

Using Computer Experiments to Construct a Cheap Substitute for an Expensive Simulation Model

CONF-9104300--1

Toby Mitchell and Max Morris*
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6367

DE92 000421

Abstract

There is widespread use of computer models as tools in scientific research. As surrogates for physical or behavioral systems, such models can be subjected to experimentation, the goal being to predict how the corresponding real system would behave under certain conditions. For long-running (expensive) model codes, there may be a severe limitation on the number of experiments that can reasonably be done. This motivates the construction of a fast-running (cheap) approximation to the original code, for use in experiments where a large number of runs may be necessary. Here we discuss our approximation of a simulation model for the compression molding of sheet molding compound, applied to the manufacture of an automobile hood. The approximation was constructed using Bayesian interpolation methods for prediction of the movement of the flow front. The predictions were based on data generated by a sequence of computer experiments, using designs chosen according to a type of D-optimality criterion.

1 Introduction

The purpose of this paper is to demonstrate the application of Bayesian methods for design and analysis of computer experiments to the construction of a "cheap" substitute for an "expensive" computer model. As our example, we shall use a computer simulation model for a compression mold-filling process that is used in the manufacture of automobile hoods. Our primary use of this model 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 this account is best considered as a realistic example rather than a complete scientific application.

*Research 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.

The submitted manuscript has been authored by a contractor of the U.S. Government under contract No. DE-AC05-84OR21400. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

Except for some 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 details of the method (e.g., choice of correlation function, design criterion) are more in line with Currin, Mitchell, Morris, and Ylvisaker (1991).

2 The Computer Model

Sheet molding compound (SMC) is composed of polymer resin, chopped fibers, filler, and additives. Prior to the molding process, a "charge", or piece of SMC, 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 of the Department of Mechanical Engineering at the University of Illinois. The version we used came to us through the courtesy of Alonzo Church, Jr. and Daniel Fleming of GenCorp Research, 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

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

for

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 Predictors and Responses

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 as constructed for the finite element method, 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 $0 < t_4 < t_3 < 1$.) 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.

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 for all

nodes $m = 1, \dots, 469$ at each time step in the simulation, where $p_m(\tau)$ denotes the proportion of node m that is filled at time τ .

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 $m = 1, \dots, 469$ and $r = 1, \dots, 5$, 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 interest were added. The first restricted the placement of the charge to be symmetric about the north-south center line, i.e., $t_3 + t_4 = 1.0$. The second required that the number of the 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.

4 Design

The central idea (which is not original with us) is to represent uncertainty about each function y_{mr} on the k -dimensional region of interest T by means of a stochastic process (random field) Y_{mr} . For simplicity and convenience, we use stationary Gaussian (normal) processes as priors. These are fully described by a constant $\mu_{mr} = E[Y_{mr}(t)]$, a constant $\sigma_{mr}^2 = V[Y_{mr}(t)]$, and a correlation function R_{mr} , where $R_{mr}(d) = \text{Corr}[Y_{mr}(t+d), Y_{mr}(t)]$ and where $t = (t_1, \dots, t_k)$ and $t+d = (t_1+d_1, \dots, t_k+d_k)$ are any two "sites" (points in T) separated by a difference vector d . For simplicity, we also take the 2345 Y_{mr} processes independent of one other, and $R_{mr}(d) = R(d)$ for all (m,r) . (The choice of independence is made at the cost of ignoring information about the relationships among the y_{mr} 's at any site. We have not found it feasible to

implement such information here.)

For a design criterion, we use the "maximum entropy" principle (Lindley 1956), which in this case leads to a kind of D-optimality, namely, the maximization of $|C_{DD}|$, where C_{DD} is, for any one of the processes Y_{mr} , the $n \times n$ matrix of *prior* correlations among the design sites (Shewry and Wynn 1987). We find this criterion appealing, for reasons given by Currin et al. (1991), but other criteria could be used. (See, e.g., Sacks, Schiller and Welch 1989 and Sacks, Welch, Mitchell, and Wynn 1989.)

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:

$$R(d) = e^{-\theta \sum |d_j|} \quad (4.1)$$

where θ is "large". Asymptotically (as $\theta \rightarrow \infty$), it can be shown that the D-optimality criterion, where (4.1) 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.

Here the set of candidate runs was formed by first letting t_1 and t_2 take any of 11 levels and t_3 and t_4 take any of 13 levels, subject to the restrictions on the region of interest noted above.

The initial 10-run design, plus an additional 5 runs that were chosen later, are shown in Table 1.

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

Table 1. Design for experiment on compression molding model.

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.

5 Prediction

Predictions were made using standard formulas for conditional normal distributions. Let $y_{mr,D}$ be the vector of the n observed values of y_{mr} . The mean of $Y_{mr}(t)$ given $Y_{mr,D} = y_{mr,D}$ is:

$$\hat{y}_{mr}(t) = \mu_{mr} + C_{iD} C_{DD}^{-1} (y_{mr,D} - \mu_{mr} J_n) \quad (5.1)$$

where C_{iD} is a row vector that holds the n prior correlations between $Y_{mr}(t)$ and $Y_{mr,D}$ and J_n is the column vector composed of n 1's. In order to use (5.1), one needs to specify the prior mean μ_{mr} and the correlation function (needed for C_{iD} and C_{DD}). In our approach, we arbitrarily chose a family of correlation functions, indexed by a set of parameters θ , and then used cross-validation to select μ_{mr} and θ .

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

$$R(d_1, \dots, d_k) = \prod_{j=1}^k R_j(d_j), \quad (5.2)$$

where k is the number of predictor variables, and

$$R_j(d_j) = 1 - 6\left(\frac{d_j}{\theta_j}\right)^2 + 6\left(\frac{|d_j|}{\theta_j}\right)^3, \quad |d_j| \in I_1 \quad (5.3a)$$

$$R_j(d_j) = 2\left(1 - \frac{|d_j|}{\theta_j}\right)^3, \quad |d_j| \in I_2 \quad (5.3b)$$

$$R_j(d_j) = 0 \quad |d_j| \in I_3, \quad (5.3c)$$

where $I_1 = [0, \theta_j / 2]$, $I_2 = [\theta_j / 2, \theta_j]$, and $I_3 = [\theta_j, \infty]$.

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) $R(d_j)$ 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_{iD} , 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 (5.2)-(5.3) produces an interpolating cubic spline with very little effort on the part of the user.

To select the parameters by "leave-one-out" cross-

validation, 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 for response m,r at the deleted site i is

$$e_{mr,i} = q_i (\mathbf{g}_{mr,i} - \mu_{mr} w_i)$$

where

$$\mathbf{g}_{mr} = C_{DD}^{-1} y_{mr,D}$$

$$w = C_{DD}^{-1} J_n$$

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:

$$\text{CVRMSE} = \left[\frac{1}{2345n} \sum_{i=1}^n \sum_{m=1}^5 \sum_{r=1}^5 e_{mr,i}^2 \right]^{1/2} \quad (5.4)$$

Given the θ_j 's, this is easy to minimize over the μ_{mr} 's, but minimization over the θ_j 's requires iterative search -- this is by far the most (computer) time-consuming part of the prediction method.

To save time in the search for the optimal correlation parameters (θ_j 's), 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} , $m = 1, \dots, 469$, and θ_j , $j = 1, \dots, 4$, were chosen to minimize (5.4) with $r=3$ and a divisor of $469n$. Then, fixing the θ_j '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.

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, and we repeated the analysis. This reduced the cross-validation errors, so the area was used as a predictor in all subsequent predictions.

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

The predictions of the responses y_{mr} , $m = 1, \dots, 469$, $r = 1, \dots, 5$ are made using (5.1), 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 $g_{mr} = C_{DD}^{-1} y_{mr,D}$ and the scalar μ_{mr} . 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. The postprocessor that normally runs on TIMS output can then be applied to the output of FTIMS. This produces plots of the position of the flow front at various times. In a test case in which $t_1 = 0.7$, $t_2 = 0.3$, $t_3 = 0.75$, and $t_4 = 0.25$, examination of these plots showed the predicted front to be just a little ahead of the true front. 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 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.

Acknowledgements

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.

References

- 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. Assn.*, to appear.
- Johnson, M., Moore, L. and Ylvisaker, D. (1990). Minimax and Maximin Distance Designs. *J. Statist. Planning and Inf.* 26, 131-148.
- Lindley, D. V. (1956), On a Measure of the Information Provided by an Experiment. *Ann. Math. Statist.* 27, 986-1005.
- 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. *Int. Polymer Processing* 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. Stat.* 14, 165-170.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

END

**DATE
FILMED**

11/10/191

