James M. Lucas
J. M. Lucas and Associates
5120 New Kent Road
Wilmington, DE 19808

REFERENCES

Anbari, F. T., and Lucas, J. M. (1994), "Super Efficient Designs," in *ASQC 48th Annual Quality Congress Proceedings*, Milwaukee: ASQC, pp. 852–863.

Coleman, D. E., and Montgomery, D. C. (1993), "A Systematic Approach to Planning for a Designed Industrial Experiment," *Technometrics*, 35, 1–12.

Crosier, R. B. (1991), "Some New Three-Level Response Surface Designs," CRDEC-TR-308, U.S. Army Chemical Research Development and Engineering Center, Aberdeen Proving Ground, MD 21010-5423.

——— (1993a), "Symmetrical Orientation for Simplex Designs," ERDEC-TR-091, Edgewood Research Development and Engineering Center, Aberdeen Proving Ground, MD 21010-5423.

——— (1993b), "Method for Design Rotation," ERDEC-TR-099, Edgewood Research Development and Engineering Center, Aberdeen Proving Ground, MD 21010-5423.

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

DuMouchel, W., and Jones, B. (1994), "A Simple Bayesian Modification of D-Optimal Designs to Reduce Dependence on an Assumed Model," *Technometrics*, 36, 37–47.

Easterling, R. G. (1989), Comment on "Design and Analysis of Computer Experiments," by J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Statistical Science*, 4, 425–427.

Lucas, J. M. (1976), "Which Response Surface Design is Best," *Technometrics*, 18, 411–417.

——— (1978), Discussion of "D-Optimal Fractions of Three-Level Factorial Designs," by T. J. Mitchell and C. K. Bayne, *Technometrics*, 20, 381–382.

——— (1989), "Achieving a Robust Process Using Response Surface Methodology," in *Proceedings of the Sesquicentennial Invited Paper Sessions, American Statistical Association*, pp. 579–593.

——— (1990), "Letter to the Editor: Comments on Cook and Nachtsheim," *Technometrics*, 32, 363–364.

——— (1994), "How to Achieve a Robust Process Using Response Surface Methodology," *Journal of Quality Technology*, 26, 248–260.

Morris, M. D., Mitchell, T. J., and Ylvisaker, D. (1993), "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction," *Technometrics*, 35, 243–255.

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," with discussion, *Statistical Science*, 4, 409–435.

Welch, W. J., Yu, T. K., Kang, S. M., and Sacks, J. (1990), "Computer Experiments for Quality Control by Parameter Design," *Journal of Quality Technology*, 22, 15–22.

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.

RESPONSE TO JAMES M. LUCAS

Lucas argues that one can take methods for the design and analysis of physical experiments, in which Lucas has considerable experience, and apply the same methods to deterministic computer experiments. We shall argue that alternative methodologies do about as well in simple situations in which simple methods are adequate and perform much better when the response relationship is nonlinear, as it often is in a computer experiment. We first discuss Lucas's example and then address his general remarks.

When trying to demonstrate the effectiveness of a method, it is tempting to simulate data from the assumed model. Lucas succumbs to this temptation. He generates data from a very simple model, then analyzes with full knowledge of the same model. Because the model has only bilinear interactions—that is, no nonlinearities—a two-level design is adequate. Not surprisingly, this method produces perfect prediction! Describing the two-level design as "optimum" is just saying, "If you already know the answer, this methodology is guaranteed to find it!"

In Welch et al. (1992) an equivalent temptation would have been to simulate data from the stochastic-process model underlying our predictor. We felt, however, that readers would be more convinced by generating responses from a real, circuit-simulation computer code (definitely not a realization of a stochastic process) and showing that our method worked well, *without major assumptions about the form of the response function.*

Lucas's example is certainly not typical of the real computer codes we have experienced. He notes that, because of wide factor ranges, asymptotes—that is, nonlinearities—are to be expected and that second-order polynomials can give poor results. Yet Lucas's model does not include nonlinear terms, nor would his design find even quadratic second-order terms. His example response surface is complex through the presence of many interactions. In our experience, although interactions can be present, computer codes tend to be complex through nonlinearities rather than through interaction.

Even though Lucas's model is very unrealistic, it is a fair question to ask how our method performs. The two-level factorial design is very inappropriate for fitting our stochastic-process model (or for fitting other models, we shall argue). In each input variable, x_j, we use the correlation function $R(d) = \exp(-\theta d^p)$, a function of the distance d between two values of x_j. A two-level design always gives $d = 0$ or $d = 2$ for all of the design points. We already know that $R(0) = 1$, so the design gives one value of d to fit a *function.* In addition, 12 parameters (10 correlation parameters, the stochastic-process variance, and an intercept) are being fitted with 16 observations. Most statisticians would beware of using maximum likelihood here, a method that relies in general on asymptotic theory for its optimality properties. Not surprisingly, maximum likelihood estimation is unreliable here.

The unreliability of maximum likelihood estimation is well diagnosed by leave-one-out cross-validation. Suppose that we fix $p_j = 2$ for $j = 1, \ldots, 5$, a value that would usually imply an assumption of smoothness for the response surface. Here, with only two levels in the design, p_j is irrelevant. Setting θ_j $(j = 1, \ldots, 5)$ to 10, 1, .1, .01, or .001 gives cross-validation root mean squared errors varying from about 3.4 to 5.1—that is, large errors. The R_p^2 criterion preferred by Lucas is around 0 or negative. We get a warning that model fitting is problematic. Lucas gets no such diagnostic when he fits an intercept, five main effects, and 10 two-factor interactions with 16 runs. A perfect fit is

guaranteed even if the fitted model is a poor approximation to the true function. With one observation removed, fitting, and hence cross-validation, is impossible.

It is quite revealing to decompose the predictor from the stochastic-process model into main effects, two-factor interactions, and so forth as described at the end of Section 1 of Welch et al. (1992). As Lucas points out, for large values of the θ_j's the predictor reproduces the two-factor-interaction data at the half-fraction design points and is 0 (i.e., a different relationship) on the complementary half fraction. This indicates interactions between more than two factors at a time. On the other hand, if we set $\theta_j = .001\ (j = 1, \ldots, 5)$, then the 5 one-factor main effects account for none of the variability in the predictor, whereas each of the 10 two-factor interactions accounts for approximately 10%—that is, the right conclusion. Moreover, the estimated joint effects are almost exactly right. For example, the joint effect of x_1 and x_2, obtained by averaging $\hat{y}(\mathbf{x})$ over $x_3 = \pm 1, x_4 = \pm 1$, and $x_5 = \pm 1$, is given in Table 1. The other nine estimated joint effects are the same up to a sign change.

Thus, with small values of the θ_j's, the fitted stochastic-process model says that data are explained by 10 two-factor interactions (and the predictor is nearly perfect). With larger values of the θ_j's the fitted predictor has higher-order interactions. This is exactly the same conclusion as Lucas's analysis! His two-level, 16-run design aliases main-effects and two-factor interactions with higher-order effects. No method can overcome this deficiency in the design. The stochastic-process model at least warns us through cross-validation that many fitted models are roughly equivalent. That small values for $\theta_j\ (j = 1, \ldots, 5)$ give excellent prediction here has some theoretical backing. Ongoing work by Y. B. Lim, J. Sacks, W. J. Studden, and W. J. Welch considers $p_j = 2$ and $\theta_j \to 0$ for all input variables. Under these conditions, the predictor can interpolate polynomials exactly up to the degree allowed by the number of runs and the design. In other words, where a simple polynomial model suffices, the stochastic-process predictor can still do well.

We would recommend more than 16 runs to investigate five input variables in a computer experiment if computing resources allow. We would also recommend against two-level designs. A data-adaptive, nonparametric predictor such as ours cannot reveal nonlinearities in the response relationship from such a design. For fitting Lucas's model, even if the true relationship is approximately the same, the bias from a two-level design can be reduced. In a computer experiment there is no variance from random error. A fitted regression model gives prediction error only through bias from model inadequacy. Box and Draper (1959, 1963) considered designs for minimizing the integrated squared bias. Suppose that the input variables are continuous, typically the case in a computer experiment, and that the integration is uniform. Box and Draper showed that a sufficient condition for minimizing the integrated squared bias is to match certain moments of the design with those of the uniform distribution. This result holds quite generally. Figure 1 shows the projection onto x_1 and x_2 of a design for five factors with 24 runs. It is a Latin hypercube (McKay, Conover, and Beckman 1979), so the five input variables are each covered uniformly. Within the class of Latin hypercubes, this design was chosen to maximize the minimum distance between pairs of design points, the maximin criterion proposed by Johnson, Moore, and Ylvisaker (1990) for deterministic computer experiments. Their maximin criterion is adapted such that distances are computed for two-dimensional projections, ensuring fairly uniform coverage for all two-dimensional projections. A space-filling design with a uniform distribution of points, like that in Figure 1, would lead to less model-inadequacy bias when fitting Lucas's model if the model is approximately correct.

Table 1. Estimated Joint Effect $\hat{y}(x_1, x_2)$ for the 16-Run, Fractional-Factorial Design, With $p_j = 2$ and $\theta_j = .001\ (j = 1, \ldots, 5)$

x_1	x_2	$\hat{y}(x_1, x_2)$
−1	−1	.998
−1	1	−.998
1	−1	−.998
1	1	.998

NOTE: The true joint effect takes values ±1.

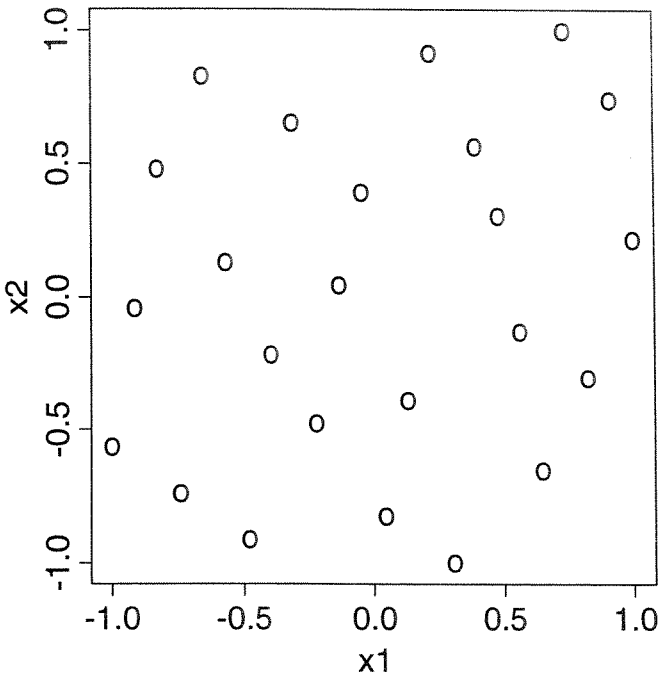

Figure 1. Space-Filling Design for Five Factors and 24 Runs, Projected Onto x_1 and x_2.

With 24 runs, maximum likelihood estimation of the correlation parameters in the stochastic-process predictor is still unreliable, but cross-validation is now conclusive. The predictor based on maximum likelihood estimates of the correlation parameters has an R_p^2 value of about .75—that is, a poor predictor. Putting $\theta_j = .0001\ (j = 1, \ldots, 5)$, however, gives a cross-validation root mean squared error of about .0037 or R_p^2 of about 99.999%. Thus, with $n = 24$ runs cross-validation identifies a near-perfect predictor. The estimated joint effect of x_1 and x_2, for example, is shown in Figure 2. It is very close to the true joint effect, $x_1 x_2$,

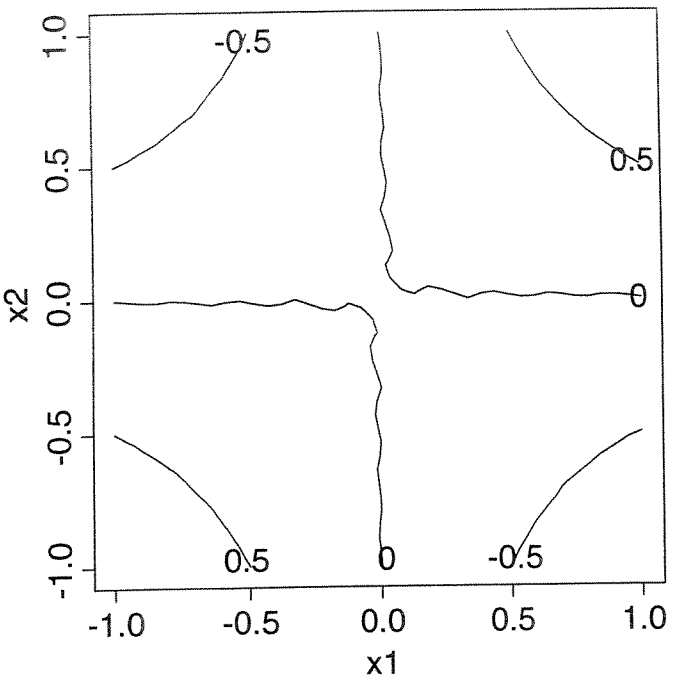

Figure 2. Estimated Joint Effect $\hat{y}(x_1, x_2)$ for the 24-Run, Space-Filling Design, With $p_j = 2$ and $\theta_j = .0001$ ($j = 1, \ldots, 5$). The true joint effect is x_1x_2.

over the entire input-variable space. Thus, with this 24-run design, we are able to predict the function well throughout the space, not just at the vertices, and cross-validation demonstrates the reliability of the predictor.

With a 32-run space-filling design, maximum likelihood chooses very small values for the θ_j's and automatically finds a near-perfect predictor.

To summarize Lucas's example:

1. Two-level designs are poor for computer experiments. It is full knowledge of the correct model, not a good design, that leads Lucas to a perfect predictor here. Fitting Lucas's model to data from a 16-run space-filling design also produces a perfect predictor! Even if the true function is *approximately* linear, bias can be reduced by covering the space more uniformly.

2. Using the 16-run, two-level design, his analysis and our predictor lead to essentially the same conclusion. The data can be explained either by 10 two-factor interaction terms *or* by higher-order interactions. Cross-validation cannot distinguish the two possibilities.

Thus, Lucas's example, apparently chosen to show the advantages of his method, shows no substantive improvement. Compare this example with the two examples of Welch et al. (1992), which are closer to real computer codes, at least in our experience. They have nonlinearities, moderate interaction, and high dimensionality (20 input variables). The methodology espoused by Lucas is woefully inadequate to deal with these examples.

We now discuss some of the more general remarks.

If R_p^2 conveys useful information, then use it. In our experience, however, engineers specify the required accuracy of prediction in terms of absolute or relative error.

Lucas says that our approach "spends little time in the planning stage." To carry out a traditional experimental design and analysis, one has to question engineers on the possibility of nonlinearities, interactions, and so forth at the design stage. Nonlinearities of unknown form cannot be modeled later if the design has only two levels, for example. Of course, we would certainly agree that prior knowledge should be used when available, and this is often much more easily incorporated into our techniques than with classical regression methods. Computer codes are internally complex, and inputs (and outputs) often have high dimensionality. Under these conditions it is unrealistic to expect definitive answers to these questions. Often, engineers are surprised at the results of an experiment. From working with scientists and engineers in various application areas, we have typically found it more fruitful to pay attention to

1. The input variables and their ranges. Computer experiments often have very large ranges. In our experience computer codes can become numerically unstable if ranges are too wide.

2. Transformation of the input variables. If we see an input ranging over several orders of magnitude, we discuss the possibility of logarithmically spacing that input in the design.

3. Parameterization. For example, in an article by Yu, Kang, Sacks, and Welch (1991), transistor widths were the inputs to a circuit-simulation code, but working with the ratio of widths was suggested by engineering knowledge for one pair of transistors.

4. Objectives, particularly trade-offs when they conflict. Many computer experiments aim to optimize an engineering system. Sequential design, adapting the input ranges as we learn about promising subregions of the input space, is particularly useful (e.g., Bernardo et al. 1992).

What is carefully planned about Lucas's half fraction and model? The design is potentially disastrous if the relationship is not linear. We are sure that Lucas never wants to see in real applications a model like the one he fits in his example. A model with no main effects, dominated by interactions, suggests poor parameterization.

Despite the quote from Easterling (1989), a Latin hypercube is not pure random-number generation. The one-dimensional margins are controlled so that we get plenty of levels and nonlinearities can be modeled. As previously, one can take a criterion to improve coverage in two- or three-dimensional projections rather than completely randomizing higher-order projections. See also Morris and Mitchell (1995). Even Latin hypercubes that randomize structure for projections of two or more dimensions probably do very well for most applications. Randomization of run order, for example, is irrelevant in a computer experiment, but randomization is also used extensively in design of experiments to deal with uncertainty about *unknown structure*. Since when is randomization a sin? In our opinion, mindlessly assuming that the relationship is linear, running a two-level design, and hoping for the best offers much more potential for disastrously misleading conclusions. The de-

sign suggested by Lucas for six factors, a composite of two-level and five-level designs, would probably also do well in many applications. It is unfortunate he did not use such a design in his example. Moreover, he gives no clues about how he would deal with 20 or more input variables, the main thrust of Welch et al. (1992).

Lucas raises the interesting question of how to choose the number of runs in a computer experiment. We suggest taking 10 times the number of active inputs (admittedly a guess). Obviously more or fewer runs will be needed depending on the amount of nonlinearity, interaction, and so forth. In most applications this leads to fairly automatic model fitting via maximum likelihood. After a first-stage experiment, accuracy of the stochastic-process predictor can be assessed by cross-validation. If accuracy is not good enough, it is very easy to augment a space-filling design, and a data-adaptive predictor will adapt to the new runs. In contrast, bias arising from an incorrect regression model is difficult to assess, and the bias cannot be reduced below a certain level without introducing a better model.

Lucas's comment that Sacks, Schiller, and Welch (1989), Sacks, Welch et al. (1989), Currin et al. (1991), and Morris, Mitchell, and Ylvisaker (1993) were concerned with design for "best estimates of the assumed correlation functions" is just wrong. In these works, designs were chosen to minimize prediction error given the correlation parameters.

Our predictor does require estimation of the correlation function. In this sense the *estimated* correlation function depends on the data. Typically, uncertainty in estimating the correlation parameters is ignored in theory and in computations. Even if this is discounted, there is a further source of uncertainty in the predictor from interpolating between design points. The latter source tends to dominate, at least if an adequate number of runs is taken. As illustrated by the examples of Welch et al. (1992), cross-validation is often a good indicator of predictive accuracy at new points. Analysis of the estimated correlation function rarely helps understanding of the computer code. Rather, we tend to use visualization of the estimated effects.

Lucas's comment, "Having only a few active factors ... raises questions about ... the complexity of the ... code" (p. 192) suggests some unfamiliarity with computer codes. Typically a code produces several outputs. (In fact, the output may include a function, for example, over time from which summaries are extracted.) Even though an input is inactive for some outputs, it might be active for others, and overall the code is complex. Moreover, in our experience, the experimenter running a computer code is often not the code's author. Code developers often do not know which factors are important until they complete and run the code: They are coding basic physics one component at a time.

Rational polynomials might be useful. Again, though, we prefer to make few assumptions about the form of the relationship. In most applications, we suspect, the experimenter is unable to specify the form.

One advantage of our methodology is that it is fairly automatic once the preceding planning decisions have been made. Lucas points out that our methodology uncovers structure in the simple example in Section 1 of Welch et al. (1992). We agree that other methods might work well after transformations, and so forth (though a two-level design would again be inadequate). Lucas's attitude seems to be that scientists and engineers should immerse themselves in several courses on design of experiments, regression diagnostics, and so forth. Of course, science and engineering would benefit immensely, but the reality is that much work goes on with little statistical expertise.

Science and engineering need methods of analysis that automatically work well in most cases without agonizing over the form of the regression model. We have worked on applications with about 40 input variables and 10 outputs of interest. Under these conditions, engineers welcome automation! Commenting on trade-offs in conflicting objectives, as indicated by visualization of the predictor, is a more profitable and a more realistic use of engineering knowledge. Some situations will require expert statistical advice, so there will always be work for specialists like Lucas.

Lucas's attitude is further revealed by his attack on computer-aided design. Cook and Nachtsheim (1990), in their response to Lucas (1990), pointed out that Lucas wants to change the experimenter's objectives to fit his design. The same attitude pervades here. Despite experience to the contrary, we should hope that a computer code will be simple enough to allow the use of traditional methods. We believe that maximum statistical impact will be made when tools for computer-aided design are widely available so that design is easy in realistic contexts.

One detail raised about computer-aided design concerns the use of centerpoints. Algorithmically, it is very easy to specify centerpoints, or any other points, in a design and then augment those points according to some criterion. Of course, if the right design is used in the first place—that is, a design with better coverage of the input space—then there is no need to fix it up with centerpoints.

At the beginning of his letter, Lucas says that a computer experiment is a simpler, special case of physical experiments. This comment is repeated later. We agree that the lack of random error simplifies analysis. But this opportunity is wasted if we carry over design and analysis methods aimed at minimizing the impact of (nonexistent) error variance. "Special case" implies that a subset of existing methodology for physical experiments will suffice. Computer codes often have very many input variables, however, and often have large nonlinearities. We believe that failing to appreciate the distinctions between physical experiments and computer experiments has led to some misconceptions. We hope that these comments shed some light on the increasingly important area of computer experiments, and we thank Lucas for initiating this discussion.

Finally, we would like to note that Toby Mitchell died before this rejoinder was written. In addition to his many other accomplishments, Toby was a leader in bringing computer experiments to the attention of statisticians. He carried out much of the pioneering work in computer experiments. Having worked closely with him for many years, we believe he would have been in broad agreement with our response.

William J. Welch
Department of Statistics
and Actuarial Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada

Robert J. Buck
252 Kennedy Drive #301
Malden, MA 02148

Jerome Sacks
National Institute
of Statistical Sciences
P.O. Box 14162
Research Triangle Park, NC 27709-4162

Henry P. Wynn
Department of Statistics
University of Warwick
Coventry CV4 7AL
United Kingdom

Max D. Morris
Computer Science
and Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-8083

Matthias Schonlau
Department of Statistics
and Actuarial Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada

ADDITIONAL REFERENCES

Bernardo, M. C., Buck, R., Liu, L., Nazaret, W. A., Sacks, J., and Welch, W. J. (1992), "Integrated Circuit Design Optimization Using a Sequential Strategy," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11, 361–372.

Box, G. E. P., and Draper, N. R. (1959), "A Basis for the Selection of a Response Surface Design," *Journal of the American Statistical Association*, 54, 622–654.

——— (1963), "The Choice of a Second Order Rotatable Design," *Biometrika*, 50, 335–352.

Cook, R. D., and Nachtsheim, C. J. (1990), "Letters to the Editor: Response to James M. Lucas," *Technometrics*, 32, 364–365.

Johnson, M. E., Moore, L. M., and Ylvisaker, D. (1990), "Minimax and Maximin Distance Designs," *Journal of Statistical Planning and Inference*, 26, 131–148.

McKay, M. D., Conover, W. J., and Beckman, R. J. (1979), "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code," *Technometrics*, 21, 239–245.

Morris, M. D., and Mitchell, T. J. (1995), "Exploratory Designs for Computational Experiments," *Journal of Statistical Planning and Inference*, 43, 381–402.

Yu, T. K., Kang, S. M., Sacks, J., and Welch, W. J. (1991), "Parametric Yield Optimization of CMOS Analogue Circuits by Quadratic Statistical Circuit Performance Models," *International Journal of Circuit Theory and Applications*, 19, 579–592.