Statistica Sinica 2(1992), 359-379

BAYESIAN DESIGN AND ANALYSIS OF COMPUTER EXPERIMENTS: TWO EXAMPLES

Toby J. Mitchell and Max D. Morris

Oak Ridge National Laboratory

Abstract: In a computer experiment, the data are produced by a computer program that models a physical system. The experiment consists of a set of model runs; the design of the experiment specifies the choice of program inputs for each run. This paper demonstrates two applications of a Bayesian method for the design and analysis of computer experiments to predict model output corresponding to input values for which the model has not been run. When the original code is long-running, the fast predictor produced by this method can serve as an efficient, though approximate, substitute. The models used in the two examples are (i) a computer model for the combustion of methane and (ii) a computer model that simulates the compression molding of sheet molding compound in the manufacture of an automobile hood.

Key words and phrases: Bayesian prediction, computer model, combustion, compression molding, interpolation, optimal design, stochastic process.

1. Introduction

The purpose of this paper is to demonstrate the application of existing Bayesian methods for design and analysis of computer experiments (DACE) to two computer models. The first example is concerned with a model for the combustion of methane; its role in the application described here is to calculate ignition delay times as a function of the reaction rates for the key reactions. The second example involves computer simulation of a compression mold-filling process used in the manufacture of automobile hoods. The role of this model, as described here, is to calculate the position of the flow front of the material as a function of time. In both examples, the original models are moderately long-running. On a Cray X-MP, the methane combustion model typically requires 20 seconds per run, while the compression molding model requires 4-5 minutes.

Our primary use of these models was to try out our current versions of Bayesian design and prediction, which are described in Currin, Mitchell, Morris and Ylvisaker (1991). Except for some minor philosophical differences, our underlying approach is essentially the same as that discussed by Sacks, Welch,

Mitchell and Wynn (1989). As noted there, versions of this approach have been used for a long time in various settings, e.g., kriging and Bayesian interpolation.

The objective in each example was to generate prediction formulas that could serve as fast substitutes for the real model in certain well-defined tasks. This done, we did not follow through any further, so the two accounts here are best considered as realistic examples rather than complete case studies or scientific applications. The reader who is already familiar with the general method and with the examples presented by the papers cited above may wish to go directly to the second example, which involves a somewhat more innovative application of the prediction method.

2. Example I: Methane Combustion Model

2.1. Introduction

The model (Frenklach and Rabinowitz (1989)) was developed by Michael Frenklach of Penn State University and Martin Rabinowitz of NASA, and came to us through the courtesy of Jerome Sacks, William Welch, Susannah Schiller, and Robert Buck, who used it in various applications of DACE carried out at the University of Illinois. (See, e.g., Example 2 in Sacks, Schiller and Welch (1989).)

The model calculates, among other things, the ignition delay time in a combustion reaction, given input variables that describe the experimental conditions, the physical properties of the substances involved, and the rates of the reactions. For an given reaction mechanism, it is of interest to find values of the reaction rates for which the computed ignition delays closely approximate the (physically) observed ignition delays in a collection of different experiments. This can be formulated as a nonlinear least squares problem, but a straightforward numerical optimization would be impractical due to the dimensionality of the domain and to the nature of the least squares objective function, which has multiple ridges and valleys (Frenklach and Rabinowitz (1989)). Since the calculation of each ignition delay requires the solution of a large set of differential equations (requiring approximately 20 seconds on a Cray X-MP computer), such a search would, according to Frenklach and Rabinowitz, "place a prohibitively large demand upon computer time". The approach taken by them was to approximate the response function of interest (ignition delay as a function of the key reaction rates) by a polynomial and then use the approximation instead of the model-calculated response values in the least squares objective function. Here we address only the approximation part of this approach, using methods of Bayesian prediction, which we regard as being better suited than polynomial approximation for functions that are observed without random error.

In the computer experiment we conducted using this model, there were seven

design variables, each corresponding to a particular reaction rate of interest. All other inputs were held fixed at nominal values. The response (y) was taken to be the logarithm of the ignition delay. (The designation of the seven rates of interest, and the choice of metric for expression of the response, were made by Frenklach and Rabinowitz.) Altogether, 50 runs of the computer model were made, in stages, where each run corresponded to a particular configuration of the design variables. We used the 50 resulting ignition delay values, $y(t^{(i)})$, $i = 1, 2, \ldots, 50$, where $t^{(i)}$ is the seven-dimensional vector of log reaction rates in the ith run, to compute the prediction function $\hat{y}$ in the manner described by Currin et al. (1991). This is based on a Bayesian approach. At any point t in the region of interest T, $\hat{y}(t)$ is the posterior mean of the random variable that is used to represent uncertainty about $y(t)$. Bayesian posterior probability bounds for $y(t)$ were also computed. Predictions can be made very quickly using $\hat{y}$; the disadvantage, of course, is that $\hat{y}(t)$ is only an approximation to $y(t)$.

Both the motivation (fast, inexpensive prediction) and the results (predictions and uncertainty intervals) have parallels in response surface methodology; see, e.g., Box and Draper (1987). There are some fundamental differences, however. In our setting, there is no random error, i.e., the function y is deterministic, so there is no basis for the "response = signal + noise" model common to response surface methods. Perhaps the most important difference between our approach and response surface methodology is that the latter is based on the use of empirical models (typically polynomials) fit to the data by the method of least squares, whereas in our approach $\hat{y}$ is an interpolator whose functional form is determined indirectly by the choice of correlation function (Section 2.3). If desired, polynomials could be fitted to $\hat{y}$ as a follow-on, but since we are interested only in fast prediction here, we have not done this.

2.2. Design

The central idea (which is not original with us) is to represent uncertainty about the function y on a k-dimensional region of interest T by means of a stochastic process (random field) Y. Although "stochastic process" often connotes random sequences of events or functions of time, our usage is more general: it simply means "random function". This is a direct generalization of the Bayesian use of "random variable" to represent uncertainty about a scalar quantity. For simplicity and convenience, we use stationary Gaussian (normal) processes as priors. These are fully described by a constant $\mu = E[Y(t)]$, a constant $\sigma^2 = V[Y(t)]$, and a correlation function R, where $R(d) = \mathrm{Corr}[Y(t + d), Y(t)]$ and where $t = (t_1, t_2, \ldots, t_k)$ and $t + d = (t_1 + d_1, t_2 + d_2, \ldots, t_k + d_k)$ are any two different "sites" (points in T) separated by a difference vector d.

Under the kriging approach, the unknown function y is assumed to be a

realization of the stochastic process $\{Y(t), t \in T\}$, which is treated as a *model* for y. This model may also be expressed as one would a regression model with correlated errors: $Y(t) = \mu + \varepsilon(t)$ where the "error" ε has mean 0, variance σ^2, and correlation structure given by R. Although the *prior* mean in this model is simply a constant, the correlation structure is very flexible, and this prior can be used even when y is quite complex.)

Let Y_D be the random n-vector that represents uncertainty about y at the n design sites and let $Y_{\bar{D}}$ be the random vector that represents uncertainty about the remaining sites in T. To motivate our design criterion, it will be convenient to let $\bar{D}$ be a finite set of $\bar{n}$ sites (as it, indeed, is on a digital computer). After the experiment, we are concerned only with the uncertainty in $Y_{\bar{D}}$, since y_D is known exactly.

It can be shown, using standard results from multinormal distribution theory, that the posterior covariance matrix for $Y_{\bar{D}}$ given $Y_D = y_D$ is

$$\sigma^2 C^*_{\bar{D}\bar{D}} = \sigma^2 \left[C_{\bar{D}\bar{D}} - C_{\bar{D}D} C_{DD}^{-1} C_{D\bar{D}} \right] \tag{2.1}$$

where $C_{\bar{D}\bar{D}}$ is the $\bar{n} \times \bar{n}$ prior correlation matrix for $Y_{\bar{D}}$, C_{DD} is the $n \times n$ prior correlation matrix for Y_D, $C_{\bar{D}D}$ is the $\bar{n} \times n$ matrix of prior correlations between the elements of $Y_{\bar{D}}$ and Y_D, and $C_{D\bar{D}}$ is the transpose of $C_{\bar{D}D}$. Note that (2.1) does not depend on y_D.

By analogy with the well known criterion of D-optimality in the design of experiments for linear model fitting, we shall adopt as our design construction criterion the minimization of $|C^*_{\bar{D}\bar{D}}|$. We find this criterion appealing, for reasons given by Currin et al. (1991), but other criteria could be used. For example, Sacks, Schiller and Welch (1989) and Sacks, Welch, Mitchell and Wynn (1989) used a criterion which amounts to minimizing the trace of $C^*_{\bar{D}\bar{D}}$, with good results in several examples.

Applying the rule for determinants of partitioned matrices, we know that

$$|C_{TT}| = |C_{DD}| \, |C^*_{\bar{D}\bar{D}}|, \tag{2.2}$$

where C_{TT} is the prior correlation matrix for the combined vector $(Y'_D, Y'_{\bar{D}})'$. Since C_{TT} is fixed by the prior, the D-optimality criterion is equivalent to maximizing $|C_{DD}|$. That is, we want to choose D to maximize the determinant of the $n \times n$ matrix of *prior* correlations among the design sites. This criterion can also be regarded as a special case of the "maximum entropy" criterion. (See Shewry and Wynn (1987) and Currin et al. (1991) for more details.)

Of course, one cannot maximize $|C_{DD}|$ without specifying how C_{DD} depends on D. For our priors, this means specifying the correlation function R. We favor using a weak correlation function, i.e., one for which $R(d)$ decreases rapidly to

zero as d increases. Such a strong conviction of prior ignorance is not useful for analysis, since one would need to observe y at very many sites, located densely in T, in order to yield predictions that are usefully precise. At the design stage, however, we feel that the choice of a weak correlation function is appropriately conservative.

For design purposes then, we use the exponential correlation:

$$R(d) = e^{-\theta \sum_j |d_j|} \tag{2.3}$$

where θ is "large". Asymptotically (as $\theta \to \infty$), it can be shown that the D-optimality criterion, where (2.3) is used to construct C_{DD}, maximizes the minimum intersite distance $\sum |d_j|$ among design points, and favors those designs with the fewest pairs whose intersite distance matches this minimum. This is a special case of a result due to Johnson, Moore and Ylvisaker (1990), who called such designs "maximin distance" designs. In this sense, the designs we construct will attempt to push the design points as far away from each other as possible.

For design construction, we use an algorithm similar to DETMAX (Mitchell (1974)). Starting with a random set of n sites, the algorithm does a series of "excursions" in which candidate sites are added to and removed from the design. When adding a site, the chosen site is intended to be the one at which the posterior variance, based on the current design, is largest. It may not be possible to ensure this if there are many sites to consider; if this is the case, the algorithm does a limited search. When removing a site, the chosen site is the one corresponding to the largest diagonal element in the inverse of the current C_{DD} matrix. See Currin et al. (1991) for further details.

A 50-run, 7-variable design for the methane combustion code was constructed in this way, using the correlation function (2.3) with $\theta = 4.6$ ($e^{-\theta} = 0.01$) to form C_{DD}. (Larger values of θ, which would be more likely to yield a maximin design, were not feasible because of numerical difficulties encountered by the design algorithm.) We shall refer to this design as D50; it consists of all 50 runs given in Table 1. The design was constructed in the unit cube $[0,1]^7$, with the intention of scaling the actual input variables (the seven log reaction rates) in the computer experiment appropriately. Since we restricted the candidates to an evenly spaced 5^7 grid, each design variable can have values only in the set (0, .25, .5, .75, 1). The minimum intersite distance $\sum_{j=1}^{7} |d_j|$ in D50 is 1.5; this occurs for six pairs of sites. Although all of the design points are seen to be on the boundary of T, this does not mean that they are all far from the center of T. The mean distance of the 50 design points from the center of T is 2.38, compared to 1.75 for a randomly chosen site and 3.5 for a corner of T. Of the 78125 sites of the 5^7 grid that served as candidates, the largest distance to the design is 2.25; the average is 1.15.

Table 1. Fifty-run design used in the methane combustion example. Designs D20, D30, D40, and D50 consist of the first 20, 30, 40, and 50 runs, respectively. y is the logarithm of the ignition delay time in μsecs.

Run	t_1	t_2	t_3	t_4	t_5	t_6	t_7	y
1	0.00	0.00	0.00	0.50	1.00	1.00	0.25	7.9315
2	0.25	0.50	0.50	0.75	0.00	1.00	0.00	6.2171
3	0.00	1.00	0.00	0.25	0.00	0.00	1.00	7.8535
4	0.50	0.50	0.75	0.00	1.00	0.25	0.00	7.5708
5	0.00	0.75	0.75	1.00	1.00	1.00	0.50	6.3491
6	1.00	0.00	1.00	0.25	0.00	1.00	0.00	5.3045
7	0.00	1.00	0.00	0.75	1.00	0.50	1.00	8.5372
8	0.75	0.25	0.00	1.00	1.00	0.00	1.00	7.8710
9	0.50	0.75	0.25	0.00	0.25	0.50	0.50	7.8725
10	0.25	1.00	0.75	0.75	0.50	0.00	0.25	6.5930
11	0.50	0.00	1.00	0.25	1.00	0.75	1.00	6.2131
12	1.00	0.00	0.00	0.50	0.50	0.50	1.00	7.6311
13	1.00	0.50	1.00	0.75	0.00	0.25	0.50	5.1090
14	0.00	1.00	0.25	0.25	0.75	1.00	0.00	8.4206
15	1.00	1.00	0.00	1.00	0.25	0.00	0.50	7.2242
16	0.50	0.00	0.25	1.00	0.00	0.25	0.75	6.0216
17	1.00	1.00	1.00	1.00	0.50	1.00	0.00	5.3495
18	1.00	1.00	0.50	0.25	0.00	1.00	1.00	6.0325
19	1.00	0.00	1.00	0.00	0.75	0.00	0.25	6.4065
20	0.50	1.00	0.75	1.00	0.25	0.75	1.00	5.5674
21	0.25	0.00	0.50	0.25	0.25	0.00	1.00	6.5214
22	0.00	0.50	0.50	0.00	0.75	0.50	0.75	7.7907
23	0.25	0.00	0.00	0.25	0.00	0.75	0.50	7.3542
24	0.75	0.75	1.00	0.00	0.00	0.00	0.00	5.8651
25	0.25	0.00	1.00	0.50	0.75	0.50	0.00	6.4489
26	0.75	1.00	0.25	0.75	1.00	0.75	0.25	7.6225
27	0.00	1.00	1.00	0.50	0.25	1.00	0.50	5.8572
28	1.00	0.50	1.00	0.00	1.00	1.00	0.50	6.5656
29	1.00	0.25	1.00	1.00	1.00	0.25	0.00	5.7137
30	0.00	0.00	0.00	1.00	0.25	1.00	1.00	6.5603
31	0.50	0.00	0.50	0.00	0.75	1.00	0.25	7.5044
32	1.00	0.00	0.75	0.75	0.50	0.75	0.25	5.8721
33	1.00	0.00	0.00	0.00	1.00	0.25	0.00	8.2060
34	1.00	0.50	0.00	1.00	0.00	0.75	1.00	6.3746
35	0.25	0.50	1.00	0.75	0.50	1.00	1.00	5.4478
36	1.00	0.75	0.00	0.00	0.50	1.00	0.75	7.6953
37	0.50	0.00	1.00	0.00	0.00	0.50	0.75	5.3423
38	0.50	1.00	0.00	1.00	0.00	1.00	0.25	6.4493
39	1.00	0.25	0.00	0.50	0.00	0.50	0.00	6.8957
40	0.75	0.50	0.25	0.50	1.00	1.00	1.00	7.5563
41	0.75	0.75	0.25	0.50	0.00	0.25	1.00	6.7549
42	0.75	0.00	0.75	0.75	0.00	1.00	1.00	5.0056
43	0.00	0.25	0.25	1.00	0.75	0.25	0.75	7.4006
44	1.00	0.25	0.75	0.00	0.25	0.00	1.00	5.6656
45	0.50	0.50	0.50	0.50	0.50	0.00	0.00	7.4111
46	0.75	1.00	0.75	0.25	0.25	0.75	0.25	6.7111
47	1.00	0.50	0.25	0.25	0.75	0.75	0.00	7.9182
48	1.00	0.25	0.50	1.00	1.00	1.00	0.50	6.2543
49	0.00	0.75	0.50	0.75	0.25	0.25	0.50	6.7319
50	0.25	1.00	0.25	1.00	0.75	1.00	0.75	6.9749

Rather than doing all 50 runs at once, we decided to run the experiment in a series of stages, starting with 20 runs, then adding 10 at each succeeding stage. The rationale was that we might gain sufficient information at an early stage to avoid the need to run the complete design. To construct the 40-run design (D40), we used D50 as the set of candidate runs and used the design algorithm to choose 40 of them, again using the D-optimality criterion and the same correlation function. The 30-run design (D30) was chosen from D40, and D20 from D30, in a similar way. Apparently, the smaller designs in this sequence do not suffer much with respect to the design criterion. Using as a baseline the best of 10 randomly generated 5-level 20-run designs, we found the gain in log $|C_{DD}|$ achieved by D20 was 99.8% of that achieved by the best 20-run design we obtained from the full 5^7 candidate set.

2.3. Experiment and prediction

The initial experiment on the methane combustion code was based on D20. Each t_j was a coded (centered and scaled) value of the logarithm of the jth reaction rate. After the initial 20 runs of the combustion code were made, predictions were made using standard formulas for conditional normal distributions. The mean and variance of $Y(t)$ given $Y_D = y_D$ are:

$$E[Y(t) \mid Y_D = y_D] = \hat{y}(t) = \mu + C_{tD}C_{DD}^{-1}(y_D - \mu J_n) \tag{2.4}$$

$$V[Y(t) \mid Y_D = y_D] = \hat{v}(t) = \sigma^2(1 - C_{tD}C_{DD}^{-1}C_{Dt}) \tag{2.5}$$

where $C_{Dt} = C_{tD}'$ is the n-vector of prior correlations between $Y(t)$ and Y_D and J_n is an n-vector of 1's. In order to use (2.4) and (2.5), one needs to specify the prior mean μ, the prior variance σ^2 and the correlation function (needed for C_{tD} and C_{DD}). In our approach, we arbitrarily choose a family of correlation functions, indexed by a set of parameters θ, and then use maximum likelihood or cross-validation to select μ, σ^2, and θ. Maximum likelihood, which we use here in the first example, is a well-accepted frequentist method if one views the prior as a model for y (Sacks, Schiller and Welch (1989), Sacks, Welch, Mitchell and Wynn (1989)). It can also be viewed as a form of Bayesian cross-validation (Currin et al. (1991)). We can offer no particularly good reason for preferring maximum likelihood to more traditional cross-validation methods (e.g., the one we use in our second example), except that it takes into account the predictive variances as well as the predictive means, which may or may not be considered an advantage.

For the present example, we choose the piecewise cubic correlation (Currin et al. (1991)):

$$R(d_1, \ldots, d_7) = \prod_{j=1}^{7} R_j(d_j) \tag{2.6}$$

$$R_j(d_j) = 1 - 6\left(\frac{d_j}{\theta_j}\right)^2 + 6\left(\frac{|d_j|}{\theta_j}\right)^3 \qquad |d_j| < \theta_j/2 \tag{2.7a}$$

$$R_j(d_j) = 2\left(1 - \frac{|d_j|}{\theta_j}\right)^3 \qquad \theta_j/2 \le |d_j| < \theta_j \tag{2.7b}$$

$$R_j(d_j) = 0 \qquad |d_j| \ge \theta_j \tag{2.7c}$$

There is no particularly compelling reason to use this instead of some other family of correlation functions. However, the piecewise cubic does have two appealing features: (i) $R_j(d_j)$ decreases to 0 as $|d_j|$ increases to θ_j, so that predictions can be made more local or less local by controlling θ_j, and (ii) $\hat{y}$ is a cubic spline in every t_j if the other t_j's are fixed. (This is because each element of C_{tD}, regarded as a function of t_j, is itself a cubic spline.) Cubic splines are quite highly regarded as interpolators and data smoothers; Bayesian prediction based on (2.6)–(2.7) produces an interpolating cubic spline with very little effort on the part of the user.

In the methane combustion example, we selected values of μ, σ, and $\theta = \{\theta_1, \ldots, \theta_7\}$ by maximum likelihood. We actually have little interest in these parameters for their own sake; we are simply trying to use the data to find a prior that works well for the data at hand. By doing this, we admittedly stray from the Bayesian path and take a turn that some would describe as empirical Bayesian. This is more for pragmatic than philosophical reasons — we have not yet found a natural approach for assigning a prior to θ. The natural "noninformative" prior for μ is the improper uniform prior $p(\mu)d\mu = d\mu$; this in fact leads to the same $\hat{y}$ as that provided by the MLE.

The log likelihood function is:

$$L = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log\sigma^2 - \frac{1}{2}\log|C_{DD}| - \frac{1}{2\sigma^2}(y_D - \mu J_n)' C_{DD}^{-1}(y_D - \mu J_n), \tag{2.8}$$

which depends on θ through C_{DD}. Given θ, this is easy to maximize over μ and σ^2, but maximization over θ requires iterative search — this is by far the most (computer) time-consuming part of the prediction method if θ has many components.

To obtain a measure of predictive error for fixed values of the parameters of the prior process, it is useful to consider the errors in a "leave-one-out" cross-validation. In principle, each of the n experimental runs is deleted, in turn, and the data at the remaining sites are used to predict y at the deleted site. Computationally, this is not as exhausting as it seems, since it can be shown that the error of prediction at the deleted site is

$$e_i = q_i(g_i - \mu w_i) \tag{2.9}$$

where

$$g = C_{DD}^{-1} y_D \tag{2.10}$$

$$w = C_{DD}^{-1} J_n \tag{2.11}$$

and q is the inverse of the diagonal of C_{DD}^{-1}. Here C_{DD} is based on the *full* n-run design. The cross-validation root mean squared error is then:

$$\text{CVRMSE} = \left(n^{-1} \sum_{1}^{n} e_i^2 \right)^{1/2} \tag{2.12}$$

This computation is done with the parameters of the prior process fixed at the values selected (by maximum likelihood, for example) for the full sample. That is, we do not recompute the MLE for every one of the n "leave-one-out" scenarios; this would usually take too much computer time.

The values of CVRMSE at the end of each stage (D20, D30, D40, D50) of the methane combustion experiment were, respectively, 0.21, 0.15, 0.13, 0.19. (The response values in each stage ranged from about 5 to 8.5.) We have no explanation for the increased CVRMSE at the last stage.

In order to find out how well the prediction procedure works in this example, we ran the methane combustion code at two different sets of test sites and compared the true y values with the predictions. The first was a set of 50 sites chosen randomly from a uniform distribution on T. The second was a set of 64 sites at the corners of T; these were chosen to correspond to a half fraction (of resolution VII with generator $I = 1234567$) of the 128 corners. Plots of $\hat{y}(t)$ vs. $y(t)$ for these two sets of test sites are given in Figure 1, where $\hat{y}$ is based on the 50-run experiment D50. For the 50 random sites, where the true response varied from 5.6 to 7.9, the root mean squared prediction error was 0.1212 and the maximum prediction error was 0.3507. For the set of 64 corner sites, where the true response varied from 4.4 to 8.9, the root mean squared error was 0.1680 and the maximum error was 0.5382.

Figure 2 shows the root mean squared error of prediction as a function of n at the two sets of test sites, where the correlation parameters at each value of n were selected by maximizing the likelihood. Also shown for comparison are values corresponding to least squares fits under first order (linear) and quadratic polynomial models. Our predictions are seen to compare favorably with these standard approximation methods, particularly at the corners of T. It is possible that other designs may be more suitable for polynomial fitting. However, we did try a 79-run central composite design, with axial points at the faces of the cube, in an attempt to see how well a standard design for fitting quadratic polynomials would do. A quadratic polynomial was fitted to the data from this design by

the method of least squares. At the 50 random sites, the predictive root mean squared error was 0.1356, and the maximum error was 0.3475. Since these values differ little from those for our design/prediction method with $n = 50$, we regard

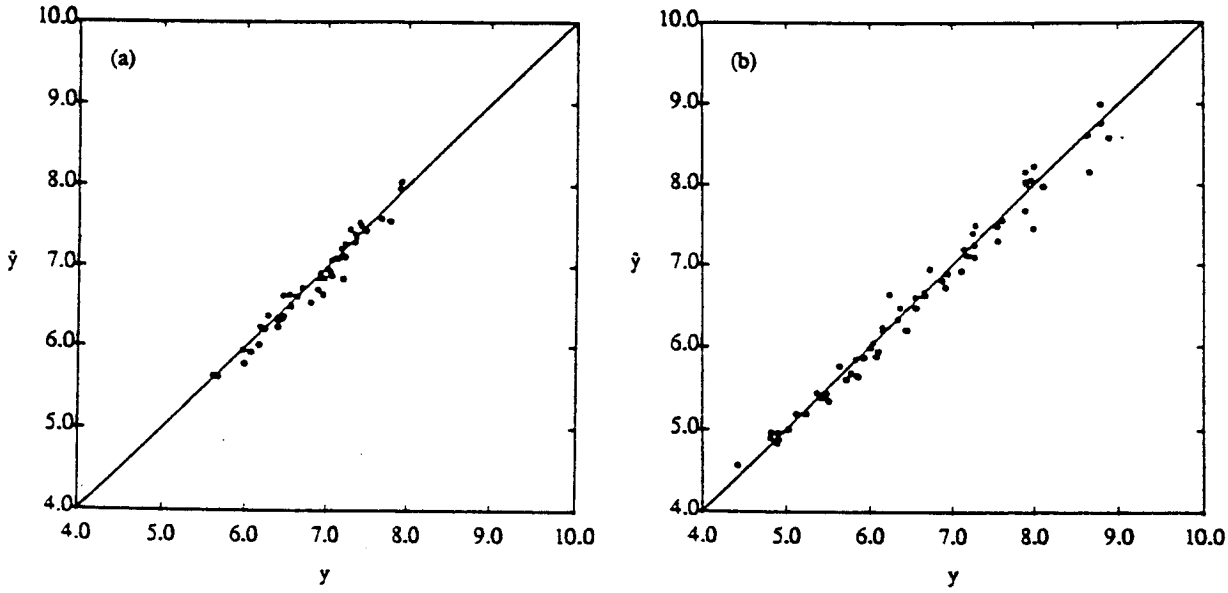

Figure 1. Predictions ($\hat{y}$) vs. true values (y) at (a) 50 random sites and (b) at the 64 sites that correspond to a half-fraction of the corners of the region of interest. Predictions are based on the 50-run design D50.

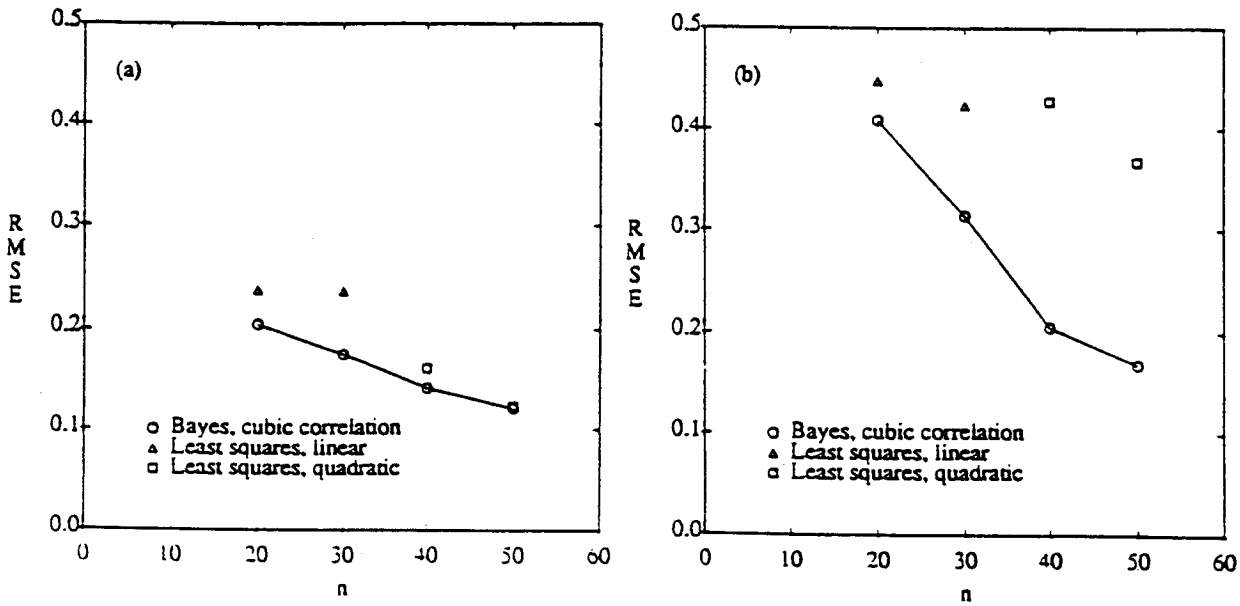

Figure 2. Root mean squared error (RMSE) of prediction vs. n at (a) 50 random sites and (b) at the 64 sites that correspond to a half-fraction of the corners of the region of interest. Bayesian predictions are based on a piecewise cubic correlation function. Also shown are the RMSE values for the least squares fit of linear and quadratic polynomials.

the latter as considerably more efficient here. (We did not do a similar comparison at the corners, since our test set there was used as the "cube" part of the central composite design.)

Sacks, Schiller and Welch (1989) used the same computer model for their Example 2. They considered a region of interest that was centered at the same point but was half as large in each dimension, and obtained very good predictions in this region using a 79-run design. Their root mean squared error at 200 random sites was 0.012, which is about one tenth the value that we obtained over the larger region, using our 50-run design. We cannot say whether this is due to the difference in density of design sites in the region of interest, the nature of the response function over the larger region, or to the difference in the choice of model and design and the method of analysis.

2.4. Estimating features of the response function

Although the coefficients in the expression for $\hat{y}$ are not directly interpretable as features of the response function y (e.g., main effects, interactions, quadratic effects, etc.), it is useful to recognize that such features can be defined as functionals of y, and can therefore be evaluated or approximated using $\hat{y}$. To illustrate this in the present example, we evaluated $\hat{y}(t)$ at the 128 corners of T, i.e. we predicted the results of a 2^7 experiment, and then used these values to compute main effects and interactions in the usual way. For comparison, we also computed main effects and interactions based on the true values of y at our 64 corner test sites, which correspond to a half-fraction of resolution VII. If 5-factor and 6-factor interactions are assumed neglible, then the main effects and two-factor interactions estimated from the resolution VII data will be very nearly correct. As Figure 3 indicates, there is a fairly strong correlation between our predicted effects and those computed from the resolution VII data, even after 20 runs. The largest effects, which are the main effects of t_2, t_3, t_4 and t_5, are accurately identified (taking the resolution VII results as truth). The smaller effects are more difficult to sort out. In fact, the largest two-factor interactions don't really emerge until $n = 50$. As a separate exercise,we added an amount equal to $0.8t_3t_4$ to each observed value of y, redid the analysis for D20, and again predicted main effects and interactions. The purpose was to observe whether the presence of an interaction having about the same magnitude as the largest main effect would be detected. In this case it was, although its predicted magnitude (0.44) was an underestimate. This raises the possibility that the prediction of main effects and interactions, following the design/prediction method presented here, may be an effective tool for screening (i.e., identifying important effects and interactions).

Our examination of main effects and interactions here is based on the predictions at the corner points of T. An Associate Editor has suggested that an

analysis of predictions corresponding to three- or four-level factorial designs (or fractions thereof) would be more informative, and we agree. A more global definition of main effects and interactions, which considers predictions over the entire region,was suggested by Sacks, Welch, Mitchell and Wynn (1989, p.418). Plots of these main effects and interactions, which are functions in one and two dimensions, respectively, can reveal much about the nature of the response, including curvature. This was done to advantage by Welch et al. (1992) for the purpose of screening and prediction.

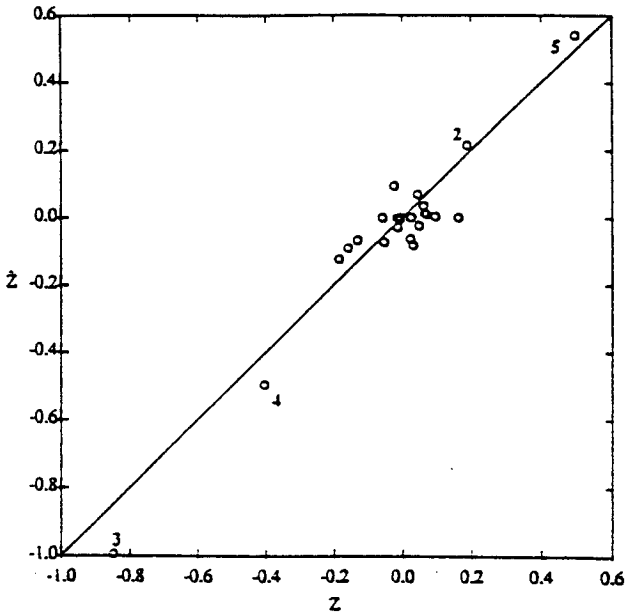

Figure 3. Main effects and two-factor interactions ($\hat{Z}$) obtained from predictions at the 128 corners of the region of interest, using the Bayesian predictor with $n = 20$ runs (design D20) and a piecewise cubic correlation function, plotted against the main effects and interactions (Z) obtained from a 64-run resolution VII experiment. The main effects for variables 2, 3, 4, and 5 are labeled.

This is as far as we took the methane combustion experiment, since our purpose was to demonstrate our approach to the approximation of y by $\hat{y}$. To achieve the ultimate goal of estimating the reaction rates by matching experimental and computed results, we would need to undertake a similar exercise for each of M experiments to obtain prediction functions $\hat{y}_m$ for $m = 1, \ldots, M$. We would then search for t such that $\hat{y}_m(t) \approx y_m$, $m = 1, \ldots, M$, where y_m is the observed ignition delay time in the mth experiment.

3. Example II: Compression Molding Model

3.1. Introduction

This example is concerned with an experiment on a simulation model of the compression molding of an automobile hood. The material used in the manufacture of this part is a sheet molding compound composed of polymer resin, chopped fibers, filler, and additives. Prior to the molding process, a "charge", or piece of this compound, is cut from a sheet and placed in a heated mold. The process is begun by closing the mold slowly; during the process the material flows and fills the mold cavity. After filling, a constant force is maintained on the mold, as the curing reaction proceeds; then the part is removed and the curing is completed.

Designers of the manufacturing process are concerned with the movement of the flow front; it is desirable that the charge fill the mold evenly and rapidly, without the presence of "knit lines" formed when two parts of the flow front meet. To help determine the effect of the design parameters (e.g., the initial shape and placement of the charge) on the flow front movement, a computer simulation model is used. This model is a version of the TIMS (ThIn Mold filling Simulation) model, which was developed by Tim Osswald and Charles Tucker (1990). The version we used came to us through the courtesy of Alonzo Church, Jr. and Daniel Fleming, who were of great help to us in learning to use it and in evaluating the results. The theory and numerical implementation are described in Osswald and Tucker (1990). The inputs to the code include the geometry of the part, the material properties (e.g., viscosity), the closing speed, the final thickness of the part, and the shape and location of the charge. The output consists of all the information needed to predict the position of the flow front as a function of time. The code uses a finite element method to solve a system of differential equations based on the physics of the process. This is not a trivial computation — each run of the model code takes 4-5 minutes on a Cray X-MP computer. For specific, well-defined experiments, it is worthwhile, therefore, to seek a fast approximation to the model; this is the purpose of the exercise we shall describe here. Of special interest to us is the highly multidimensional nature of the response (flow front movement). Previous applications of our prediction method, and of similar methods described by other authors, have been concerned with prediction of a single response computed from the output. Although we shall do nothing more than apply the same prediction method separately to 2345 related responses, we shall see that even this kind of naive approach can be useful.

3.2. Predictors and response variables

In this example, we are concerned only with the effect of the initial shape and location of the charge. The input that defines this is a list of "nodes" (in the finite element discretization of the mold surface) that are filled initially by the charge. There are 469 nodes altogether, and the initial charge typically fills

30 to 40 of them. (Although nodes are actually points, each is associated with a small subvolume of the mold. When we refer to a node as being "filled", we are really referring to this associated subvolume.) In order to represent the list of initially filled nodes by a few predictor variables, we require the initial shape of the charge to be rectangular. The predictor variables are then defined by the boundaries of the rectangle. This is done conveniently using the node map (Figure 4), where the nodes form an approximately uniform grid over the part of the mold where the charge might be placed. The north and south boundaries of the charge correspond to the predictor variables t_1 and t_2, while the east and west boundaries correspond to t_3 and t_4, respectively. (The scaling is such that $0 < t_2 < t_1 < 1$ and $0 < t_4 < t_3 < 1$.) See Figure 4 for an example. For other geometries, of both the charge and the region of the mold into which the charge is to be placed, the representation of the initial shape and location of the charge by a few predictor variables might be considerably more difficult.

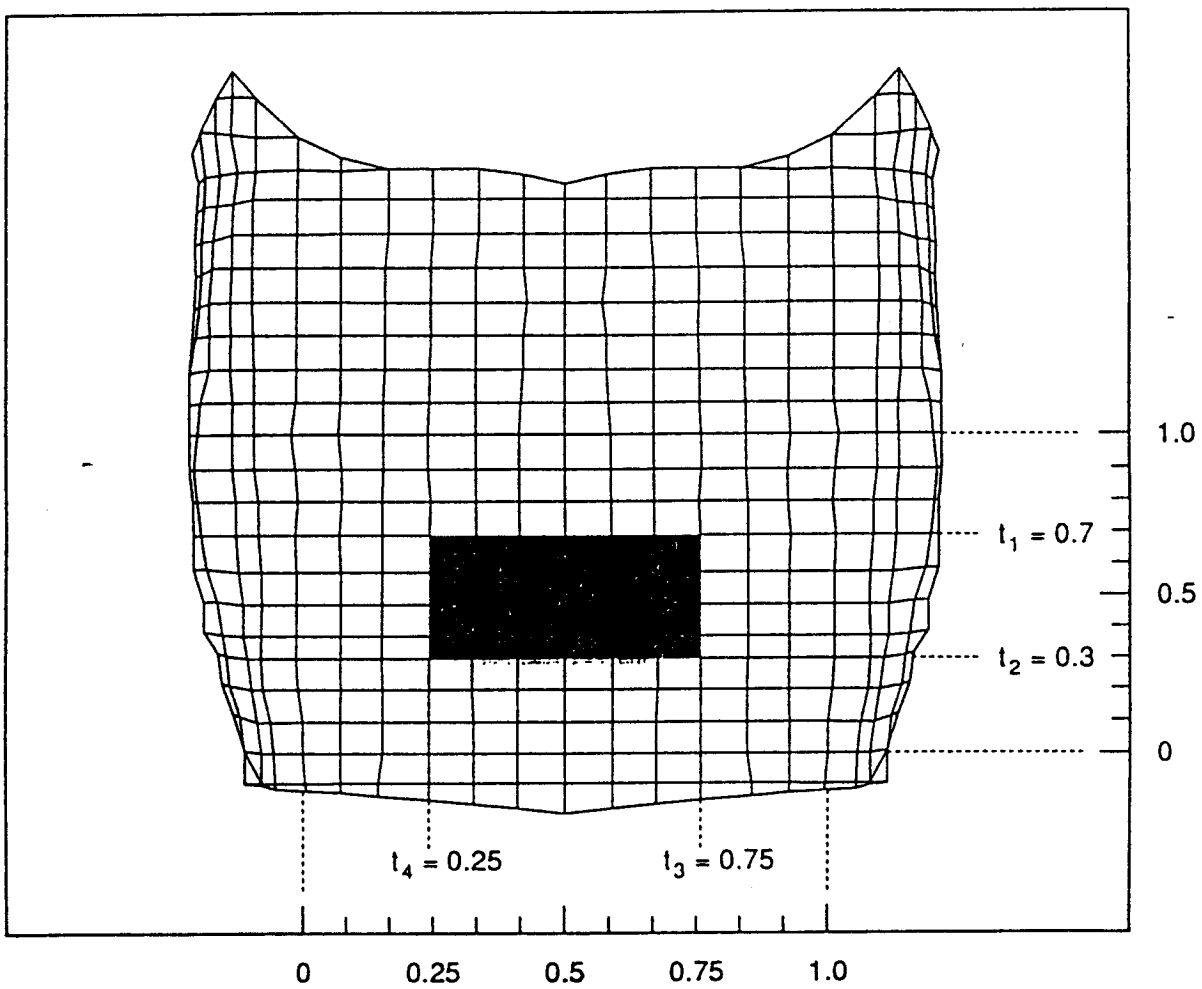

Figure 4. Node map, showing coordinate system for predictor variables t_1, t_2, t_3, t_4.

The next part of the setup of the prediction problem is to define, from the mass of output, a manageable set of response variables that will permit prediction of the flow front. The output gives values of the function $p_m(\tau)$ for all nodes $m = 1, \ldots, 469$ at each time step in the simulation, where $p_m(\tau)$ denotes the proportion of node m that is filled at time τ. Table 2 shows a portion of the output; the full output consists of a sequence of many such tables, each corresponding to a time step.

Table 2. Output of TIMS at a single point in simulated time. The full output is a sequence of many such tables.

Time = 6.37210	
Charge Information	
Node (Cell) Number	Fraction Filled
1	0.00000
2	0.00000
.	.
.	.
.	.
.	.
129	0.06009
130	0.28437
131	0.56793
132	0.69239
133	0.65781
134	0.43808
.	.
.	.
.	.
151	1.00000
152	1.00000
.	.
.	.
.	.
.	.
.	.
469	0.00000

At each node m, we defined the five responses

y_{m1} : the last recorded time at which node m is empty ($p_m(y_{m1}) = 0$),
y_{m2} : the time at which node m becomes 25% full ($p_m(y_{m2}) = 0.25$),
y_{m3} : the time at which node m becomes 50% full ($p_m(y_{m3}) = 0.50$),
y_{m4} : the time at which node m becomes 75% full ($p_m(y_{m4}) = 0.75$),
y_{m5} : the first recorded time at which node m is 100% full ($p_m(y_{m5}) = 1$).

Since these values are not given directly by the output, which gives values of p_m at various times, we approximated them by linear interpolation of the output data. The prediction problem was then taken to be: Approximate the 2345 functions $y_{mr} = y_{mr}(t_1, t_2, t_3, t_4)$, where $m = 1, \ldots, 469$ and $r = 1, \ldots, 5$, over the region defined by $0 < t_2 < t_1 < 1$, $0 < t_4 < t_3 < 1$. Two further practical constraints on the region of ineterest were added. The first restricted the placement of the charge to be symmetric about the north-south center line, i.e., $t_3 + t_4 = 1.0$; see Figure 4. The second required that the number of nodes initially filled by the charge be between 30 and 40; this was our way of implementing a requirement that the area of the mold surface initially covered by the charge be fairly constant.

3.3. Design

The design of the experiment was obtained using our design algorithm for D-optimality with a weak correlation function as described in Section 2. The set of candidate runs was formed by first letting t_1 and t_2 take any of 11 levels (corresponding to the horizontal grid lines in Figure 4), and t_3 and t_4 take any of 13 levels (corresponding to the vertical grid lines). Imposition of the restrictions mentioned just above reduced the number of feasible candidate runs to 41. (If we were doing this problem solely for application, and discovered that there were only 41 scenarios of interest, we would probably run them all using TIMS and be done with it! However, for the sake of demonstration, we shall carry on.)

The initial 10-run design, plus an additional 5 runs that were chosen later, are shown in Table 3. The need for the additional runs was clear after inspection of the cross-validation predictions based on the initial experiment. These runs were chosen using the same algorithm and the same correlation function which generated the first ten runs. The full 15-run design populates the region of interest (which is relatively small here) quite densely; the maximum distance $\sum_{j=1}^{4} |t_j - s_j|$ between any feasible site t not in the design and the closest design site s is 0.2.

Table 3. Design for experiment on compression molding model.

Initial Design

Run	t_1	t_2	t_3	t_4
1	0.40	0.00	0.75	0.25
2	0.40	0.20	1.00	0.00
3	0.80	0.60	1.00	0.00
4	1.00	0.00	0.58	0.42
5	0.80	0.40	0.75	0.25
6	0.60	0.40	0.92	0.08
7	0.50	0.20	0.83	0.17
8	0.70	0.10	0.67	0.33
9	0.90	0.60	0.83	0.17
10	1.00	0.50	0.67	0.30

Additional Points

Run	t_1	t_2	t_3	t_4
11	0.50	0.00	0.67	0.33
12	0.70	0.40	0.83	0.17
13	1.00	0.60	0.75	0.25
14	0.60	0.20	0.75	0.25
15	0.90	0.20	0.67	0.33

3.4. Prediction

The prediction method was essentially the same as that described above in Section 2.3, except for the criterion for selecting the parameters of the prior. Equation (2.4) is the key formula; it was applied to each of the 2345 responses in turn. We allowed μ to differ among responses, but the correlation parameters (θ_j in (2.7)) were taken to be the same for all responses. This was done for reasons of expediency; it would have taken an excessive amount of computation to optimize the choice of the θ's separately for all 2345 responses.

Since $t_3 + t_4 = 1$ throughout the prediction region (due to our requirement of symmetry of the initial charge), we might have omitted one of these variables as a predictor in order to simplify the analysis. This would change the correlations, so we could not expect the same predictions, but we have not determined whether we would have done better or worse by taking this route.

We did add another predictor variable, however. In our first analysis, the cross-validation results at particular nodes indicated that the predictions of y_{mr} tended to be lower than the true values when the area of the charge was smaller than average and higher than the true values otherwise. That is, the predictions had the flow front moving too fast when the area of the charge was relatively small. We assumed that this was due to the increase in the height of the charge when

the area is small (since the volume is held constant), which would presumably result in a slowing of the movement of the front as computed by TIMS. At any rate, we decided to introduce an additional predictor: $t_5 = (t_1 - t_2)(t_4 - t_3)$, which represents the approximate area of the charge.

Predictions were based on the product piecewise cubic correlation function given above in (2.6)–(2.7). To save time in the search for the optimal correlation parameters, we used only one response at each node, namely y_{m3}, the time to 50% filling. This seemed reasonable since we expected the other response functions to be similar in form. The values of μ_{m3}, $m = 1, \ldots, 469$, and θ_j, $j = 1, \ldots, 5$, were chosen to minimize the sum of squared cross-validation errors. Then, fixing the θ's at these values, we determined values of μ_{mr} for all m and r (again by cross-validation), this time using all 5 responses at each node.

We then implemented the prediction equations for all responses in the form of a short computer code "FTIMS", which serves as a fast emulator of TIMS for investigating the effects of changing the shape and location of the charge. The input and output files for FTIMS are of exactly the same form as those for TIMS. The only difference is that the output for FTIMS is based on the prediction equations that followed from the computer experiment we described here, rather than the finite element solution to the differential equations of the model.

FTIMS converts the TIMS input into the site $(t_1, \ldots, t_5)$ at which predictions are desired. The 15×1 vector C_{tD} of correlations between this site and the design sites are computed using the values of θ_j, $j = 1, \ldots, 5$, that we found to be optimal by the cross-validation criterion.

The predictions of the responses y_{mr}, $m = 1, \ldots, 469$, $r = 1, \ldots, 5$, are made using (2.4), where the 15×1 vector $w = C_{DD}^{-1} J_n$ (which is the same for all m, r) is provided by a fixed input file, as is the 15×1 vector $g = C_{DD}^{-1} y_D$ and the scalar μ (both of which depend on m and r). FTIMS then adjusts the five predicted responses at each node, if necessary, to incorporate the knowledge that the true responses are nonnegative and nondecreasing. (We do not expect this adjustment to be needed very often, since the predictions interpolate data that satisfy these requirements. In the test case that we report below, the adjustment was needed at only two of the 469 nodes.) Monotonicity is enforced in a straightforward way, based on the notion that, of the five responses at node m, $\hat{y}_{m3}$ (i.e., the time to 50% filling) is generally the most reliable. This response is therefore left unchanged, and $\hat{y}_{m2}$ and $\hat{y}_{m4}$ are adjusted, if necessary, so that $\hat{y}_{m2} \leq \hat{y}_{m3} \leq \hat{y}_{m4}$. Keeping these three predicted responses constant, $\hat{y}_{m1}$ and $\hat{y}_{m5}$ are adjusted similarly.

To convert the five predicted responses at each node into estimates of $p(\tau)$ at the values of time desired, FTIMS again uses linear interpolation. The results are then printed in exactly the same form as the output produced by TIMS (Table

2). The postprocessor that normally runs on TIMS output can then be applied to the output of FTIMS. This produced Figure 5, which shows the position of the flow front seven seconds after the start of the simulation, as computed by TIMS and FTIMS for a test case in which $t_1 = 0.7$, $t_2 = 0.3$, $t_3 = 0.75$, and $t_4 = 0.25$. The predicted front is seen to be just a little ahead of the true front in this case. Plots similar to Figure 5 were made at several different time points, and the predicted flow front matched that computed by TIMS quite faithfully. On average, the predicted time to 50% filling in this case was 0.14 seconds less than the time calculated by TIMS; the root mean squared error for $\hat{y}_{m3}$ over all nodes was 0.23 seconds. In seven other randomly chosen test cases, the root mean squared error for $\hat{y}_{m3}$ over all nodes varied from 0.01 sec to 0.68 sec, with a median of 0.27 sec. In these test cases, the "true" times to 50% filling, averaged over all nodes, varied from 6.4–9.1 seconds.

The range of applications of the current version of FTIMS is obviously quite limited. Further generalizations, modifications, and tests would need to be made before it could be considered a practical tool for optimizing this particular sheet molding process. Even at that stage, we would regard FTIMS as only an occasional replacement for TIMS, when one wants to consider many scenarios quickly and one is willing to accept an approximate result. The computing time for the single run of FTIMS in the first test case described above was about 43 seconds on a Sun 3/50 Workstation, only 5 seconds of which were used to compute the predicted response vector at each node. The rest of the time was used for input and output. We have already noted that each run of TIMS takes 4–5 minutes on a Cray X-MP, so the availability of a practical and well-tested version of FTIMS would permit more extensive exploration of the effects of shape and position of the charge on the movement of the flow front.

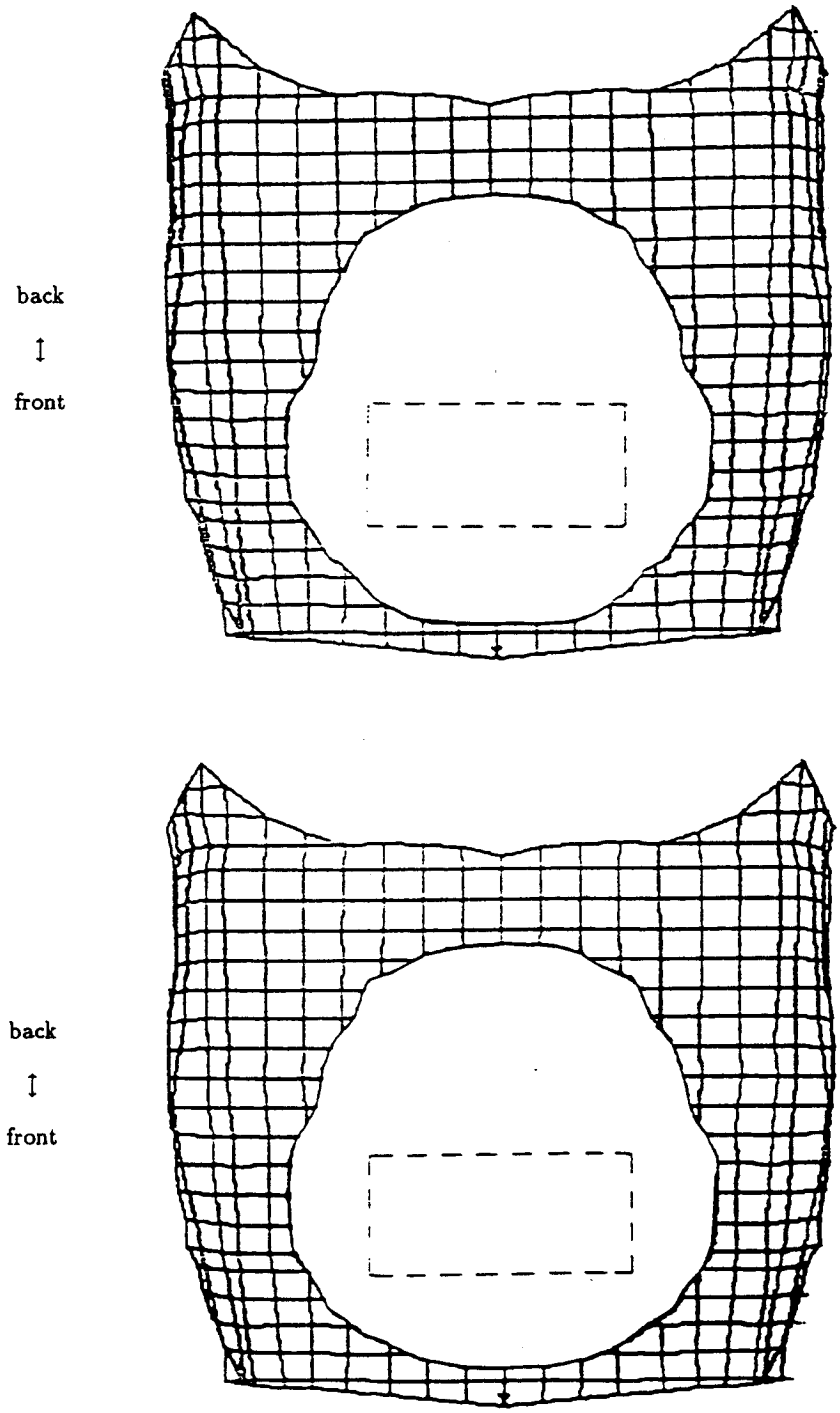

Figure 5. Position of flow front 7 seconds after start of simulation, as computed by TIMS (lower) and FTIMS (upper). Top view of hood is shown, with nodes indicated as in Figure 4. Dashed line shows position at start of simulation.

Acknowledgements

This work was sponsored by the Applied Mathematical Sciences Research Program, Office of Energy Research, U.S. Department of Energy Contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc. We are indebted to Prof. Michael Frenklach of Penn State University for allowing us to use the methane combustion code, to Prof. Jerome Sacks of the University of Illinois for permission to use his version of it, and to Robert Buck of the University of Illinois for sending it to us and helping us learn how to use it. Similarly, we are grateful to Prof. Charles Tucker of the University of Illinois for allowing us to use the compression molding code (TIMS), to Dr. Alonzo Church of GenCorp Research for permission to use GenCorp's version of it, and to Dr. Daniel Fleming of GenCorp Research for sending us an executable version and helping us learn how to use it. Dr. Fleming also provided us with the node map in Figure 4 and the plot of the flow front in Figure 5.

References

Box, G. E. P. and Draper, N. R. (1987). *Empirical Model-Building and Response Surfaces.* John Wiley, New York.

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. Amer. Statist. Assoc.*, **86**, 953–963.

Frenklach, M. and Rabinowitz, M. (1989). Optimization of large reaction systems. *Proc. of 12th IMACS World Congress on Scientific Computation,* **3** (Edited by R. Vichnevetsky, P. Borne and J. Vignes), 602–604. Gerfidn, Cedex, France.

Johnson, M., Moore, L. and Ylvisaker, D. (1990). Minimax and maximin distance designs. *J. Statist. Plann. Inference* **26**, 131–148.

Mitchell, T. J. (1974). An algorithm for the construction of 'D-optimal' experimental designs. *Technometrics* **16**, 203–210.

Osswald, T. A. and Tucker, C. L. (1990). Compression mold filling simulation for non-planar parts. *Internat. Polymer Process.* **5**, 79–87.

Sacks, J., Schiller, S. B. and Welch, W. J. (1989). Designs for computer experiments. *Technometrics* **31**, 41–47.

Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statist. Sci.* **4**, 409–422. Comments and rejoinder: 423–435.

Shewry, M. C. and Wynn, H. P. (1987). Maximum entropy sampling. *J. Appl. Statist.* **14**, 165–170.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J. and Morris, M. D. (1992). Screening, predicting, and computer experiments. *Technometrics* **34**, 15–25.

Engineering Physics and Mathematics Division, Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831–6367, U.S.A.

(Received December 1990; accepted February 1992)