u and v at time $t + 1$, given the simulated numerical local truncation errors at time $t - 1$ and the values of the numerical solutions at times $t - 1$ and t, with t between 2 and 1252. Thus, simulated outputs for the first temporal segment among the m is completed. It is important to notice that these simulated values for the numerical solutions obtained iteratively at the final time steps $t = 1251$, $t = 1252$ and $t = 1253$ will be different from the simulated values for the same output data obtained in section 3.4.3.1 above from the posterior distribution.

The later simulated values can be regarded as a version of those obtained in 3.4.3.1, updated with information about the dynamics of the PDE system provided by the finite difference code through the use of formula (3.4). These updated simulated values will replace those obtained in 3.4.3.1 at time steps $t = 1251$, $t = 1252$ and $t = 1253$. This re-setting of simulated output helps prevent a faster accumulation of the errors when the iterative relationship (3.4) is used for the next temporal segments. To generate simulated values for the second temporal segment among the m, with t between 1254 and 2503, we use the updated simulated values for the numerical solutions in the relationship (3.4) in order to generate R updated simulated values for the local truncation errors $E(h)_{i,j}^{1251}$, $E(u)_{i,j}^{1251}$ and $E(v)_{i,j}^{1251}$. These, along with the R simulated values for $E(h)_{i,j}^{2501}$, $E(u)_{i,j}^{2501}$ and $E(v)_{i,j}^{2501}$ obtained at step 3.4.3.2, will be used as end points in DBBs of variance estimated in section 3.4.2 to simulate values for the local truncation errors at time steps between 1251 and 2501. As above, these will be used in the relationship (3.4) to obtain iteratively R simulated values for the numerical solutions at time steps t between 1251 and 2503. The R simulated values for the numerical solutions at time steps $t = 2501$, $t = 2502$ and $t = 2503$ obtained as a result of this iteration will replace those obtained in section 3.4.3.1 from the posterior distribution, and the outputs for the second temporal segment among the m is simulated. The rest of the simulated temporal segments will be obtained similarly. Thus, there will be R simulated numerical solutions generated at each of the $N_0^t = 10001$ time steps and each of the $N_0^x \times N_0^y = 11 \times 21$ spatial indices. Medians and percentile-based intervals will be reported to summarize the prediction.

3.5 Coarse numerical solutions as auxiliary information

In this section we shall present methods for using auxiliary information provided by coarser numerical solutions which are less expensive but not as accurate as a fine-grid numerical solution. This is similar, in some respects, to the method presented in Kennedy and O'Hagan (2000). However, here we look more closely at the output data generating mechanism and incorporate code information into our statistical method (section 3.5.2).

3.5.1 The direct approach

This method will follow closely the steps presented in section 3.3. Let $X^t_{\gamma,i,j}$ denote generically any of the coarse grid $(\Delta t_1, \Delta x_1, \Delta y_1)$ output data $h^t_{i,j}$,$u^t_{i,j}$ or $v^t_{i,j}$ at design site γ and $X^{t_k}_{\gamma,i,j}$ be X sampled at the $\Lambda_1 \times (1 : N_0^x) \times (1 : N_0^y) \times (1 : D)$ array of indices. Then the fine grid data $\underline{Y}_{i,j}$ are modeled as $N(\mu^0_{i,j}\underline{1}+\mu^1_{i,j}\underline{X}_{i,j}, \sigma^2\Gamma_{\underline{\gamma},\underline{\gamma}})$, with $\Gamma_{\underline{\gamma},\underline{\gamma}} = C_{\underline{\gamma},\underline{\gamma}} \otimes C_t$. Here, $\underline{Y}_{i,j}$ and $\underline{X}_{i,j}$ are the $(m+1) \times D$ arrays of output data $Y^{t_k}_{\gamma,i,j}$ and $X^{t_k}_{\gamma,i,j}$ respectively, in vector format at location (i,j). The MLEs for the regression parameters and the variance are

$$(\hat{\mu}^0_{i,j}, \hat{\mu}^1_{i,j}) = ([\underline{1}, \underline{X}_{i,j}]'\Gamma^{-1}_{\underline{\gamma},\underline{\gamma}}[\underline{1}, \underline{X}_{i,j}])^{-1}([\underline{1}, \underline{X}_{i,j}]'\Gamma^{-1}_{\underline{\gamma},\underline{\gamma}}\underline{Y}_{i,j})$$

and

$$\hat{\sigma}^2 = \frac{1}{DmN_0^xN_0^y}\sum_{i,j}(\underline{Y}_{i,j} - \mu^0_{i,j}\underline{1} - \mu^1_{i,j} * \underline{X}_{i,j})'\Gamma^{-1}_{\underline{\gamma},\underline{\gamma}}(\underline{Y}_{i,j} - \mu^0_{i,j}\underline{1} - \mu^1_{i,j} * \underline{X}_{i,j}).$$

The MLEs of the parameters θ's are obtained by numerical maximization of the likelihood. We obtained $[\hat{\theta}_\tau, \hat{\theta}_\alpha, \hat{\theta}_\nu, \hat{\theta}_t, \hat{\sigma}]_h = [0.5698, 0.0339, 1.3033, 2.4214, 3.5714]$, $[\hat{\theta}_\tau, \hat{\theta}_\alpha, \hat{\theta}_\nu, \hat{\theta}_t, \hat{\sigma}]_u = [0.7819, 0.0592, 2.2296, 3.0891, 0.0149]$, $[\hat{\theta}_\tau, \hat{\theta}_\alpha, \hat{\theta}_\nu, \hat{\theta}_t, \hat{\sigma}]_v = [0.9251, 0.0430, 1.5846, 4.3096, 0.0104]$. The correlation matrices used are the same as in section 3.3.

Next, simulated values for the fine grid numerical solutions at intermediate time points need to be obtained. One option is to regress the fine grid output data on the coarse grid output data and then model the residuals as DBBs. This approach, however, requires saving and manipulating the complete sets of fine and coarse grid output data in order to obtain regression coefficient estimates and residuals, which we try to avoid to minimize the requirement for computational resources. Instead, we model the difference between fine and coarse output data as DBBs. This approach does not require a great computational effort since the DBB variances can be estimated simultaneously with output data generation. These estimates are $[\hat{\sigma}_{DBB}(h); \hat{\sigma}_{DBB}(u); \hat{\sigma}_{DBB}(v)] = [0.12180; 0.00108; 0.00118] \times 10^{-3}$. In contrast, we were able to model the fine grid output Y at time points in Λ_1 by regression on the coarse grid output (as described at the beginning of this section 3.5.1) because this regression approach for the end points did not require a great computational effort.

In order to predict at a new site γ_0 the coarse numerical solution will be generated first.

Denote by $X^{t_k}_{\gamma_0,i,j}$ the vector of coarse numerical solutions at time points in Λ_1. Then R simulated values for the fine grid numerical solutions at time points in Λ_1 will be obtained from the posterior distribution of mean vector

$$M_{i,j} = \mu^0_{i,j}\underline{1} + \mu^1_{i,j} * \underline{X}_{\gamma_0,i,j} + [(C_{\gamma_0,\underline{\gamma}} C^{-1}_{\underline{\gamma},\underline{\gamma}}) \otimes I_{m+1}](\underline{Y}_{i,j} - \mu^0_{i,j}\underline{1} - \mu^1_{i,j} * \underline{X}_{i,j}),$$

where $\underline{X}_{\gamma_0,i,j}$ is the vector of coarse output data $X^{t_k}_{\gamma_0,i,j}$ at location (i,j) and new site γ_0. The posterior covariance matrix at the new site will have the same form as the one presented in section 3.3 since it is not influenced by the first order moment. A total of R simulations $Y^{t_k}_{\gamma_0,i,j}(r)$ will be obtained from this posterior distribution. The prediction will be completed with the simulation of fine grid numerical solutions at the intermediate time points. For example, simulations at times t between 1 and 1251 will be obtained as a sum between $X^t_{\gamma_0,i,j}$ and simulations from DBBs of variance estimated above and end points $Y^{t_1}_{\gamma_0,i,j}(r) - X^{t_1}_{\gamma_0,i,j}$ and $Y^{t_2}_{\gamma_0,i,j}(r) - X^{t_2}_{\gamma_0,i,j}$, where $t_1 = 1$ and $t_2 = 1251$. Medians and percentile-based intervals will be reported as prediction summaries.

3.5.2 Method incorporating code information

The auxiliary information provided by the coarse numerical solution can be used in an alternative way. If $Y^{t_k}_{\gamma,i,j}$ and $X^{t_k}_{\gamma,i,j}$ are the fine and coarse output data saved at the points $\Lambda_2 \times (1 : N^x_0) \times (1 : N^y_0) \times (1 : D)$ then $\underline{Y}_{i,j}$ can be modeled as $N(\mu^0_{i,j}\underline{1} + \mu^1_{i,j}\underline{X}_{i,j}, \sigma^2\Gamma_{\underline{\gamma},\underline{\gamma}})$, with $\Gamma_{\underline{\gamma},\underline{\gamma}} = C_{\underline{\gamma},\underline{\gamma}} \otimes C_t$, with C_t of dimension $3(m+1)$, and where $\underline{Y}_{i,j}$ and $\underline{X}_{i,j}$ are the fine and coarse data in vector format at location (i,j). The MLEs for the parameters $(\mu^0_{i,j}, \mu^1_{i,j})$ and σ^2 are similar to those in section 3.5.1 and the correlation parameters θ in the covariance matrix will again be estimated by numerical maximization of the likelihood. These estimates are: $[\hat{\theta}_\tau, \hat{\theta}_\alpha, \hat{\theta}_\nu, \hat{\theta}_t, \hat{\sigma}]_h = [0.5258, 0.0350, 1.2704, 3.1578, 2.2063]$ for the h output data, $[\hat{\theta}_\tau, \hat{\theta}_\alpha, \hat{\theta}_\nu, \hat{\theta}_t, \hat{\sigma}]_u = [0.7342, 0.0448, 2.4591, 4.1238, 0.0101]$ for the u output data and, finally, $[\hat{\theta}_\tau, \hat{\theta}_\alpha, \hat{\theta}_\nu, \hat{\theta}_t, \hat{\sigma}]_v = [0.6402, 0.0328, 1.5951, 5.1747, 0.0095]$ for the v output data. The difference of the fine and coarse grid numerical local truncation errors will be modeled locally as DBBs. The corresponding estimates are $[\hat{\sigma}_{DBB}(h); \hat{\sigma}_{DBB}(u); \hat{\sigma}_{DBB}(v)] = [0.20208; 0.00101; 0.00131] \times 10^{-4}$. In order to predict at a new site γ_0 one must first obtain R simulated fine grid numerical

solutions at time points in Λ_2 from the posterior distribution based on the above multivariate normal distribution. Each of these will be used to produce a single set of R simulated fine grid numerical local truncation errors $E(h)_{i,j}^{t_k}(r)$, $E(u)_{i,j}^{t_k}(r)$ and $E(v)_{i,j}^{t_k}(r)$ at time points t_k in Λ_1 by using the relationship (3.4). In the case $m = 8$, if $e(h)_{i,j}^{t_1}$, $e(u)_{i,j}^{t_1}$, $e(v)_{i,j}^{t_1}$ and $e(h)_{i,j}^{t_2}$, $e(u)_{i,j}^{t_2}$, $e(v)_{i,j}^{t_2}$ denote the coarse grid numerical local truncation errors at time points $t_1 = 1$ and $t_2 = 1251$, then simulated values for the difference between fine and coarse grid numerical local truncation errors at time points t between t_1 and t_2 will be obtained by using DBBs with fixed end points $E(h)_{i,j}^{t_1}(r) - e(h)_{i,j}^{t_1}$ and $E(h)_{i,j}^{t_2}(r) - e(h)_{i,j}^{t_2}$, $E(u)_{i,j}^{t_1}(r) - e(u)_{i,j}^{t_1}$ and $E(u)_{i,j}^{t_2}(r) - e(u)_{i,j}^{t_2}$, $E(v)_{i,j}^{t_1}(r) - e(v)_{i,j}^{t_1}$ and $E(v)_{i,j}^{t_2}(r) - e(v)_{i,j}^{t_2}$, respectively. Simulated values for the fine grid numerical local truncation errors at intermediate time points between t_1 and t_2 will be obtained by adding coarse grid numerical local truncation errors to these DBB simulations of differences. Now we are in the situation described in section 3.4. Simulated values for fine grid numerical solutions at intermediate time points will be obtained iteratively from the relationship (3.4), therefore simulating the first temporal segment among the m. These simulated values for the fine grid numerical solutions at $t_2, t_2 + 1, t_2 + 2$ obtained iteratively will replace the former ones, obtained from the posterior distribution. The procedure will be repeated for all m temporal segments. Medians and percentile-based intervals will be reported as predictions.

3.6 Results

The methods presented in the previous sections will be tested by comparing the predictions against the fine grid numerical solutions at a set of $P = 20$ new testing sites which are plotted as open circles in Figure 3.1. These were obtained as a maximin LHD within a set of 1,000,000 LHDs (the same process used in constructing the experimental design, section 3.2) and are considered representative for the rest of untested sites. In terms of computational time, one run of fine grid numerical solution with $(\Delta t, \Delta x, \Delta y) = (12.5min, 15km, 15km)$ takes about 100min on a 1GB RAM, 400MHz 64-bit processor workstation in MATLAB. The coarse grid solution corresponding to $(\Delta t_1, \Delta x_1, \Delta y_1) = (50min, 45km, 45km)$ which was used

as auxiliary information in section 3.5 takes about 5.5 min. The prediction with $R = 19$ for the Direct method of section 3.3 takes about 6.5min and the prediction method based on code information presented in section 3.4 takes about 11.5min per testing site, again with $R = 19$. The later method is slower because it requires the use of relationship (3.4) at each iteration. The computational time for the predictions of section 3.5 is approximately the sum of the computational times reported above. More precisely, the Direct method using auxiliary information takes about 5.5min + 6.5min = 12min per testing site, whereas the method based on local truncation errors using auxiliary information takes about 5.5min + 11.5min = 17min per testing site. The predictions were also compared against a numerical solution of intermediate grid size triplet $(\Delta t, \Delta x, \Delta y) = (25min, 30km, 30km)$ taking about 13.5 min, that is about $8 = 2^3$ faster than the fine grid numerical solutions since its grid increments are twice as big in each direction. This computational time required to solve the problem on the intermediate size grid is comparable to that of the statistical predictions.

The prediction accuracy measures considered here are the mean coverages (COVER) of the fine grid solution and an overall Euclidean distance between the fine grid solution and predicted values over the set of $P = 20$ testing sites. For example, to evaluate the measures for the function h, one possibility is to define this distance as the root mean square error

$$RMSE = \sqrt{\frac{1}{N_x N_y N_t P} \sum_{i,j,t,\gamma_0} (h^t_{i,j,\gamma_0} - \hat{h}^t_{i,j,\gamma_0})^2},$$

where $\hat{h}^t_{i,j,\gamma_0}$ is the median among simulations at each point (i, j, t, γ_0). This measure, however, does not account for the prediction error. Another measure is the statistical root mean square error

$$SRMSE = \sqrt{\frac{1}{N_x N_y N_t P} \sum_{i,j,t,\gamma_0} [\frac{1}{R} \sum_r (h^t_{i,j,\gamma_0} - h^t_{i,j,\gamma_0}(r))^2]},$$

where $h^t_{i,j,\gamma_0}(r)$ is the r^{th} simulation of h^t_{i,j,γ_0}; this measure does incorporate the prediction error. These measures were evaluated at only 1,001 equally spaced time points among the $N_t = 10,001$ due to computer memory limitations. However, since the time series are highly smooth one can consider that one output datum is representative for 10 other output data in its centered temporal neighborhood. Moreover, only $R = 19$ simulations were considered,

again due to computational limitations. One can argue that this is a small number of simulations by statistical standards, but in some problems more simulations cannot be afforded. Because of these reasons perhaps one should regard the results presented in Table 3.1 with some reservations. However, it appears that some patterns in this table develop more clearly than others and these are that we want to emphasize. The extreme order statistics for the $R = 19$ simulations at each point will provide 90% prediction intervals. In Table 3.1, Direct (NAI) refers to the direct approach presented in section 3.3 where no auxiliary information provided by a coarse numerical solution has been used. TER (NAI) refers to the method presented in section 3.4 based on local truncation errors without using coarse numerical solutions as auxiliary information. Direct (AI) and TER (AI) refer to these two methods based on auxiliary information provided by a coarse numerical solution, which was the subject of section 3.5. The mean coverage provided by all methods seems to be reasonably close to the 90% target.

When no auxiliary information (NAI) has been used, the SRMSEs are somewhat smaller for the TER method than for the Direct method. By these measures, the TER method seems to be better than the Direct method, although it costs more. The last two columns of Table 3.1 show the results obtained when the coarse numerical solution with $(\Delta t_i, \Delta x_i, \Delta y_i) = (50min, 45km, 45km)$ has been used as auxiliary information according to the method presented in section 3.5. The Direct method seems to perform better than the TER. Also, when compare the (AI) versus (NAI), the use of auxiliary information improved the accuracy measures for each method, the Direct and TER (except for the SRMSE of h in the TER case). Note however that the improvement seems to be more substantial in the case of the Direct method.

To have a better understanding of these results, we calculated the discrepancy measure between the fine grid numerical solution and the intermediate resolution grid $(\Delta t_I, \Delta x_I, \Delta y_I) = (25min, 30km, 30km)$ numerical solution. This measure is similar to the RMSE formula above, with the intermediate numerical solution replacing the medians. Notice that there is no noise associated with this intermediate numerical solution, and therefore the SRMSE measure has no meaning here. These are $RMSE(h) = 1.8914$, $RMSE(u) = 0.0056$ and $RMSE(v) = 0.0045$.

All the statistical methods described here provide predictions of much better accuracy than this intermediate grid numerical solution, at a comparable cost.

We visually inspected several time series of numerical solutions and numerical local truncation errors at different resolution levels for different design sites and noticed that the fine and coarse grid numerical solution time series are "more similar" (in a loose sense) than fine and coarse grid numerical local truncation error time series, which perhaps explains in part why the coarse grid output was more helpful as auxiliary information for the Direct than for the TER method. These results appear to be in some agreement with the findings in chapter 2 where the addition of auxiliary information improved the Direct method far more than the TER method. However, it is not easy to directly compare the results presented in the two chapters because the nature of the two systems of differential equations is different: the system analyzed in chapter 2 has two steady states (winner and loser) for most of the input points, whereas the system of PDEs presented here does not have a similar property.

Table 3.1 Coverage, RMSE and SRMSE measures

h	Direct (NAI)	TER (NAI)	Direct (AI)	TER (AI)
COVER	0.92893	0.90399	0.94403	0.94281
RMSE	0.18833	0.18269	0.15654	0.17368
SRMSE	0.24550	0.21005	0.19601	0.21642
u	Direct (NAI)	TER (NAI)	Direct (AI)	TER (AI)
COVER	0.95306	0.93782	0.95633	0.94210
RMSE	0.00117	0.00122	0.00098	0.00107
SRMSE	0.00153	0.00138	0.00127	0.00127
v	Direct (NAI)	TER (NAI)	Direct (AI)	TER (AI)
COVER	0.91189	0.92526	0.93668	0.95604
RMSE	0.00080	0.00082	0.00066	0.00074
SRMSE	0.00099	0.00095	0.00082	0.00095

While Table 3.1 showed average results over the design space, it is also useful to look at predictions at particular testing sites. Figure 3.4 shows predictions for a testing site somewhere close to the center of the experimental region. More specifically, the parameters corresponding to this site are: the fluid viscosity $\nu = 710.5$, the wind stress $\tau = 0.0421$ and the wind asymmetry $\alpha = 0.026$. For this site, we have chosen a separate spatial location for each

function h, u and v as follows: the grid point $(i, j) = (3, 15)$ for h, $(i, j) = (4, 15)$ for u and $(i, j) = (4, 9)$ for v. Then the fine grid time series is plotted along with point predictors and 90% prediction intervals corresponding to the Direct (NAI, AI) and TER (NAI, AI) methods. Also plotted is the intermediate resolution grid with $\Delta x_I = 30km$ numerical solution (denoted 'C30') at the same testing site and spatial locations. For the purpose of plotting only, the time series of length $N_t = 10,001$ were divided into 100 equally spaced intervals and only the 101 end points of these intervals were retained. This has little effect on the appearance of the figure because the time series are quite smooth. Since the DBB is linear in the marginal means as shown in the Appendix B, the Direct (NAI) method prediction inherits this locally linear shape. The point predictors, however, are accompagnied by prediction intervals which provide coverage that is close to the nominal value as Table 3.1 shows. The predictions produced by the TER (NAI) method are qualitatively better since predictions of numerical local truncation errors that are locally linear will be transformed into curvilinearly shaped predictions for the numerical solutions by using the relationship (3.4). When auxiliary information (AI) has been used according to the method of section 3.5, the predictions provided by the Direct method are no longer piece-wise linear. Rather, these predictors borrow from the shape of the coarse numerical solutions used as auxiliary information. One can notice that all the statistical prediction time series are more accurate than the corresponding time series C30 of the intermediate resolution grid requiring comparable resources, which is in agreement with the overall RMSE results.

3.7 Conclusion

We presented methods of statistical analysis for multivariate computer output of finite difference solvers when the temporal dimension is large. The main challenge is the need to avoid saving and manipulating the resulting very large sets of output data. We suggested and demonstrated two-step methods that require saving only a small fraction of these data. All the statistical methods presented here result in more accurate predictions than a faster running finite difference solver using comparable resources. There are however noticeable

differences among these statistical methods for this example. When no auxiliary information has been used, the method based on direct analysis of the output is faster but less accurate than the method based on the analysis of the numerical local truncation errors. The statistical predictors using auxiliary information in regression type models are somewhat slower but more accurate than the statistical predictors that do not use auxiliary information. Overall, taking into account the computational resources used and the accuracy, the analysis suggests that the statistical method based on local truncation errors without auxiliary information and the statistical method using auxiliary information and analyzing the numerical solutions directly are better. However, the differences in performance between all these statistical methods are relatively small and results could be different for other computer models. One possible explanation why none of these statistical methods distinguishes itself in performance from the others is that this performance depends in large on the DBB processes since most of the output data is modeled by such a stochastic process. One way to reduce the influence of the DBB is to improve the local linearity assumption of the time series by increasing m, the number of approximately locally linear segments. However, this requires saving more output data and the prediction procedure associated with the $m+1$ end points of DBBs will be slower. Another way to increase the performance of the statistical predictors, which we did not investigate here, would be to replace the DBBs with stochastic models that are smoother in time. However, such stochastic models need to be simple enough to simulate from and to allow obtaining parameter estimates simultaneously with the numerical solution.

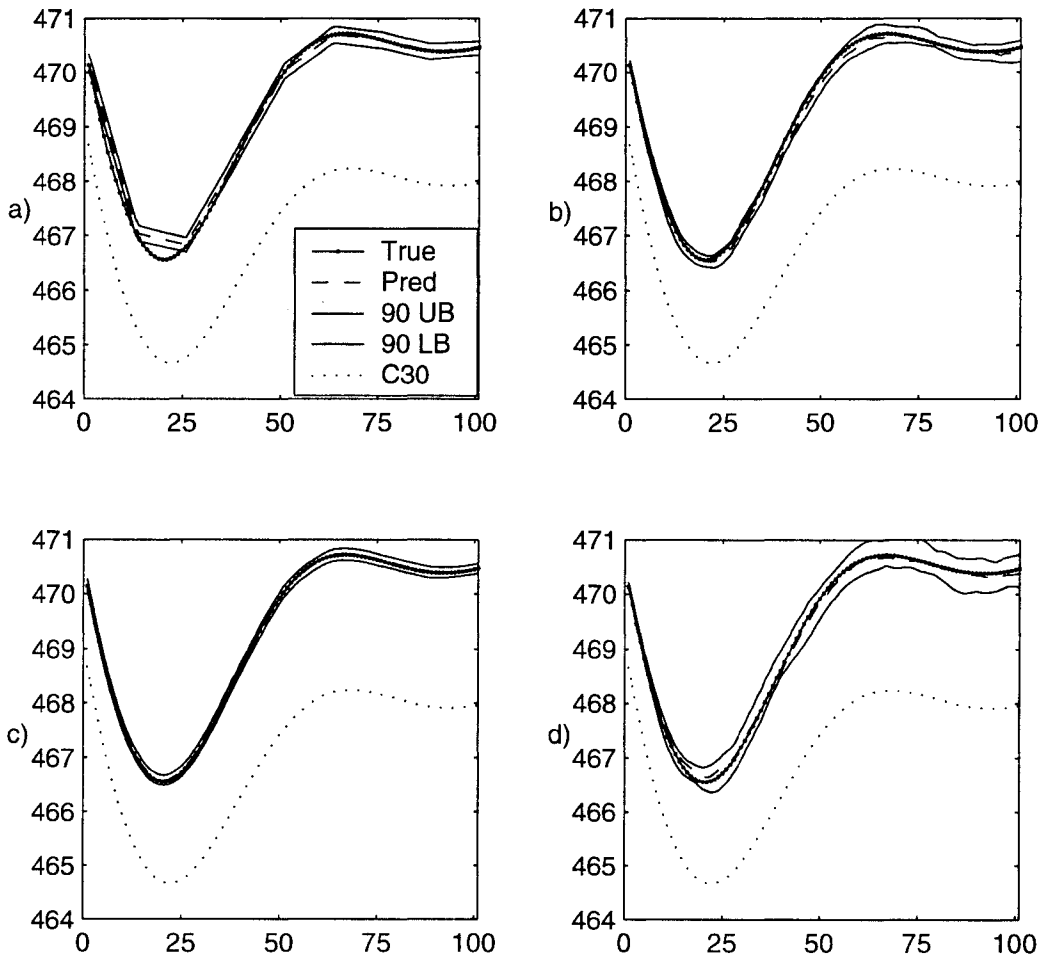

Figure 3.4 Predicted and true h time series at spatial location $(i, j) = (3, 15)$

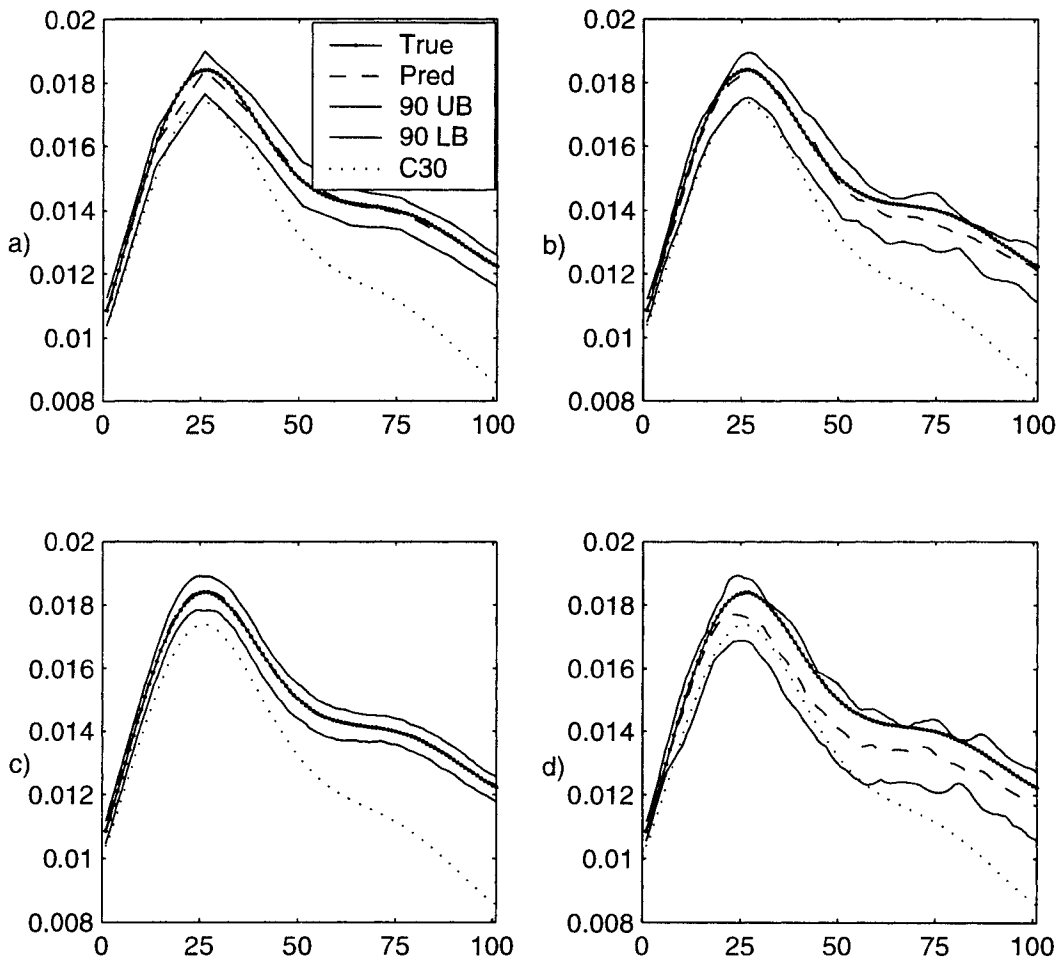

Figure 3.5 Predicted and true u time series at spatial location $(i, j) = (4, 15)$

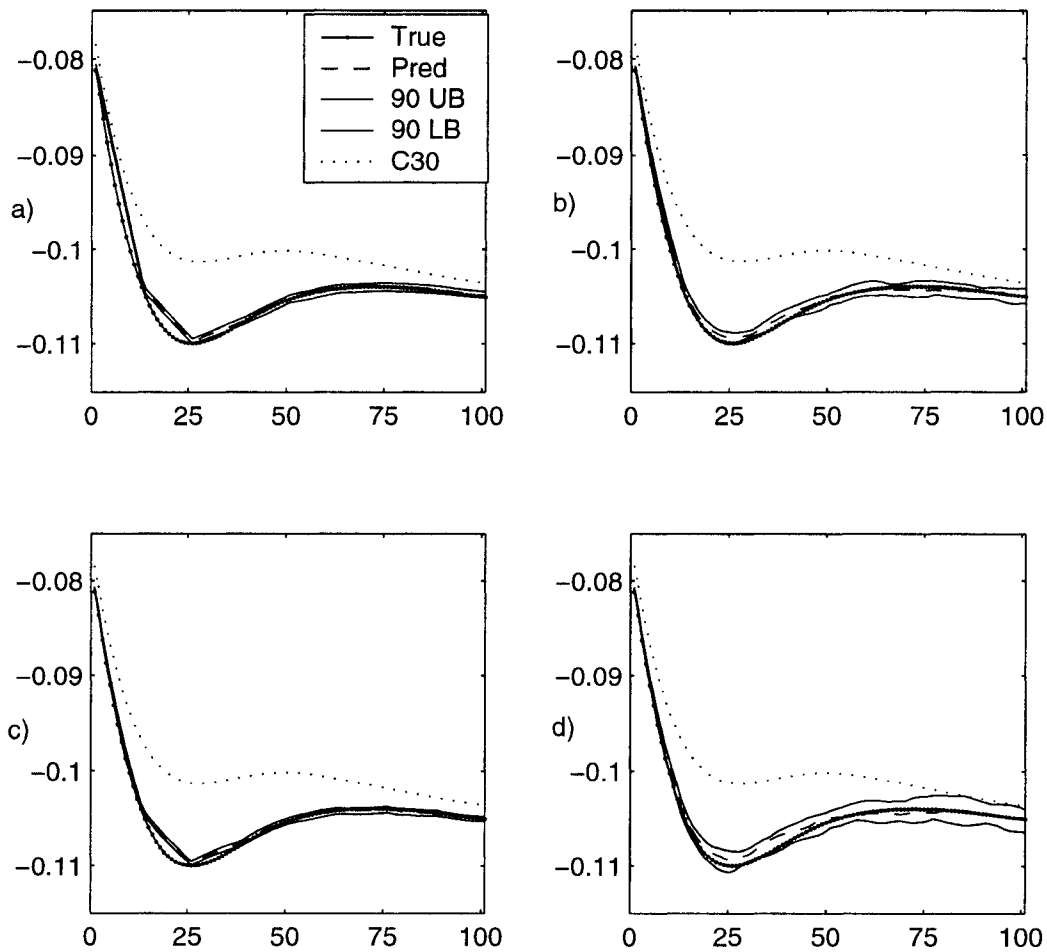

Figure 3.6 Predicted and true v time series at spatial location $(i, j) = (4, 9)$

CHAPTER 4. TWO ADDITIONAL CASES OF MULTIVARIATE COMPUTER OUTPUT STATISTICAL ANALYSES: INTRODUCTORY IDEAS

4.1 A statistical pilot study for characterizing the uncertainty in scaling AOGCM results

There is a continuing scientific debate about the extent to which human activities influence the climate. An important tool used in computational investigations is the *climate scenario.* According to the IPCC 2001 report, a climate scenario is a "plausible representation of future climate that has been constructed for explicit use in investigating the potential impacts of anthropogenic climate change". While there are various methods for constructing climate scenarios, the most commonly used are based on output from a coupled Atmosphere-Ocean General Circulation Model (AOGCM) which is based on several sub-models corresponding to atmosphere, ocean, ice, etc, that are coupled to form a meaningful model of the global climate. Mathematically, the model is a large, nonlinear system of partial differential equations which are solved numerically. Solving the system requires a massive computational effort and only a few institutions in the world are equipped to run AOGCMs. Even on high performance computers configured for this specific task, the average computational time is measured in weeks. Another limitation of the AOGCMs is that the spatial resolution used is still too coarse (hundreds of kilometers) for explaining local effects. Moreover, there is uncertainty from various sources that accompanies the solution. For example, the forces used in the model are not perfectly known and the errors associated with them propagate through the system. Another source of uncertainty originates in the feedbacks from processes that are not resolved by the model (e.g. thunderstorms) but have an important role in the large scale motion of the

system. The IPCC 2001 report gives an extensive account of the various sources of uncertainty associated with the AOGCMs. One possible approach to reducing the uncertainty is to run a large set of AOGCMs that cover the range of the unknown parameters or quantities. However, because each individual run requires considerable computational effort, this is an expensive approach. Alternatively, one could obtain more runs from simplified global models which are faster than the AOGCMs and are intended to reproduce their large scale behavior, such as global mean temperatures. A recent approach combines a few AOGCM runs with such faster simple global models to predict AOGCM results at untried runs. In this section we evaluate the variability in the climate simulated by AOGCM experiments, by means of using a limited number of slow runs and a faster running simple model. In other applications the focus is on characterizing the variability that corresponds to various emission configurations of greenhouse gases (e.g. CO_2) and aerosols. However, the range of these emissions is quite large and it requires careful planning of the experiment to obtain the optimal effect with a small number of slow runs.

In the sequel, by an AOGCM *control run* we mean an AOGCM run without any CO_2 emission increase over the period of study. A more realistic AOGCM run is to assume an increase in the CO_2 emission over the period of study (most of it due probably to human activities). The CO_2 emission is greenhouse-gas forcing which is an input in the AOGCM model and influences output variables such as temperature and precipitation. At the initial time of this more realistic AOGCM run it is assumed that the level of CO_2 is the same as in the control run, then gradually the CO_2 level is increased over the time period of study. In order to study the effect of the initial conditions, it is customary to take "snapshots" at several time points in the control run and use them as initial conditions in the AOGCM runs that assume increase in the CO_2. These runs then form an "ensemble" (Cubasch et al. 1994, Mitchell et al. 1999). The elements of an ensemble are conceptually similar to statistical replicates, and several such runs provide an empirical assessment of the variability due specifically to the initial conditions. This study is concerned with modeling statistically the elements of an ensemble.

There are several versions of AOGCMs in use today throughout the world. The AOGCM used in this study is the NCAR Community Climate System model. The ensemble consists of four elements with initial conditions determined by snapshots at four different times of the control run (which is not a member of the ensemble). The four times were selected to be sufficiently far apart to ensure "uncorrelation" between the snapshots. Each member of the ensemble is the spatio-temporal temperature output from the model forced with 1% per year increase in the CO_2 over the period of study, which is 70 years starting from 2002. In fact, only the 70 summer averages of output temperature at a 16×16 pixel region covering the western part of North America and a small part of the Pacific ocean were retained for analysis.

Pattern-scaling is a method of combining a few AOGCM slow runs with information provided by simpler models. Mitchell et al. (1999) suggest the use of predictors of separable form $T^*(x, y, t) = \overline{T}(t)P(x, y)$, where x is the longitude coordinate, y is the latitude and t is time, so that T^* is a predictor for the actual AOGCM output temperature. The global mean temperature $\overline{T}$ is obtained from faster running, simplified climate models and $P(x, y)$ is a scaling pattern estimated from the slow AOGCM runs.

Here we shall modify the above approximation in a statistical model which will provide a way of characterizing the variability associated with the prediction. In particular, the covariance function that describes the spatial structure of the errors from pattern scaling will be found using a nonstationary model applied to the ensemble results. Let $r = 1, ..., n$ be the n slow AOGCM runs. Then $T^*(x, y, t, r) = \overline{T}(t, r)P(x, y) + \epsilon(x, y, t, r)$ is a statistical model that could be used for this purpose. In our case the ensemble has 4 members of which 3 are used for the estimation of the scale pattern and the 4^{th} is used for prediction to test the model. Thus, $n = 3$. The $16 \times 16 \times 70 \times 3$ four dimensional array of data corresponding to (x, y, t, r) will be reshaped in a vector format, with x the most inner dimension and r the most outer. The above statistical model can be written in a more compact form as $\underline{T}^* = \overline{\tau}\underline{P} + \underline{\epsilon}$, with $\overline{\tau}$ the design matrix of size 16^2 by $16^2 \times 210$ and $\underline{P}$ the vector of regression parameters, of length 16^2. The OLS estimated regression parameters at each pixel (x, y) (after reshaping them back in a 16×16 matrix) is the actual estimated scaled pattern and is plotted in Figure 4.1a. The residuals $\hat{\underline{\epsilon}}$

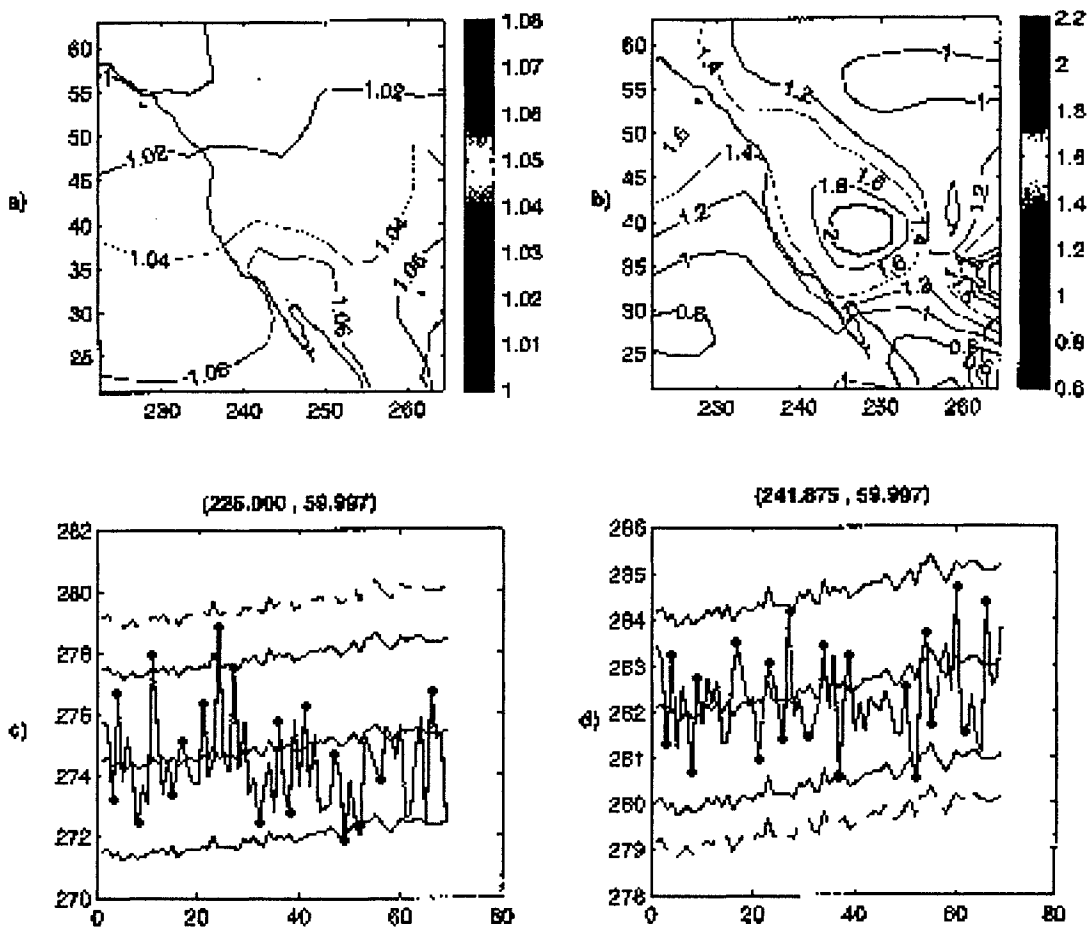

Figure 4.1 Estimated scale pattern, standard errors and prediction

appear to be uncorrelated in the t and r dimensions but correlated in the space dimensions x and y. Therefore we shall assume a block diagonal covariance matrix $I_{210} \otimes \Sigma_{256}$ for residuals, written for simplicity as a Kronecker product. It is convenient for our modeling purpose to reshape the vector of residuals in a matrix $Z_{i,j}$ of size 256×210, where the rows correspond to (x, y) and the columns correspond to (t, r) so that the columns are uncorrelated among each other, but the elements of individual columns are correlated. We shall fit a nonparametric and nonstationary model as described in Nychka et al. (2002) for the newly reshaped residuals Z. Thus one can consider that there are 210 replicates of a random vector Z of length 256 whose mean is $\underline{0}$ and covariance matrix is Σ. Probably the decomposition of Σ most often used is the eigenvalue decomposition, that is $\Sigma = \Psi D \Psi'$ with D the diagonal matrix of eigenvalues and Ψ the matrix of eigenvectors, which form a basis for the Euclidian space of dimension equal to the size of the covariance matrix. When the size of Σ is large, there are typically many small positive eigenvalues, so the determinant is close to zero and thus the covariance matrix is unstable to algebraic operations such as computing its inverse.

Nychka et al. (2002) suggest working with an alternative basis, such as a multiresolution wavelet. Under this basis, the inner matrix D in the decomposition of Σ is nondiagonal, but a large number of off-diagonal elements are typically "small". The idea is to enforce sparsity in the square root of the inner matrix by replacing these small off-diagonal elements with zero. More precisely, let $\hat{\Sigma} = \frac{1}{210} \sum_{i=1}^{210} Z_i Z_i'$ denote the sample estimator of Σ. Then the square root of the inner matrix in its eigenvalue decomposition is $\hat{H} = \hat{D}^{1/2} = (\Psi^{-1} \hat{\Sigma} \Psi'^{-1})^{1/2}$. The procedure then is to retain all diagonal elements of $\hat{H}$ and set to zero all off diagonal elements with absolute value below a prescribed threshold, such as the 98% quantile for the absolute value of the elements of $\hat{H}$. Denote by $\tilde{H}$ this sparse approximation of $\hat{H}$. Then $\tilde{\Sigma} = \Psi \tilde{H}^2 \Psi'$ is an estimator of Σ which is more stable to algebraic operations. The square root of the diagonal elements of $\tilde{\Sigma}$ are the estimated marginal standard errors at each spatial location (after reshaping them in a 16×16 matrix) and are plotted in Figure 4.1b. Since the errors are correlated, one should use a generalized least squares (GLS) criterion instead of the OLS. Although not done here (but possibly in a future draft), a more statistically correct

analysis would require reestimating the GLS regression parameters (called scaled pattern in this application) as $\underline{\hat{P}} = [\overline{r}'(I \otimes \hat{\Sigma}^{-1})\overline{r}]^{-1}[\overline{r}'(I \otimes \hat{\Sigma}^{-1})\underline{T}^*]$. Then the new GLS residuals would be treated as the previous OLS residuals in order to reestimate Σ by the method of Nychka et al. (2002). This iterative procedure would continue until convergence in the estimated scale pattern and covariance matrix is achieved.

Figures 4.1c and 4.1d show results for the 4^{th} member of the ensemble: the global mean temperature, AOGCM temperature and prediction at two pixels for which the estimated scale pattern is above and below 1 respectively. In both cases the global mean temperature is outside the 95% prediction bounds. This shows evidence that the statistical model presented here, although simple, could discriminate between the global mean temperature and pixel-wise temperature time series. However, for this pilot study, the variability associated with the estimated parameters is not included in the prediction bands. The parameters were simply replaced by their estimates in the regression prediction. A more elaborate study requires including the variability associated with the estimated scale pattern as well as the variability associated with the pixel-wise standard errors estimates resulted from the nonparametric nonstationary model presented above.

4.2 Stochastic closure for multi-scale simulations

This section outlines a statistical method for predicting fine grid multivariate output restricted on a coarser grid, when coarse grid numerical solutions are available. The method is illustrated through an example based on an ocean model that appears in Greatbatch and Nadiga (2000) and, if successful, will be applied to other models of interest. The mathematical form of this ocean model is

(4.1) $\frac{\partial q}{\partial t} + J(\psi, q) = F - D,$

with ψ the velocity streamfunction, $q = \nabla^2\psi + \beta y = \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} + \beta y$ the potential vorticity and $J(\psi, q) = -\frac{\partial\psi}{\partial y}\frac{\partial q}{\partial x} + \frac{\partial\psi}{\partial x}\frac{\partial q}{\partial y}$ the Jacobian of the these functions. The sinusoidal wind force F and the dissipation D are described in the referenced paper. The function of interest in this study is the velocity streamfunction ψ. After boundary and initial conditions are imposed, the model

(4.1) is implemented in a finite difference scheme, with an adaptive, changing time step. Two spatial grids are considered: a fine grid of 101×201 points and a coarse grid of 26×51 time points. The fine grid leads to a slower running, more accurate code than the coarser grid. The fine grid run takes about 7.5 hours in FORTRAN on a 1GB RAM, 400MHz 64-bit processor speed workstation whereas the coarse grid takes about 22min. The problem here is to find a statistical model that runs on the coarse grid and predicts the fine grid output at the coarse grid points. This is a part a larger problem (described below) dealing with the study of local effects.

Suppose we want to run a fine scale model because we want to study regional effects. The "brute force" approach would be to actually run the fine scale model, obtain output and look at the region of interest. This would require large amounts of computational resources. Instead, one could sample a number of smaller spatial regions and run the fine scale model for each of these regions separately (after necessarily adapting the boundary conditions). Since these are smaller regions relative to the whole domain, the fine grid model run for these regions would be faster than the global model run with the same grid size, but would still require a considerable amount of computational resources. The task is to build a statistical model that would relate the fine and coarse grid output for each of these regions. The statistical parameters estimates are based on the whole sample of spatial regions. If one is interested in local effects for a certain region included in the larger domain, then this becomes a statistical prediction problem for that new region, based on the sampled regions. In what follows, we only describe a statistical model that would relate fine and coarse grid output for a certain region. We do not develop the full study involving sampled smaller regions. As a first step in this direction, we shall develop a statistical model that would relate fine and coarse grid output on the whole domain of the ocean model. The assumption then is that a similar statistical model could be developed for any subregion of the domain.

Let Y be a stochastic process defined as $Y_{i,j,t+1} = \rho_t Y_{i,j,t} + \sigma\epsilon_{i,j,t}\sqrt{1-\rho_t{}^2}$, with $\rho_t = e^{-\frac{dt}{\alpha}}$, where dt is the adaptive coarse temporal increment, i and j are indices corresponding to a 3×3 moving window over the 26×51 coarse grid and ϵ are independent standard normal random

variables. This process is considered a stochastic kernel defined over a 3×3 window. The coarse grid numerical model produces a numerical approximation $\tilde{J}(i,j,t)$ of the Jacobian $J(i,j,t)$. Denote by $\hat{J}$ the stochastic process (called stochastic Jacobian in the sequel) resulting from the convolution of the coarse numerical Jacobian and the stochastic kernel Y defined above. More precisely, $\hat{J}_{i,j,t} = \frac{1}{9}\sum_{k,l=\{-1,0,1\}} \tilde{J}(i+l,j+k,t)Y_{i+l,j+k,t}$. The coarse numerical model runs with this stochastic Jacobian $\hat{J}$ replacing the coarse grid numerical Jacobian $\tilde{J}$, and a stochastic velocity streamfunction $\hat{\psi}_{i,j,t}$ is obtained at the end of the run. The hope is that this stochastic velocity streamfunction is a good prediction of the fine grid velocity streamfunction restricted at the coarse grid. The statistical task was to find estimates for the parameters α and σ appearing in the expression of Y, that provide the best fit for the statistical model $\hat{\psi}$. The main difficulty is that the temporal dimension of the problem is very large and, due to the non-linearity of the problem, likelihood or least-squares numerical optimization would require a run of the stochastic coarse numerical model at each step. Therefore, an iterative numerical optimization requires large amounts of computational resources itself. Our preliminary approach was to compute the root mean square error (RMSE) between the fine grid velocity streamfunction ψ (restricted at the coarse grid) and its statistical predictor $\hat{\psi}$ on a grid of values (suggested by physicists) for the statistical parameters. Then declare as estimates that combination of statistical parameters that minimize RMSE. Data and predictions were retained at each 100^{th} coarse grid time point. Due to the smoothness of the time series involved, the RMSEs based on the whole time series and those based on data sampled at each 100^{th} time point will not differ too much. As an aside, since the numerical scheme is time adaptive, perhaps we need to change this sampling scheme to a more adaptive one, by using the time step as an indicator. For example, if the time step becomes small in a temporal sub-interval, then we should probably sample more time points there.

We considered two RMSE measures. The first involves individual time points and is defined as $RMSE_T = \sqrt{\frac{1}{R}\sum_{r=1}^{R}[\frac{1}{26*51*N_T}\sum_{i,j,t}(\psi_{i,j,t} - \hat{\psi}_{i,j,t,r})^2]}$, where $N_T = 40$ is the number of retained time points and R is the number of prediction simulations. The second measure involves averages over time and is defined as $RMSE_A = \sqrt{\frac{1}{R}\sum_{r=1}^{R}[\frac{1}{26*51}\sum_{i,j}(\overline{\psi}_{i,j} - \overline{\hat{\psi}}_{i,j,r})^2]}$.

We considered $R = 2$ for computational convenience, and the results are presented in Table 4.1. The best agreement when time averages are involved corresponds to $\alpha = 0.046$ and $\sigma = 1.2$. The estimated values obtained for the time averaged RMSE measures are in agreement with the expectations of the physicists involved. It was surprising, however, to see that the estimates were different for RMSE involving time sequences. It is not clear whether this change in estimates reflects a real qualitative difference between the stochastic output and its time average, or is an artifact due to small R and/or small N_T. One could increase R and N_T, but this would require additional computational cost and storage space for the larger data sets.

Table 4.1 RMSE measures

$RMSE_T$ ($RMSE_A$)	$\alpha = 0.375$	$\alpha = 0.187$	$\alpha = 0.046$
$\sigma = 0.1$	0.9151 (0.4036)	0.9146 (0.4019)	0.9114 (0.4012)
$\sigma = 0.4$	0.9185 (0.4071)	0.9189 (0.4005)	0.9220 (0.4122)
$\sigma = 0.9$	0.9832 (0.3305)	0.9444 (0.3365)	0.9444 (0.3425)
$\sigma = 1.2$	1.6182 (0.4667)	1.3839 (0.3668)	1.0991 (0.3053)
$\sigma = 1.8$	7.7810 (3.8878)	8.1109 (3.9034)	3.8200 (1.8035)

APPENDIX A. PROOF OF ALMOST SURE CONVERGENCE FOR THE TER STATISTICAL MODEL

A.1 No auxiliary information

Denote $\Gamma = C_D \otimes C_2$ of size $q = 2D$. Then the MLE of τ^2 is

$$\hat{\tau}^2 = \frac{1}{2MD}(\underline{T}^Y - \hat{\mu}\underline{1})'[C_D^{-1} \otimes (C_2^{-1} \otimes I_M)](\underline{T}^Y - \hat{\mu}\underline{1}) =$$

$$\frac{1}{Mq}(\underline{T}^Y - \hat{\mu}\underline{1})'(\Gamma^{-1} \otimes I_M)(\underline{T}^Y - \hat{\mu}\underline{1}) = \frac{1}{Mq}\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}[\sum_{t=1}^{M}(T_{i,t}^{Y\prime} - \hat{\mu})(T_{j,t}^{Y} - \hat{\mu})],$$

with $\hat{\mu} = \frac{\underline{1}'[\Sigma]^{-1}\underline{T}^Y}{\underline{1}'[\Sigma]^{-1}\underline{1}}$. From section 2.2.3 it follows that the exact local truncation errors appearing in (2.4) are of order h_c^2. Consider the limit as h_c approaches zero. Since $0 < h_f < h_c$, then h_f converges to zero also, which implies $T^Y - T$ converges to 0. Thus T^Y is asymptotically of order h_c^2 also. The MLE of μ is

$$\hat{\mu} = \frac{\underline{1}'[\Sigma]^{-1}\underline{T}^Y}{\underline{1}'[\Sigma]^{-1}\underline{1}} = \frac{\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{T}_{j,.}^Y)}{\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{1})} = \frac{MO(h_c^2)\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}}{M\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}} = O(h_c^2),$$

where the 'dot' stands for the time series as a vector. To see this more clearly, the absolute value of an arbitrary unweighted local truncation error term in the numerator can be written as

$$|\underline{1}'\underline{T}_{j,.}^Y| = |\sum_{k=1}^{M}T_{j,k}^Y| \leq \sum_{k=1}^{M}|T_{j,k}^Y| \leq h_c^2\sum_{k=1}^{M}(Upp) = MO(h_c^2)$$

where Upp is the upper bound of all second order derivatives of the exact solutions of (2.1). Here the local truncation error has been approximated by its leading, second order term in the remainder of the Taylor series (2.5). It follows

$$\hat{\tau}^2 = \frac{MO(h_c^4)\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}}{Mq} = O(h_c^4),$$

which implies $\hat{\tau} = O(h_c^2)$. Table A.1(a) shows an asymptotic study for $\hat{\tau}$ as h_c becomes smaller. Notice that $\hat{\tau}/h_c$ seems to converge to zero, $\hat{\tau}/h_c^2$ seems to converge to a non-zero constant and $\hat{\tau}/h_c^3$ appears to grow unboundedly. Thus, there is empirical evidence that $\hat{\tau}$ behaves as h_c^2, indeed. The marginal distributions of $\tilde{T}^{1,Y}_{d,t_{i+1}}$ and $\tilde{T}^{2,Y}_{d,t_{i+1}}$ appearing in (2.7) are standard normals of mean $O(h_c^2)$ and variance $O(h_c^4)$. The relationship (2.7) can be rewritten as

(A.1a) $\tilde{Y}^1_{d,t_{i+1}} = \tilde{Y}^1_{d,t_i} + h_c(\frac{r_1}{K_1}\tilde{Y}^1_{d,t_i}(K_1 - \tilde{Y}^1_{d,t_i}) - \frac{r_1\alpha_{12}}{K_1}\tilde{Y}^1_{d,t_i}\tilde{Y}^2_{d,t_i}) + O(h_c^2) + O(h_c^2)[Z]^1_{d,t_i}$

(A.1b) $\tilde{Y}^2_{d,t_{i+1}} = \tilde{Y}^2_{d,t_i} + h_c(\frac{r_2}{K_2}\tilde{Y}^2_{d,t_i}(K_2 - \tilde{Y}^2_{d,t_i}) - \frac{r_2\alpha_{21}}{K_2}\tilde{Y}^2_{d,t_i}\tilde{Y}^1_{d,t_i}) + O(h_c^2) + O(h_c^2)[Z]^2_{d,t_i}.$

for standard normal random variables $[Z]^1_{d,t_i}$ and $[Z]^2_{d,t_i}$. Notice that the statistically predicted local truncation errors $\tilde{T}^{1,Y}_{d,t_{i+1}}$ and $\tilde{T}^{2,Y}_{d,t_{i+1}}$ in (2.7) are of the same order of magnitude, i.e. $O(h_c^2)$, as the fine numerical local truncation errors in (2.6). The relationship (A.1) can be rewritten as

(A.2a) $\sum_{j=0}^{1}\alpha_j\tilde{Y}^1_{d,t_{i+j}} = h_c\phi_1(t_i, \tilde{Y}^1_{d,t_i}, \tilde{Y}^2_{d,t_i}, h_c)$

(A.2b) $\sum_{j=0}^{1}\alpha_j\tilde{Y}^2_{d,t_{i+j}} = h_c\phi_2(t_i, \tilde{Y}^1_{d,t_i}, \tilde{Y}^2_{d,t_i}, h_c),$

with $\alpha_0 = -1$, $\alpha_1 = 1$,

$$\phi_1(t_i, \tilde{Y}^1_{d,t_i}, \tilde{Y}^2_{d,t_i}, h_c) = (\frac{r_1}{K_1}\tilde{Y}^1_{d,t_i}(K_1 - \tilde{Y}^1_{d,t_i}) - \frac{r_1\alpha_{12}}{K_1}\tilde{Y}^1_{d,t_i}\tilde{Y}^2_{d,t_i}) + O(h_c) + O(h_c)[Z]^1_{d,t_i}$$

$$\phi_2(t_i, \tilde{Y}^1_{d,t_i}, \tilde{Y}^2_{d,t_i}, h_c) = (\frac{r_2}{K_2}\tilde{Y}^2_{d,t_i}(K_2 - \tilde{Y}^2_{d,t_i}) - \frac{r_2\alpha_{21}}{K_2}\tilde{Y}^2_{d,t_i}\tilde{Y}^1_{d,t_i}) + O(h_c) + O(h_c)[Z]^2_{d,t_i}.$$

In a more compact vector form the relationship (A.2) can be written as

(A.2') $\sum_{j=0}^{1}\alpha_j\underline{\tilde{Y}}_{d,t_{i+j}} = h_c\underline{\phi}(t_i, \underline{\tilde{Y}}_{d,t_i}, h_c),$

where the vectors $\underline{\tilde{Y}}$ and $\underline{\phi}$ are two dimensional.

Next, we shall use a fundamental result stating that the numerical solution is convergent if and only if the finite difference scheme is consistent and zero-stable. The reader can consult Lambert (1991) ,*Theorem 2.2*, for more details. Loosely said, the finite difference scheme is consistent if its local truncation error divided by h_c converges to zero as h_c converges to zero and the finite difference scheme is zero-stable if small perturbations of the finite difference scheme result in small perturbations of its numerical solution. A consistency criterion is: $\sum_{j=0}^{1}\alpha_j = 0$ and $\underline{\phi}(t, \underline{N}(t), 0)/\sum_{j=0}^{1}(j\alpha_j) = \underline{f}(t, \underline{N}(t))$, with $\underline{f}$ defined in (2.1'). The first consistency condition is obviously satisfied. The second consistency condition is satisfied since if $h_c = 0$ then the stochastic terms $O(h_c)[Z]^1_{d,t_i}$ and $O(h_c)[Z]^2_{d,t_i}$ are identically zero, except perhaps for

a set of zero probability. Finally, the method is zero-stable if it satisfies the root condition, which in this case is equivalent to the fact that the only root of the characteristic polynomial $\sum_{j=0}^{1}\alpha_j x^j$ is on the unit circle. Table A.2(a) shows empirical evidence of convergence in mean square since the RMSE measures appear to converge to zero as h_c converges to zero. However, this mean square convergence is weaker than the strong convergence and therefore Table A.2(a) does not show direct empirical evidence of strong convergence.

A.2 Auxiliary information used

Denote by h_C an intermediate time-step so that $0 < h_f < h_C < h_c$ and let X be the corresponding numerical solution to (2.1). Denote $\underline{T}^X$ the vector of local truncation errors obtained as in (2.6), with X replacing Y. Also, it will be assumed that T^X is of order h_c^2 when h_c (and h_C) converges to 0. The proof will be similar to that in section A.1, although some steps require further attention. The MLE of $\underline{\mu} = [\mu_0, \mu_1]$ is

$$\underline{\hat{\mu}} = ([\underline{1}, \underline{T}^X]'[\Sigma]^{-1}[\underline{1}, \underline{T}^X])^{-1}([\underline{1}, \underline{T}^X]'[\Sigma]^{-1}\underline{T}^Y),$$

assuming that the inverse matrix exists. Writing $\Sigma = \Gamma \otimes I_M$ it can be shown that

$$[\underline{1}, \underline{T}^X]'[\Sigma]^{-1}[\underline{1}, \underline{T}^X] = \begin{pmatrix} \sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{1}) & \sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{T}_{j,.}^X) \\ \sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{T}_{i,.}^{X'}\underline{1}) & \sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{T}_{i,.}^{X'}\underline{T}_{j,.}^X) \end{pmatrix}$$

and

$$[\underline{1}, \underline{T}^X]'[\Sigma]^{-1}\underline{T}^Y = \begin{pmatrix} \sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{T}_{j,.}^Y) \\ \sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{T}_{i,.}^{X'}\underline{T}_{j,.}^Y) \end{pmatrix}.$$

Then

$$[\hat{\mu}_0, \hat{\mu}_1]' = \frac{1}{\det([\underline{1}, \underline{T}^X]'[\Sigma]^{-1}[\underline{1}, \underline{T}^X])} *$$

$$\begin{pmatrix} [\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{T}_{i,.}^{X'}\underline{T}_{j,.}^X)][\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{T}_{j,.}^Y)] - [\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{T}_{i,.}^{X'}\underline{1}][\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{T}_{i,.}^{X'}\underline{T}_{j,.}^Y)] \\ -[\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{T}_{j,.}^X)][\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{T}_{j,.}^Y)] + [\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{1})][\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{T}_{i,.}^{X'}\underline{T}_{j,.}^Y)] \end{pmatrix},$$

where the determinant is

$$\det([\underline{1}, \underline{T}^X]'[\Sigma]^{-1}[\underline{1}, \underline{T}^X]) =$$

$$[\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{1})][\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{T}_{i,\cdot}^{X'}\underline{T}_{j,\cdot}^{X})]-[\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{T}_{j,\cdot}^{X})][\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{T}_{i,\cdot}^{X'}\underline{1})]=$$

$$[\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{1})][\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{T}_{i,\cdot}^{X'}\underline{T}_{j,\cdot}^{X})]-[\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{T}_{j,\cdot}^{X})]^2.$$

It will be shown at the end of this proof that $\det([\underline{1},\underline{T}^X]'[\Sigma]^{-1}[\underline{1},\underline{T}^X]) = M^2h_c^4A(h_c)$, where $A(h_c)$ converges to a strictly positive constant as h_c converges to zero. Then

$$\hat{\mu}_0=\frac{1}{M^2h_c^4A(h_c)}[M^2O(h_c^6)+M^2O(h_c^6)]=O(h_c^2)$$

and

$$\hat{\mu}_1=\frac{1}{M^2h_c^4A(h_c)}[M^2O(h_c^4)+M^2O(h_c^4)]=O(1).$$

Finally, the MLE of τ^2 is

$$\hat{\tau}^2=\frac{1}{Mq}(\underline{T}^Y-\hat{\mu}_0\underline{1}-\hat{\mu}_1\underline{T}^X)'(\Gamma^{-1}\otimes I_M)(\underline{T}^Y-\hat{\mu}_0\underline{1}-\hat{\mu}_1\underline{T}^X)=$$

$$\frac{1}{Mq}\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}[\sum_{t=1}^{M}(T_{i,t}^Y-\hat{\mu}_0-\hat{\mu}_1T_{i,t}^X)(T_{j,t}^Y-\hat{\mu}_0-\hat{\mu}_1T_{j,t}^X)]=\frac{MO(h_c^4)\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}}{Mq}=O(h_c^4),$$

which implies $\hat{\tau}=O(h_c^2)$. This can also be seen empirically in Table A.1(b), where the intermediate step size $h_C=\frac{L}{15*2^5}$ has been used. Next, a similar argument as in section A.1 can be used to prove almost sure convergence of the statistical simulations to the exact solution. Empirical evidence of this convergence is shown in Table A.2(b). The reason these SRMSE measures are much smaller than the ones presented in Table 2.1 is that the intermediary grid size numerical solution X is quite informative here since h_C is small.

We close this proof by showing that $\det([\underline{1},\underline{T}^X]'[\Sigma]^{-1}[\underline{1},\underline{T}^X]) = M^2h_c^4A(h_c)$, where $A(h_c)$ converges to a strictly positive constant as $h_c\to 0$. Indeed,

$$\det([\underline{1},\underline{T}^X]'[\Sigma]^{-1}[\underline{1},\underline{T}^X])=[\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{1})][\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{T}_{i,\cdot}^{X'}\underline{T}_{j,\cdot}^{X})]-[\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\underline{1}'\underline{T}_{j,\cdot}^{X})]^2=$$

$$\frac{h_c^4}{4}M^2\{[\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}][\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\frac{1}{M}\sum_{k=1}^{M}N_j''(t_k)N_i''(t_k))]-[\sum_{i,j=1}^{q}\Gamma_{i,j}^{-1}(\frac{1}{M}\sum_{k=1}^{M}N_j''(t_k))]^2\},$$

where the indices i,j are combinations of species and input vector indices. The expressions of the local truncation errors as they appear in the relationship (2.5) have been used in this equality, with the reminders R_1 and R_2 being neglected. Denote by $A(h_c)$ the last factor in the

expression of the determinant. Then, as $M \to \infty$ (or, equivalently, $h_c \to 0$), $A(h_c)$ converges to

$$[\sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}][\sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}(\frac{1}{L}\int_0^L N_j''(t)N_i''(t)dt)] - [\sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}(\frac{1}{L}\int_0^L N_j''(t)dt)]^2.$$

Denote by $\underline{N}''(t)$ the vector of second order derivatives at an arbitrary point in $[0, L]$. This vector has length q and its components are $N_i''(t)$, $i = 1, ..., q$.

The 2×2 matrix $[\underline{1}, \underline{N}''(t)]'\Gamma^{-1}[\underline{1}, \underline{N}''(t)]$ is non-negative definite because

$$a'[\underline{1}, \underline{N}''(t)]'\Gamma^{-1}[\underline{1}, \underline{N}''(t)]a = (\Gamma^{-1/2}[\underline{1}, \underline{N}''(t)]a)'(\Gamma^{-1/2}[\underline{1}, \underline{N}''(t)]a) \geq 0$$

for any 2×1 vector a. Then, for any $t \in [0, L]$,

$$[\sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}][\sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}N_j''(t)N_i''(t)] - [\sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}N_j''(t)]^2 = \det([\underline{1}, \underline{N}''(t)]'\Gamma^{-1}[\underline{1}, \underline{N}''(t)]) \geq 0.$$

Rewriting and integrating this inequality on $[0, L]$, one obtains

$$[\sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}][\sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}(\frac{1}{L}\int_0^L N_j''(t)N_i''(t)dt)] \geq \frac{1}{L}\int_0^L [\sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}N_j''(t)]^2 dt.$$

Finally, from the Schwartz inequality for the function $\sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}N_j''(t)$, one has

$$\frac{1}{L}\int_0^L [\sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}N_j''(t)]^2 dt > [\frac{1}{L}\int_0^L \sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}N_j''(t)dt]^2 = [\sum_{i,j=1}^{q} \Gamma_{i,j}^{-1}(\frac{1}{L}\int_0^L N_j''(t)dt)]^2.$$

The inequality is strict because the function is not constant.

Table A.1 Asymptotic study of the noise standard error

(a) h_c	0.2000	0.1000	0.0500	0.0250
$\hat{\tau}/h_c$	0.2005	0.1023	0.0513	0.0256
$\hat{\tau}/h_c^2$	1.0027	1.0230	1.0257	1.0220
$\hat{\tau}/h_c^3$	5.0137	10.2298	20.5134	40.8816
(b) h_c	0.2000	0.1000	0.0500	0.0250
$\hat{\tau}/h_c$	0.0070	0.0038	0.0019	0.0010
$\hat{\tau}/h_c^2$	0.0349	0.0380	0.0387	0.0387
$\hat{\tau}/h_c^3$	0.1746	0.3802	0.7743	1.5498

Table A.2 Asymptotic study of convergence

(a) h_c	0.2000	0.1000	0.0500	0.0250
RMSE(N_1)	0.1069	0.0299	0.0213	0.0117
SRMSE(N_1)	0.1163	0.0448	0.0244	0.0138
RMSE(N_2)	0.1342	0.0406	0.0244	0.0129
SRMSE(N_2)	0.1355	0.0584	0.0286	0.0138
(b) h_c	0.2000	0.1000	0.0500	0.0250
RMSE(N_1)	0.0026	0.0013	0.0013	0.0008
SRMSE(N_1)	0.0041	0.0025	0.0017	0.0009
RMSE(N_2)	0.0024	0.0014	0.0011	0.0006
SRMSE(N_2)	0.0035	0.0022	0.0015	0.0007

APPENDIX B. DERIVATION OF THE DISCRETE-TIME BROWNIAN BRIDGE RELATIONSHIP

In this Appendix we shall derive an explicit, iterative expression for the DBB constrained to pass through two prespecified, fixed ends. We also show that the marginal means lie on the straight line connecting the two end points and that the marginal variances are quadratic in time. Let X be the autoregression $X_{i+1} = X_i + \sigma\epsilon_{i+1}$, with $i = 1, ..., N$. We shall consider that N, X_1 and X_N are fixed constants. The method of derivation is based on rewriting conveniently the likelihood of the autoregression and discarding successively proportionality constants. In the sequel, "$\propto$" signifies "proportional to". The likelihood of the above autoregression is

$L(X_1, X_2, ..., X_{N-1}, X_N|\sigma) = (\frac{1}{\sqrt{2\pi\sigma^2}})^N \exp\{-\frac{1}{2\sigma^2}\sum_{i=2}^{N}(X_i - X_{i-1})^2\}$

$\propto \exp\{-\frac{1}{\sigma^2}(X_{N-1}^2 + X_{N-2}^2 + X_{N-3}^2 + ... + X_3^2 + X_2^2 - X_{N-1}X_N - X_{N-2}X_{N-1} - ... - X_1X_2)\}$

$= \exp\{-\frac{1}{\sigma^2}[X_{N-1}^2 - X_{N-1}(X_N + X_{N-2}) + X_{N-2}^2 + X_{N-3}^2 + ... + X_3^2 + X_2^2 - X_{N-3}X_{N-2} - ... - X_1X_2]\} = \exp\{-\frac{1}{\sigma^2}[[X_{N-1} - (\frac{X_N}{2} + \frac{X_{N-2}}{2})]^2 - \frac{1}{4}(X_N + X_{N-2})^2 + X_{N-2}^2 + X_{N-3}^2 + ... + X_3^2 + X_2^2 - X_{N-3}X_{N-2} - ... - X_1X_2]\} \propto \exp\{-\frac{1}{\sigma^2}[[X_{N-1} - (\frac{X_N}{2} + \frac{X_{N-2}}{2})]^2 + \frac{3}{4}X_{N-2}^2 - X_{N-2}(\frac{X_N}{2} + X_{N-3}) + X_{N-3}^2 + ... + X_3^2 + X_2^2 - X_{N-4}X_{N-3} - ... - X_1X_2]\} = \exp\{-\frac{1}{\sigma^2}[[X_{N-1} - (\frac{X_N}{2} + \frac{X_{N-2}}{2})]^2 + \frac{3}{4}[X_{N-2} - (\frac{X_N}{3} + \frac{2X_{N-3}}{3})]^2 - \frac{3}{4}(\frac{X_N}{3} + \frac{2X_{N-3}}{3})^2 + X_{N-3}^2 + ... + X_3^2 + X_2^2 - X_{N-4}X_{N-3} - ... - X_1X_2]\} \propto \exp\{-\frac{1}{\sigma^2}[\frac{2}{2}[X_{N-1} - (\frac{X_N}{2} + \frac{X_{N-2}}{2})]^2 + \frac{3}{4}[X_{N-2} - (\frac{X_N}{3} + \frac{2X_{N-3}}{3})]^2 + \frac{4}{6}[X_{N-3} - (\frac{X_N}{4} + \frac{3X_{N-4}}{4})]^2 - \frac{4}{6}(\frac{X_N}{4} + \frac{3X_{N-4}}{4})^2 + X_{N-4}^2 + ... + X_3^2 + X_2^2 - X_{N-5}X_{N-4} - ... - X_1X_2]\} \propto ... \propto \exp\{-\frac{1}{\sigma^2}\sum_{i=N-1}^{2}\frac{N-i+1}{2(N-i)}[X_i - (\frac{1}{N-i+1}X_N + \frac{N-i}{N-i+1}X_{i-1})]^2\} = \exp\{-\frac{1}{2\sigma^2}\sum_{i=2}^{N-1}\frac{N-i+1}{N-i}[X_i - (\frac{1}{N-i+1}X_N + \frac{N-i}{N-i+1}X_{i-1})]^2\}$.

Since X_1 and X_N are considered constants, it follows that the last expression is proportional to $L(X_2, ..., X_{N-1}|X_1, X_N, \sigma)$. Therefore this conditional likelihood implies that the iterative process

$$(*) \qquad X_i = \frac{1}{N-i+1}X_N + \frac{N-i}{N-i+1}X_{i-1} + \sigma\sqrt{\frac{N-i}{N-i+1}}\epsilon_i,$$

with ϵ_i independent and identically distributed $N(0,1)$ and $i = 2, ..., N-1$, provides an expression for the DBB constrained to pass through $(1, X_1)$ and (N, X_N).

Next we shall prove by induction that the marginal means of the DBB are situated on the straight line connecting $(1, X_1)$ and (N, X_N). Obviously, $(1, X_1)$ is on this straight line and the statement is true since $EX_1 = X_1$. Suppose $(i-1, EX_{i-1})$ is on the straight line. Then $EX_{i-1} = X_1 + (X_N - X_1)\frac{i-2}{N-1}$ which implies $EX_{i-1} = X_N\frac{i-2}{N-1} + X_1\frac{N-i+1}{N-1}$. Taking expectations in $(*)$, it follows that $EX_i = \frac{1}{N-i+1}X_N + \frac{N-i}{N-i+1}EX_{i-1} = \frac{1}{N-i+1}X_N + \frac{N-i}{N-i+1}(X_N\frac{i-2}{N-1} + X_1\frac{N-i+1}{N-1}) = X_N\frac{i-1}{N-1} + X_1\frac{N-i}{N-1}$, which proves that (i, EX_i) is also on the straight line.

We now derive the expression for the marginal variances. Denote $v_i = var(X_i)$. Then $v_i = (\frac{N-i}{N-i+1})^2 v_{i-1} + \sigma^2\frac{N-i}{N-i+1}$ which implies $v_i = \sigma^2[\frac{N-i}{N-i+1} + (N-i)^2\sum_{k=N-i+1}^{N-2}\frac{1}{k(k+1)}] + v_1(\frac{N-i}{N-1})^2$. Since $\frac{1}{k(k+1)} = \frac{1}{k} - \frac{1}{k+1}$ and $v_1 = 0$ it follows that $v_i = \sigma^2[\frac{N-i}{N-i+1} + (N-i)^2(\frac{1}{N-i+1} - \frac{1}{N-1})] = \sigma^2[\frac{N-i}{N-i+1} + \frac{(N-i)^2}{N-i+1} - \frac{(N-i)^2}{N-1}] = \sigma^2[(N-i) - \frac{(N-i)^2}{N-1}] = \sigma^2(i-1)(1-\frac{i-1}{N-1})$. It is clear that $v_N = 0$ and that the marginal variance is quadratic in $i-1$, attaining its maximum half way between 1 and N. These results allowed us to use DBB in chapter 3 as an ingredient in the locally linear approximation of smooth and large sample size time series.

We close this Appendix with a comment on the brownian bridge process. If x, y are real numbers and $L > 0$, then a brownian bridge process is a continuous time Gaussian process $X_t, 0 \le t \le L$ such that $X_0 = x$, $EX_t = x + (y-x)\frac{t}{L}$ and $\text{cov}(X_t, X_s) = min(s,t) - \frac{st}{L}$. It is easy to show (see also Borodin and Salminen 2002) that $X_L = y$ almost sure. Also, it is clear that the marginal means are linear in time and $\text{var}(X_t) = t(1-\frac{t}{L})$ are quadratic in time. A simple time discretization of such a continuous time brownian bridge is not very helpful for simulation purposes because of the large dimension of the resulting covariance matrix. Instead, the iterative process derived in this Appendix requires less computational effort. A similar discrete-time brownian bridge process has been used as an importance sampler by Durham and Gallant (2002). Their derivation, however, is based on a time discretization of a brownian bridge written as the solution of a stochastic differential equation, while the derivation in this Appendix is more elementary.

BIBLIOGRAPHY

Apel J.R. (1987). *Principles of Ocean Physics.* Academic Press.

Aslett, R., Buck, R.J., Duval, S.G., Sacks, J. and Welch, J.W. (1998). Circuit optimization via sequential computer experiments: design of an output buffer, *Applied Statist., 47*, 31-48.

Borodin A. N. and Salminen P. (2002) *Handbook of Brownian Motion - Facts and Formulae.* Second Edition, Birkhauser Verlag.

Craig, P.G., Goldstein, M., Rougier, J.C. and Seheult, A.H. (2001), Bayesian Forecasting for Complex Systems Using Computer Simulators, *J. Am. Statist. Assoc., 96*, 717-729.

Cubasch U., Santer B.D., Hellbach A., Hegerl G.C., Hock H., Maier-Reimer E., Mikolajewicz U., Stossel A. and Voss R. (1994), Monte Carlo climate change forecasts with a global coupled ocean atmosphere climate model. *Clim. Dyn., 10*, 1-19.

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

Durham G. B. and Gallant A. R. (2002). Numerical Techniques for Maximum Likelihood Estimation of Continuous-Time Diffusion Processes. *Journal of Business and Economic Statistics, 20* (3), 297-338.

Gill A.E. (1982), *Atmosphere-Ocean Dynamics.* Academic Press.

Greatbatch R.J. and Nadiga B.T. (2000), Four-gyre circulation in a barotropic model with double-gyre wind forcing, *J. Physical Ocean., 30*, 1461-1471.

Hartman P. (1964) *Ordinary Differential Equations.* John Wiley & Sons.

IPCC (Intergovernmental Panel of Climate Change), *2001: Climate Change*, 2001, Cambridge Univ. Press., Cambridge, U.K.

Jiang S., Jin F.F. and Ghil M. (1995). Multiple equilibria, Periodic, and Aperiodic Solutions in a Wind-Driven, Double-Gyre, Shallow-Water Model. *J. Physical Oceanography, 25*, 764-786.

Jones D.A., Poje A.C. and Margolin L.G. (1997). Resolution Effects and Enslaved Finite-Difference Schemes for a Double Gyre, Shallow-Water Model. *Theoret. Comput. Fluid Dynamics, 9* 269-280.

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.

Kennedy M.C. and O'Hagan, A. (2001). Bayesian calibration of computer models, *J. R. Statist. Soc. B, 63*, 425-450.

Kot, M. (2001). *Elements of mathematical ecology.* Cambridge University Press.

Lambert J.D. (1991) *Numerical methods for ordinary differential systems, the initial value problem.* John Wiley & Sons.

MATLAB, Version 5.3.1.29215a (R11.1), 1999, The MathWorks, Inc.

McCalpin J.D. (1995). The Statistics and Sensitivity of a Double Gyre Model: The Reduced-Gravity, Quasigeostrophic Case. *J. Physical Oceanography, 25*, 806-824.

McMillan, N.J., Sacks, J., Welch, W.J. and Gao, F. (1999), Analysis of protein activity data by Gaussian stochastic process models. *Journal of Biopharmaceutical Statistics, 9*, 145-160.

Mitchell J.F.B., Johns T.C., Eagles M., Ingram W.J. and Davis R.A. (1999), Towards the construction of climate change scenarios. *Climatic Change, 41*, 547-581.

Mitchell, T.J. and Morris, M.D. (1992), Bayesian Design and Analysis of Computer Experiments: Two Examples. *Statistica Sinica, 2*, 359-379.

Morris, M. D. (1989). Discussion of "Design and analysis of computer experiments" by Sacks et al. *Statistical Science, 4*, 423-425.

Morris, M. D. and Solomon, A.D. (1995). Design and Analysis of an Inverse Problem Arising from an Advection-Dispersion Process. *Technometrics, 37*, 293-302.

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.

Nychka D., Wikle C. and Royle J.A. (2002), Multiresolution models for nonstationary spatial covariance functions, *Statistical Modelling, 2*, 315-332.

O'Hagan, A. (1978). Curve fitting and optimal design for prediction. *J. R. Statist. Soc. B, 40*, 1-42.

Poje A.C., Jones D.A. and Margolin L.G. (1996). Enslaved Finite Difference Approximations for Quasigeostrophic Shallow Flows. *Physica D, 98*, 559-573.

Sacks J., Welch W.J., Mitchell T.J. and Wynn H.P. (1989). Design and analysis of computer experiments. *Statistical Science, 4*, 409-423.

Welch, W.J., Buck, R.J., Sacks, J., Wynn, P.W., Mitchell, T.J. and Morris, M.D. (1992), Screening, Predicting and Computer Experiments. *Technometrics, 34*, 15-25.

Wynn H. P. (2001). Discussion of "Bayesian calibration of computer models" by Kennedy M.C. and O'Hagan, A., *J. R. Statist. Soc. B, 63*, 450-451.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Max D. Morris for his guidance, patience and support throughout this research and the writing of this thesis. His encouragement has often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my committee members: Dr. William Q. Meeker Jr., Dr. Michael W. Smiley, Dr. Stephen B. Vardeman and Dr. Huaiqing Wu. The ideas they have communicated during formal classes and informal discussions have had a great impact on my doctoral research.

I would additionally like to thank Dr. Doug Nychka and the National Center for the Atmospheric Research, Dr. Dave Higdon and the Los Alamos National Laboratory for the support that I received while I visited these two institutions as a graduate student. Chapter 4 of this dissertation would have not been possible without this support.

Dr. Don Jones of Arizona State University generously provided the ocean PDE model Fortran code and reviewed a section of the third chapter. His help is gratefully acknowledged.