Physical Experimental Design in Support of Computer Model Development

Max D. Morris
Deptartment of Statistics, and
Deptartment of Industrial & Manufacturing Systems Engineering
Iowa State University
Ames, IA 50011-1210

Working Paper: October, 2012
Do not cite or distribute.

Abstract

Computer models of physical systems are often written based on known theory or "first principles" of a system, reflecting substantial knowledge of each component or subsystem, but also the need to use a numerical approach to mimic the more complex behavior of the entire system of interest. However, in some cases, there is insufficient known theory to encode all necessary aspects of the system, and empirical studies are required to generate approximate functional forms. We consider the question of how a physical experiment might be designed to approximate one module or subroutine of a computer model that can otherwise be written from first principles. The concept of preposterior analysis is used to suggest an approch to generating a kind of I-optimal design for this purpose when the computer model is a composition of nonlinear functions that can be directly evaluated as part of the design process. Extensions are then described for situations in which one or more known components must themselves be approximated by metamodels due to the large number of evaluations needed, and for computer models that have iterative structure. A simple "toy" model is used to demonstrate the ideas.

Key words and phrases: Computer experiment, I-optimality, Metamodel, Pre-posterior analysis.

1. Introduction

Consider the problem of writing a deterministic computer model of a complex physical process or system. In most cases, the model is of modular form, with the various components of the process represented by different modules or subroutines. Specifically, consider the case in which most of these modules can be written based on known theory or extensive experience with the process, but for which one module is not known and is to be estimated via physical experimentation.

This situation is represented schematically in Figure 1. Our notation is that the model output, z, is expressed as a function of inputs $x = (x_1, x_2, x_3)$, $z = f(x_1, x_2, x_3)$, within an appropriate domain Δ_x. The full functional form of f is not known. On the other hand, the function

$$z = z(x_1, x_2, y)$$

is fully known (and can be coded), where y is a quantity that can, in principle, be determined by model inputs x_2, x_3. The quantity y is most naturally expressed as an unknown function:

$$y = y(t)$$

where t is a set of "intermediate" variables that are known functions of model inputs:

$$t = t(x_2, x_3).$$

Hence, based on knowledge of the system, t can be computed from x_2 and x_3, and z can be computed from x_1, x_2 and y, but the function relating t to y is not known and must be determined from physical experimentation. x_1 represents those inputs that can only influence z through known functional forms, x_3 represents those inputs that are only needed for determining t, and x_2 represents those inputs that influence z both through their influence on t and through other components of the model. While there are models fitting this description in which any of the described variables are nominal-, integer-, or real-valued, we focus on the case in which all functions and arguments are real-valued and continuous.

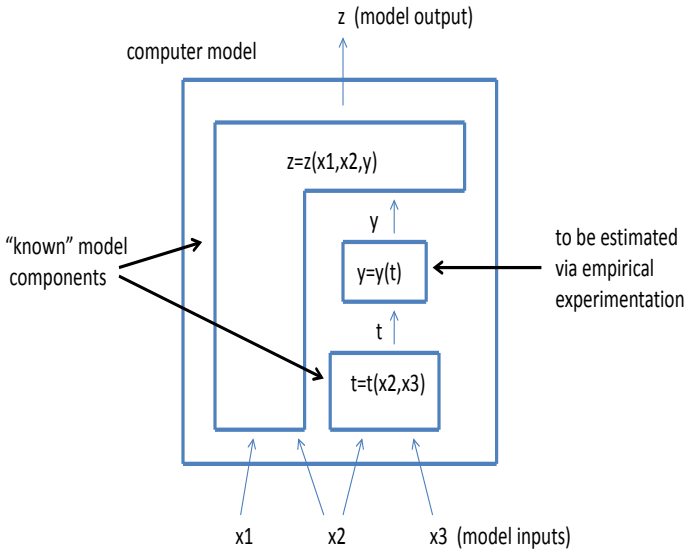

Figure 1: Schematic of sections of the code that are known and that must be determined empirically.

One application area in which this problem is found is in the modeling of biofuel production. Most of the components of this process are well-understood and can be modeled based on what is known. However, different types of biomass (e.g. switchgrass, corn stover, wood chips) react differently to the initial reduction steps. Physical experimentation is required to understand, for example, how much bio-char and bio-oil in each fraction can be expected from a particular type of stock, processed at a given water-to-biomass ratio, biomass particle size, for a given reactor residence time, et cetera. A response surface experiment, often using a quadratic polynomial model, is performed so that this relationship can be estimated and inserted into the overall model.

Since the function $y(t)$ must be estimated by physical experimentation, that estimate will be subject to some imprecision or uncertainty. Response surface methodology (RSM) is a well-developed body of statistical modeling techniques for fitting approximating functions, usually low-order polynomials, to experimental data. Along with this, there is a large body of statistical work on how experiments can be designed so that the estimate of the function y is of high quality (e.g. Myers, Montgomery, and Anderson-Cook). In particular,

the standard experimental design problem is to select a set of n values of t, each used as the controlled condition for obtaining a corresponding "noisy" value of y, so that the estimate of the function $y(t)$, usually constructed via least-squares, is precise and/or accurate. Here we consider a different, but related, design problem: Given that the purpose of the experiment is to generate an estimate of $y(t)$ that will be inserted into an overall model that is otherwise determined, how should the experiment be designed so as to yield an overall model (approximation to f) of high quality?

2. Design Criterion

The usual RSM development focuses on the quality of the estimate of the function $y(t)$. Briefly, an experimental design $D = \{t_1, t_2, t_3, ..., t_n\}$ is specified, and an experiment of n *runs* or *trials* is carried out, each yielding an observed response value; denote the response value associated with the conditions t_i by y_i^*, $i = 1, 2, 3, ..., n$. The usual statistical framework interprets y_i^* as a "noisy" version of $y(t_i)$:

$$y_i^* = y(t_i) + \epsilon_i,\ i = 1, 2, 3, ..., n$$

where ϵ is the random variable representing "noise", with mean zero. We will also make the common assumptions that the variance of each ϵ_i is a constant-but-unknown quanitity, σ^2, and that every pair (ϵ_i,ϵ_j), $i \neq j$, are statistically independent. If an adequate parameterized functional form for $y(t)$ can be specified, the data can be used to fit a "response surface" $\hat{y}(t)$, a statistical estimate of the function of interest over a specified region Δ_t. The goal of experimental design in this context is to select D so that the resulting estimate will be of high quality. For settings in which the form of the fitted model is deemed to be correct, figures of merit for potential designs are often functions of the variance of the fitted function, $Var(\hat{y}(t))$. For example, any design for which:

$$max_{t \in \Delta_t} Var(\hat{y}(t))$$

is minimized is called *G-optimal* (for "global"), while any design for which:

$$\int_{t \in \Delta_t} Var(\hat{y}(t)) dt$$

is minimized is called *I-optimal* (for "integrated").

In this paper, we pursue a common strategy of using a linear regression model for inferences about $y(t)$. In this set-up, we require:

$$y_i = \mathbf{t}_i'\boldsymbol{\beta}, \quad y_i^* = \mathbf{t}_i'\boldsymbol{\beta} + \epsilon_i$$

where $\mathbf{t}_i$ is an appropriately developed vector of functions of the elements of t_i and $\boldsymbol{\beta}$ is a vector of unknown parameters. Taken together, the model for all n observations can be written as:

$$\mathbf{y}^* = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{y}^*$ is an n-vector, and the n-row matrix $\mathbf{X}$ has $\mathbf{t}_i'$ as its ith row. The least-squares estimate of $\boldsymbol{\beta}$ is $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}^*$, for which the distribution is at least approximately $N(\boldsymbol{\beta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$; we refer to this distribution in the following as $BH(\boldsymbol{\beta})$. For any specified value of t, the least-squares estimate of $y(t)$ is $\hat{y}(t) = \mathbf{t}'\hat{\boldsymbol{\beta}}$ which has an approximate or exact normal distribution with mean $y(t)$ and variance $v(t) = \sigma^2\mathbf{t}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{t}$.

Now suppose that the reason for inference about $y(t)$ is the completion of the overall computer model. The model that will actually be assembled will be:

$$\hat{z} = z(x_1, x_2, \hat{y}(t(x_2, x_3))).$$

The notation $\hat{z}$ is to emphasize that, since part of the function has been estimated, it is subject to some uncertainty that propagates from the use of $\hat{y}$ in place of y. We focus on uncertainty expressed as variance, and consider the variance of $\hat{z}$, integrated over Δ_x, as a figure of merit for experimental design. We generalize the integration over Δ_x with a probability distribution Ω and associated density function ω to allow different weighting across the design region:

$$\phi = \int_{\Delta_x} Var(z(x_1, x_2, \hat{y}(x_2, x_3)))\omega(x)dx.$$

With respect to a random variable x with distribution Ω,

$$\phi = E_x(Var(z(x_1, x_2, \hat{y}(x_2, x_3)))).$$

ϕ cannot be determined because it is a function of the unknown vector $\boldsymbol{\beta}$. We address this via *preposterior analysis* (e.g. Bayarri and Berger, 2004) by placing a prior distribution (for purposes of design) on $\boldsymbol{\beta}$, denoted as B, and taking the expectation of ϕ with respect to this distribution:

$$\phi_1 = E_{\boldsymbol{\beta},x}Var(z(x_1, x_2, \hat{y}(x_2, x_3))).$$

3. Basic Implementation

Case 1: z and t are simple functions

Focus first on the case in which $t(x_2, x_3)$ and $z(x_1, x_2, y)$ are simple/fast, and can be evaluated as often as desired in a Monte Carlo integration. Then for a specified design D, the criterion ϕ_1 can be evaluated numerically via the following algorithm:

1. Begin E-loop (expectation)
   (a) DRAW $\boldsymbol{\beta}$ from B.
   (b) DRAW (x_1, x_2, x_3) from Ω.
   (c) COMPUTE $t(x_2, x_3)$.
   (d) Begin V-loop (variance)
      i. DRAW $\hat{\boldsymbol{\beta}}$ from $BH(\boldsymbol{\beta})$.
      ii. COMPUTE $\hat{y}(t) = \mathbf{t}'\hat{\boldsymbol{\beta}}$.
      iii. COMPUTE $z = z(x_1, x_2, \hat{y})$
   (e) COMPUTE $Var_{\hat{y}}(z|\boldsymbol{\beta}, x)$.
2. COMPUTE $E_{\boldsymbol{\beta},x} Var_{\hat{y}}(z|\boldsymbol{\beta}, x)$.

Case 2: z and t are complicated functions

If f is a computationally difficult function, at least one of z or t must be also, and the large numbers of evaluations of these functions required by the previous algorithm may not be practical. In this case, it is more sensible to substitute surrogates or metamodels for these functions. We suppose that one or more separate computer experiments have been performed to construct metamodels or surrogates for each of $z(x_1, x_2, y)$ and $t(x_2, x_3)$. We assume a Bayesian or empirical Bayesian approach is taken, and that the metamodels take the form of predictive distributions $T(x_2, x_3)$ and $Z(x_1, x_2, y)$, respectively. As with $\boldsymbol{\beta}$, the uncertainty in t and z is managed by expectation over the distributions that represent these quantities:

$$\phi_2 = E_{\boldsymbol{\beta},x,t,z} Var(z(x_1, x_2, \hat{y}(t(x_2, x_3)))).$$

Then, a modified algorithm for experimental design is:

1. Begin E-loop
   (a) DRAW $\boldsymbol{\beta}$ from B.
   (b) DRAW (x_1, x_2, x_3) from Ω.
   (c) DRAW t from $T(x_2, x_3)$.
   (d) DRAW $\mathcal{Z}(y)$ from $Z(x_1, x_2, y)$, jointly for all values of y.
   (e) Begin V-loop
      i. DRAW $\hat{\boldsymbol{\beta}}$ from $BH(\boldsymbol{\beta})$.
      ii. COMPUTE $\hat{y}(t) = \mathbf{t}'\hat{\boldsymbol{\beta}}$.
      iii. COMPUTE $z = \mathcal{Z}(\hat{y})$.
   (f) COMPUTE $Var_{\hat{y}}(z|\boldsymbol{\beta}, x, t)$.
2. COMPUTE $E_{\boldsymbol{\beta},x,t,z}(Var_{\hat{y}}(z|\boldsymbol{\beta}, x, t))$.

Note that in this case, a random draw from $\mathcal{Z}$, taken in the outer expectation loop, must be a functional draw because it is used repeatedly for multiple draws of $\hat{y}$ within the inner variance loop.

Example

To demonstrate the design criteria outlined above, consider a simple test case in which inputs $(x_1, x_2, x_3) \in [0, 1]^3$, with Ω taken as the continuous uniform distribution over this space. The transformed inputs to be used in empirical modeling are taken to be:

$$t_1 = log_2(1 + \tfrac{x_2+x_3}{2}), \qquad t_2 = log_2(1 + x_2), \qquad z = x_2 e^{-(x_1+y^2)}$$

and $y(t_1, t_2)$ will be estimated based on data collected in a physical experiment, with quadratic polynomial regression, i.e.

$$y(t_1, t_2) = \beta_0 + \beta_1 t_1 + \beta_2 t_2 + \beta_{11} t_1^2 + \beta_{22} t_2^2 + \beta_{12} t_1 t_2.$$

An experimental run at a given value of (t_1, t_2) will result in $y^*(t_1, t_2) + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$. We express our initial uncertainty about the regression coefficients using a multivariate normal distribution:

$$B = N(\mu_{\boldsymbol{\beta}}, \mathbf{V}_{\boldsymbol{\beta}}).$$

In the calculations to follow, we use:

$$\sigma^2 = \tfrac{1}{100}, \qquad \mu'_{\boldsymbol{\beta}} = (\tfrac{1}{2}, 0, 0, 0, 0, 0, 0), \qquad \mathbf{V}_{\boldsymbol{\beta}} = \tfrac{1}{100}\mathbf{I}$$

For reasons mostly of simplicity, in this example, σ^2 is regarded as known, as might be reasonably assumed when ϵ represents primarily measurement error generated from a well-documented measurement system. Alternatively, a distribution could also be proposed for this parameter, and values of it drawn in the expectation loop along with those of x and $\boldsymbol{\beta}$.

Note that for the functions we've selected, the domain of (t_1, t_2) is $[0, 1]^2$ and so we consider experimental design in this region. Specifically, we consider the class of complete 3^2 factorial design for which 0 and 1 are the low and high levels for each of t_1 and t_2, the intermediate level of each is varied independently between 0.2 and 0.8, and 4 experimental runs are included at the run defined by the intermediate levels of t_1 and t_2 for replication. Hence all designs considered contain 12 runs counting replicates; an example is depicted in Figure 2. (Richer classes of designs could certainly be considered; our intent here is only to demonstrate how the approach can be used to select from among a reasonable collection of designs.)

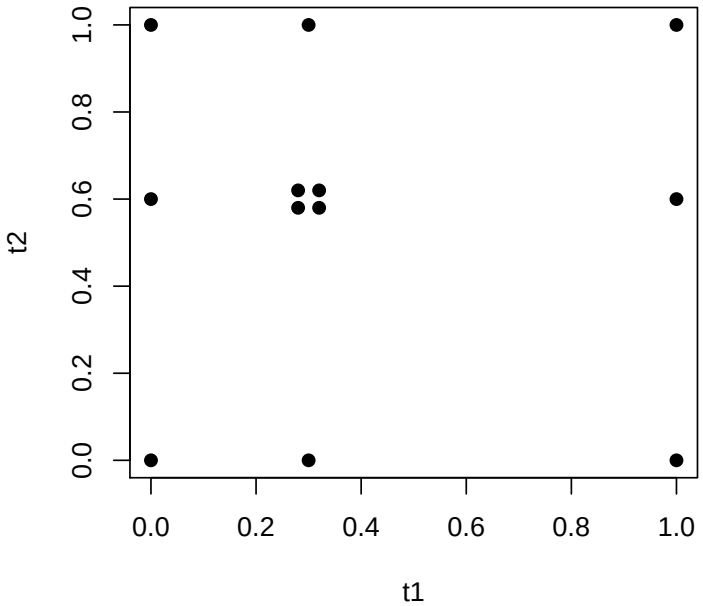

Figure 2: Example from the class of experimental design considered.

Beginning with case 1 for which $t_1(x_2, x_3)$, $t_2(x_2, x_3)$, and $z(x_1, x_2, y)$ can each be evaluated easily, ϕ_1 was computed via stochastic simulation for each candidate design. The outer expectation loop requires drawing $\boldsymbol{\beta}$ from B and (x_1, x_2, x_3) from Ω, and direct calculation of (t_1, t_2) from the known formulae. The inner variance loop calls for repeated draws of $\hat{\boldsymbol{\beta}}$, and from this, $\hat{y}$ that would be estimated from the planned regression experiment at the selected value of (t_1, t_2), assuming the drawn value of $\boldsymbol{\beta}$ and experimental noise with variance σ^2. More specifically, letting $\mathbf{t}$ represent the 6-element vector of monomials of (t_1, t_2) required by the model,

$$\hat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1}), \qquad \hat{y}(t) = \mathbf{t}'\hat{\boldsymbol{\beta}}$$

where $\mathbf{X}$ is the 12×6 model matrix determined by the design being evaluated. For each drawn $\hat{y}$, $z(x_1, x_2, \hat{y})$ is evaluated, and the variance of the z's generated within the inner loop is calculated. The algorithm output, for each experimental design under consideration, is the average of this variance over the quantities drawn in the outer loop. In the calculation presented here, the expectation loop was executed 500 times, and the variance loop was also executed 500 times (within each execution of the expectation loop). The resulting calculations of $\sqrt{\phi_1}$ are displayed in Figure 3 for the experimental designs considered; the two axes represent the mid-level values assigned to each of t_1 and t_2 for each design.

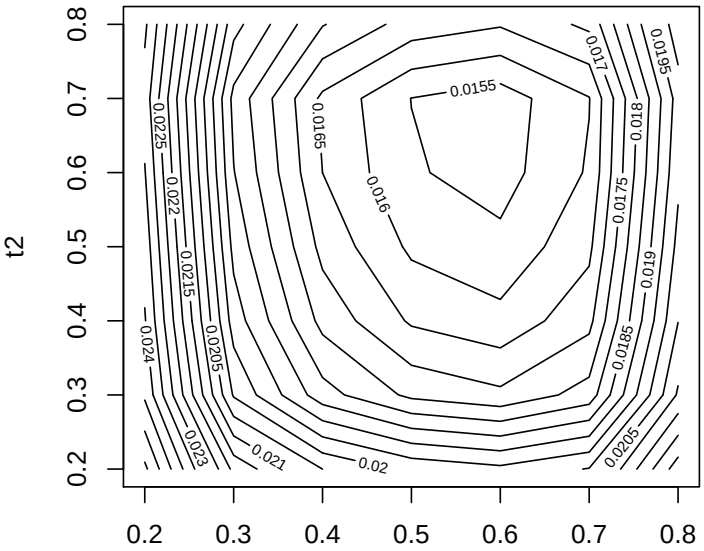

Figure 3: $\sqrt{\phi_1}$ for the experimental designs considered.

Case 2 anticipates that $t_1(x_2, x_3)$, $t_2(x_2, x_3)$, and $z(x_1, x_2, y)$ cannot be easily computed

and must themselves be represented by metamodels constructed from relatively small computer experiments. Here, we do this through parametric kriging as described in Currin et al. (1991) and elsewhere. A Latin hypercube design in 6 runs was constructed for the two variables (x_2, x_3), the corresponding values of t_1 and t_2 were calculated, and separate kriging predictors were constructed independently for each. A Gaussian covariance function was used, the process parameters (mean, variance, and correlation parameter) were estimated via approximate maximum likelihood, and those estimates were used for prediction as if they were known parameter values (i.e. the "empirical Bayes" formulation outlined in Currin et al.). Where estimates of t_1 and t_2 are needed, they are drawn (in this case independently) from their respective conditional (on the 6 observed function values) normal distributions, denoted T. Figure 4 displays the functions t_1 and t_2, and the means and standard deviations of their respective predictive distributions, each plotted as functions of x_2 and x_3. Maximum and average (over x_2 and x_3) standard deviations for t_1 were 0.00194 and 0.00771 respectively, and for t_2 were 0.00493 and 0.01935 respectively. Similarly, a 21-run Latin hypercube design was constructed for the three variables (x_1, x_2, y), corresponding values of z were calculated, and a kriging predictor constructed. Figure 5 displays the functions $z(x_1, x_2, y = 0.25)$ and $z(x_1, x_2, y = 0.75)$, and the means and standard devciations of their respective predictive distributions, each plotted as functions of x_1 and x_2. Conditioned on each of these values of y, the average and maximum standard errors of z were approximately 0.020 and 0.025 respectively. Where values of z are needed, they are drawn from this distribution, denoted by Z.

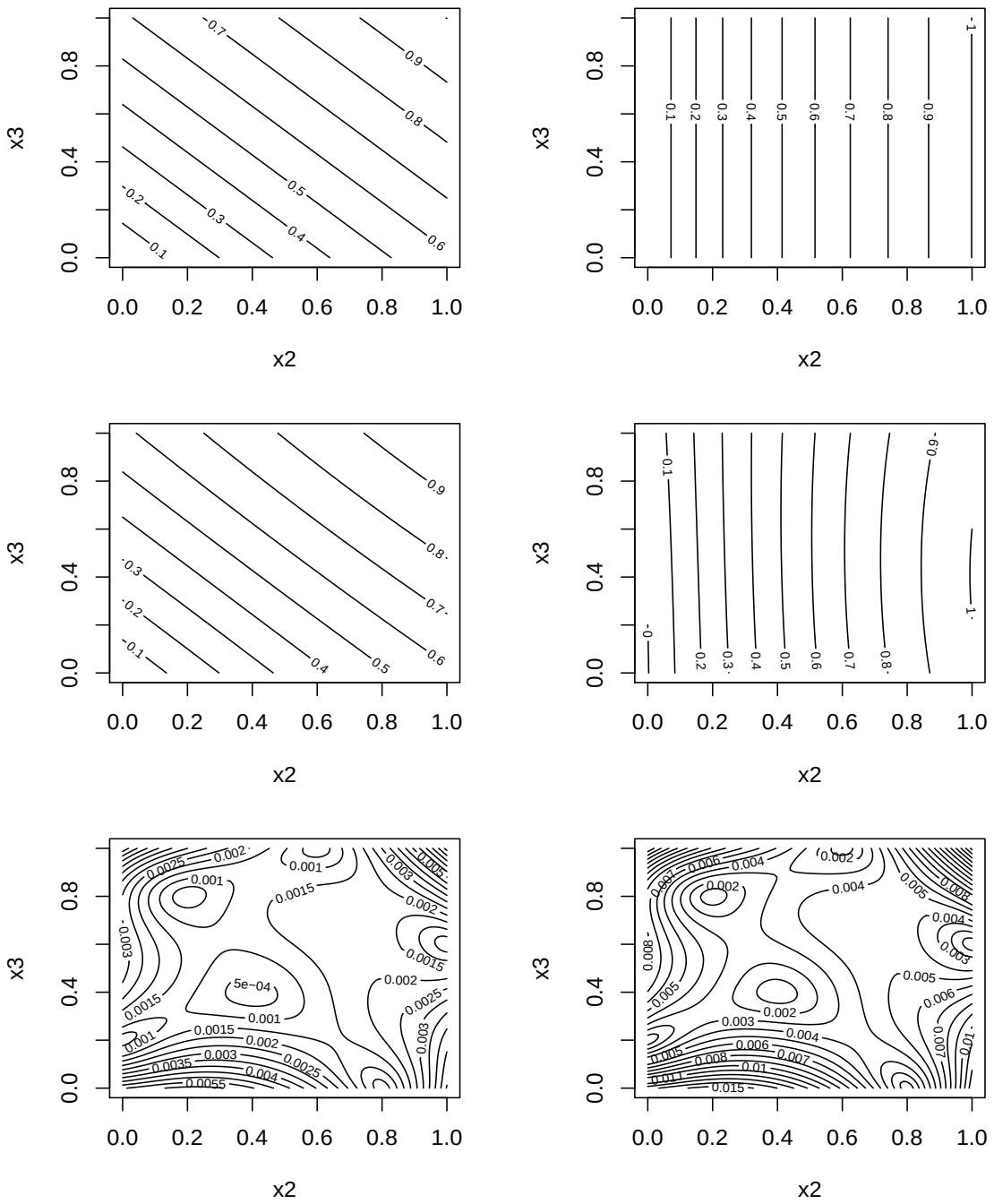

Figure 4: Functions t_1 and t_2 (top row), predictive expectations of t_1 and t_2 (middle row), predictive standard deviations of t_1 and t_2 (bottom row).

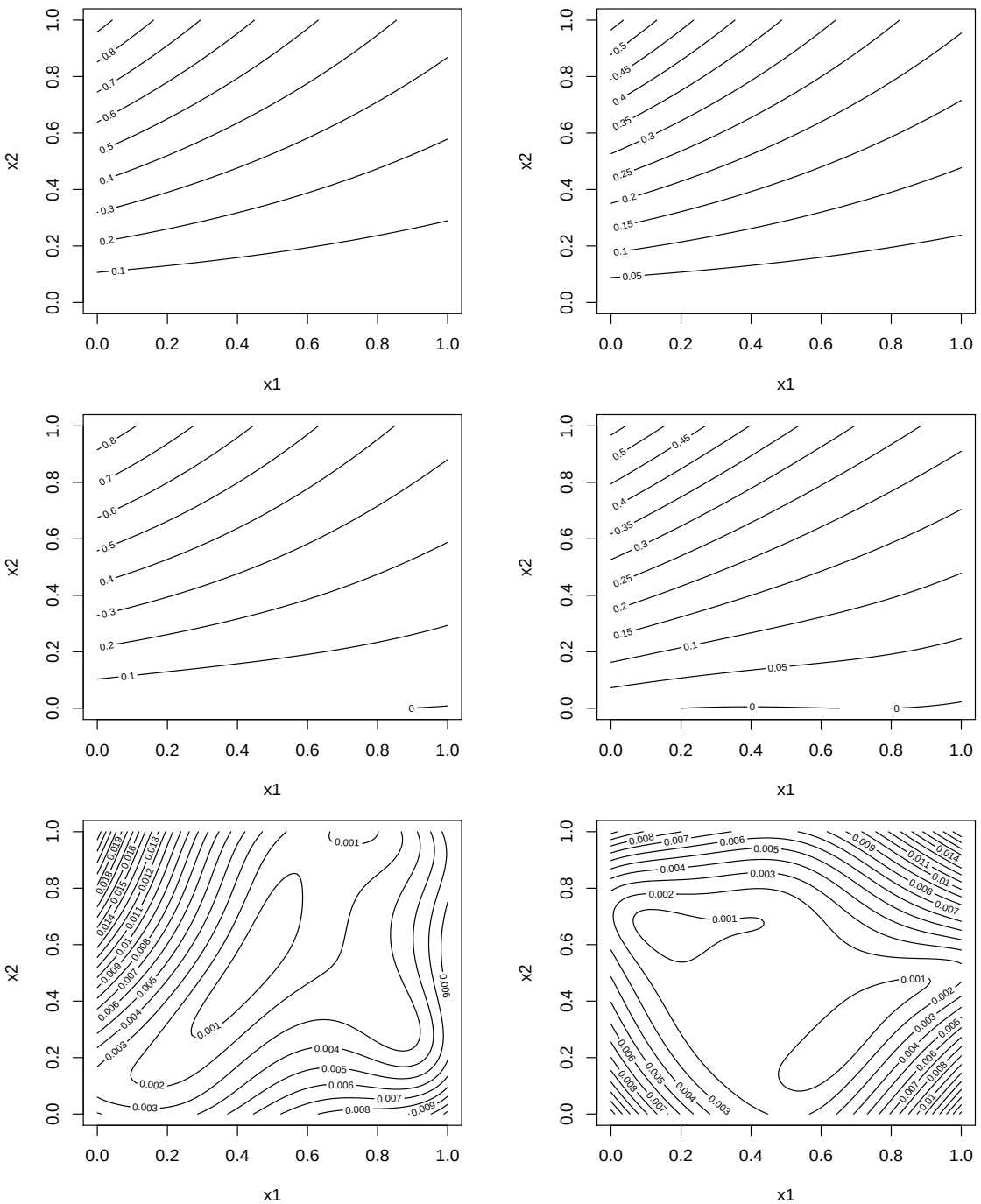

Figure 5: Functions $z(x1, x2, y = 0.25)$ and $z(x1, x2, y = 0.75)$ (top row), corresponding predictive expectations (middle row), and corresponding predictive standard deviations (bottom row).

For case 2, the expectation loop calls for draws of $\boldsymbol{\beta}$ from B and (x_1, x_2, x_3) from Ω as in case 1, and for the selected (x_2, x_3), a draw of (t_1, t_2) from T. The the function $\mathcal{Z}(y)$ is drawn from Z for the specified values of x_2 and x_3, jointly for all possible values of y (at least conceptually). To accomplish this in the simulation, we jointly draw $\mathcal{Z}$ over a grid of values of y, and interpolate this set of values where draws are needed in the variance loop. Within this inner loop, draws are made of $\hat{\boldsymbol{\beta}}$ from $BH(\boldsymbol{\beta})$, and used to calculate predictions $\hat{y}(t) = \mathbf{t}'\hat{\boldsymbol{\beta}}$, the resulting values of $\mathcal{Z}(\hat{y})$ are computed, and the variance of these values calculated; the expectation of this variance is again estimated as the average over expectation loop iterations. Again, both outer and inner loops were executed 500 times, and the resulting calculations of $\sqrt{\phi_2}$ are displayed in Figure 6.

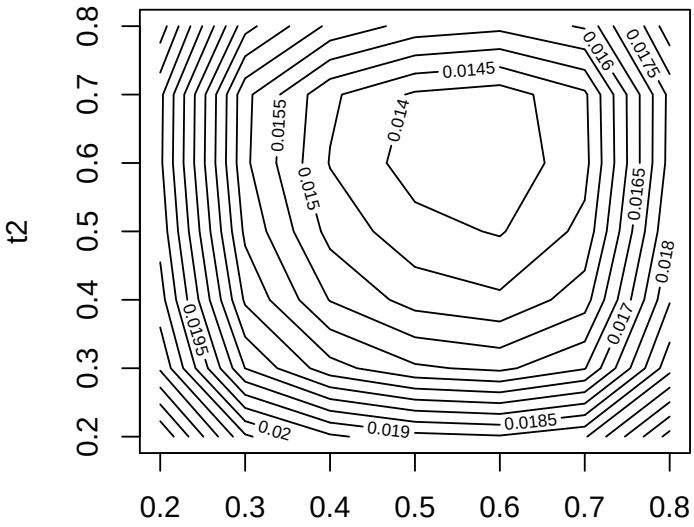

Figure 6: $\sqrt{\phi_2}$ for the experimental designs considered.

With respect to either ϕ_1 or ϕ_2 (as defined using the distributions employed here), the optimal design from the class examined is based on mid-level values of 0.6 for t_1 and 0.7 for t_2. Some intuition for this comes from the functional forms of t_1 and t_2; the joint distribution induced for these variables from uniform Ω is concentrated at relatively large values for each.

4. Iterative Models

The structure of computer models considered to this point is that of a direct composition of functions $t(x_2, x_3)$, $z(x_1, x_2, y)$, and $y(t_1, t_2)$, the first two of which may be nonlinear. In many applications, computer models have an iterative form in which some or all components are evaluated several times in the course of one model run. A broad class of models that fit this description are those that solve partial differential equations or difference equations, for which each component models the change in relevant variables over time. In these applications, some or all of the outputs may be "evolved" versions of inputs, that define *initial conditionis* for the system.

Figure 7 depicts one iterative generalization of the simple model used in the examples above. Here x_2 is an initial value of the output of interest, the entire model as previously described is iterated with the value of z computed at each iteration used as the value of x_2 in the next. For any specified input vector, the uncertainty in model output derives from multiple evaluations of $\hat{y}$, and the vectors t that specify these evaluations, determined jointly by all three of $t(x_2, x_3)$, $z(x_1, x_2, y)$, and $y(t_1, t_2)$. While this complicates design evaluation, the preposterior analysis approach described above can also be used here, by including the general structure of iteration required by the model inside the variance loop, to be completed by multiple evaluations of $\hat{y}$ based on the same (drawn) value of $\hat{\boldsymbol{\beta}}$.

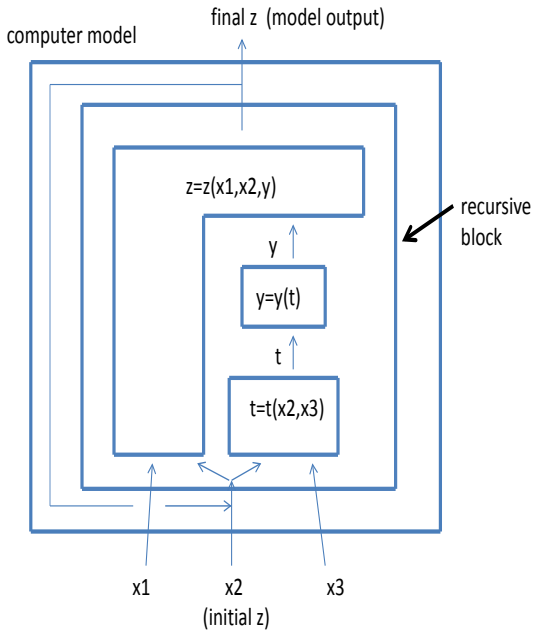

Figure 7: Schematic of sections of the code that are known and that must be determined empirically.

1. Begin E-loop
   (a) DRAW $\boldsymbol{\beta}$ from B.
   (b) DRAW (x_1, x_2, x_3) from Ω.
   (c) Begin V-loop
      i. DRAW $\hat{\boldsymbol{\beta}}$ from $BH(\boldsymbol{\beta})$.
      ii. Begin model iteration
         A. COMPUTE $t(x_2, x_3)$.
         B. COMPUTE $\hat{y}(t) = \mathbf{t}'\hat{\boldsymbol{\beta}}$.
         C. COMPUTE $z = z(x_1, x_2, \hat{y})$
         D. $x_2 \leftarrow z$
   (d) COMPUTE $Var_{\hat{y}}(z|\boldsymbol{\beta}, x)$.
2. COMPUTE $E_{\boldsymbol{\beta},x} Var_{\hat{y}}(z|\boldsymbol{\beta}, x)$.

Figure 8 displays $\sqrt{\phi_1}$ for the iterative model shown in Figure 7, for 1 (repeated from Figure 3 for convenience), 2, 5, and 10 model iterations. As the number of iterations increases, the optimal center values for each of t_1 and t_2 both move closer to 0.5, indicating that a more spatially balanced design is preferred for repeated use of the estimated function in this context.

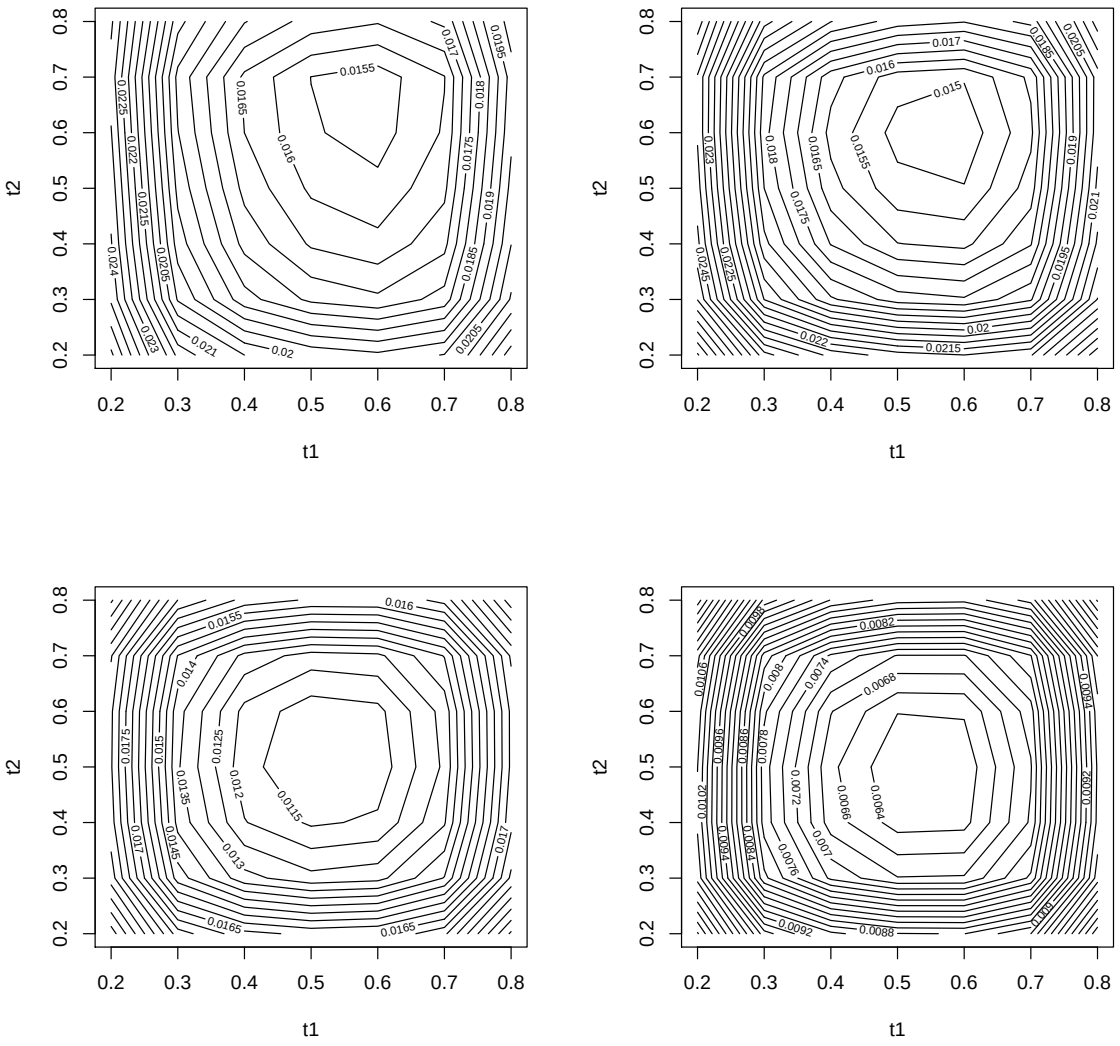

Figure 8: $\sqrt{\phi_1}$ for the experimental designs considered, for 1-, 2-, 5-, and 10-iteration models.

Figure 9 shows the results of the analogous "Case 2" calculation for this iterative model example, in which the functions z and t are represented by Gaussian stochastic processes

coniditioned on 6 and 21 function evaluations, respectively, as described in Section 3. The pattern of "shrinkage" of the experimental mid-values of t_1 and t_2 toward 0.5, as the number of iterations increases, is qualitatively similar here to the previous calculations in which metamodels were not used.

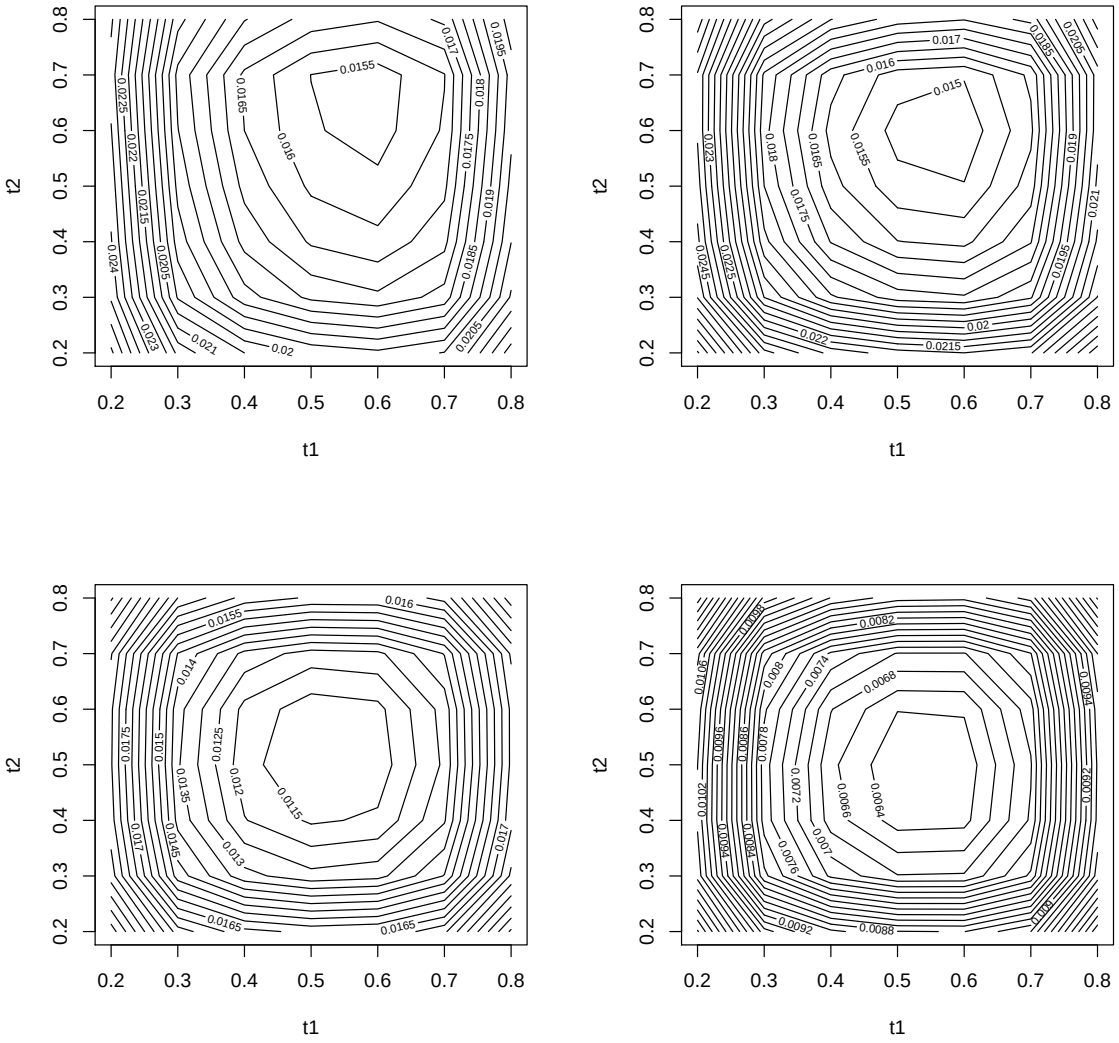

Figure 9: $\sqrt{\phi_2}$ for the experimental designs considered, for 1-, 2-, 5-, and 10-iteration models.

A more general depiction of design evaluation is symboized in Figure 10, for iterative models and where some of the known components are represented by metamodels (as in Case II), and outlined below. In general, draws from Ω, B, and from all metamodels are made

once in each execution of the expectation loop. Conditioned on these values, realizations of $\hat{\boldsymbol{\beta}}$ are drawn within the variance loop, and the iterative calculation of the code completed using all draws. Note that this means $\hat{y}(t)$ (as well as the metamodels) is evaluated at a potentially different argument for each iteration, but that all calculations leading to a single value of z must be based on a single realization.

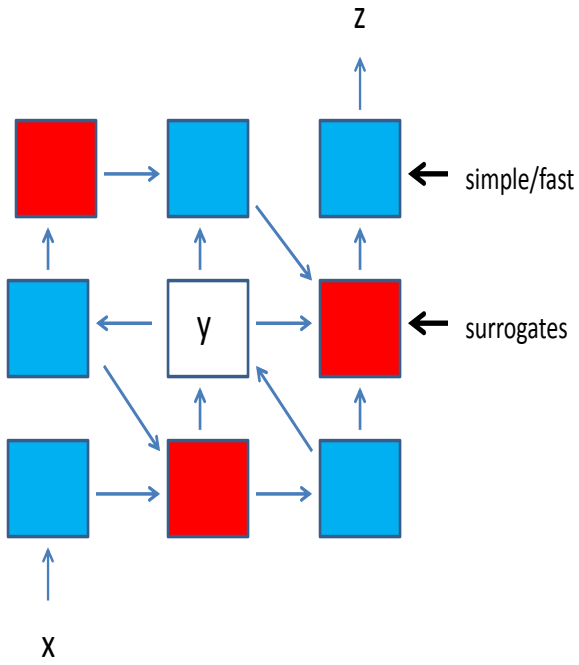

Figure 10: General conceptual schematic of a model with one unknown component for which an experiment is to be design, known components that can be evaluated explicitly, and known components that must be represented by metamodels.

An abreviated sketch of the algorithm corresponding corresponding to this case is:

1. Begin E-loop
    (a) DRAW $\boldsymbol{\beta}$ from B.
    (b) DRAW (x_1, x_2, x_3) from Ω.
    (c) DRAW* surrogates, including t
    (d) Begin V-loop
        i. DRAW $\hat{\boldsymbol{\beta}}$ from $BH(\boldsymbol{\beta})$.
        ii. EXECUTE $\rightarrow z$

(e) COMPUTE $Var_{\hat{y}}(z|...)$

2. COMPUTE $E_{...}Var_{\hat{y}}(z|...)$

* Model structure determines the extent to which draws need to be joint (which arguments are constant and which are not).

5. Discussion

In application, scientific modelers often use a combination of informal uncertainty quantification about $y(t)$ and numerical sensitivity analysis for other components of the model, to guide the selection of experimental effort. This approach is reasonable, and captures at least some of the structure of what is being suggested more formally in this paper. For more complex models, especially with iterative structure, the additional power of a comprehensive statistical approach may be helpful.

It should be noted that in some of the applications motivating this work, the uncertainty about $y(t)$ is far from uniform over Δ_t. In fact, this function may be quite well-known over most of its domain, with substantial uncertainty limited to edges or corners of the region representing, for example, extreme physical conditions, or the extent of material fatigue after long periods of time. Therefore, in realistic applications, substantial effort should be put into characterizing B so that it actually reflects the uncertainty of concern.

Finally, the discussion here is given at a very general level, and as with most applications of methodology, the real work is in developing the details needed in particular cases. Here are some brief descriptions of (sometimes substantial) questions that would arise.

- What is the best way to organize the design of computer experiments to create metamodels of known-but-complicated modules? Where it is more practical to run the complete model, rather than individual components, might data for all metamodels be collected together, and if so, how should values of y be generated in place of the absent module?

- For iterative problems, how should thel range of t be determined for the physical experiment? This issue was avoided in the artificial example presented here, by using

only functions for which values are within $[0, 1]$, but this will not be character of many real problems.

- For iterative models, what is the most efficient and/or practical representation of functional realizations drawn in the expectation loop? For applications of higher dimension, these draws may require substantial space allocations.

- How can experimental designs for this purpose be constructed iteratively? In the exercises presented here, only a small number of designs were considered and their criterion values were simply computed in parallel evaluation. But this will not be practical for larger problems where the conceptual class of candidate designs is much larger, and strategies more like the "point exchange algorithms" used for linear models may be more appropriate.

References

Bayarri, M.J. and J.O. Berger (2004). "The Interplay of Bayesian and Frequentist Analysis," *Statistical Science* **19**, 58-80.

Currin, C., T.J. Mitchell, M.D. Morris, D. Ylvisaker (1991). "Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiments," *Journal of the American Statistical Association* **86**, 953-963.

Myers, R.H., D.C. Montgomery, C.M. Anderson-Cook (2009). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments, 2nd Edition*, ISBN 978-0-470-17446-3, John Wiley & Sons, Hoboken, NJ.