Experimental Design Matrix of Realizations for Optimal Sensitivity Analysis

Oy Leuangthong (oy@ualberta.ca) and Clayton V. Deutsch (cdeutsch@ualberta.ca)
Department of Civil and Environmental Engineering
University of Alberta

Abstract

Experimental design principles can be used in natural resource management for greater efficiency in sensitivity analysis. Traditionally, practical application of these designs have been limited due to the availability of very specific designs. An approach to determine a design matrix for sensitivity analysis for any case is proposed. The methodology uses an objective function to minimize the difference between first and second order sensitivity terms, that is, the optimized design permits the most reliable inference of first and second order sensitivity terms. Any number of input variables, response variables and case values are permitted. The output design consists of specific settings to run each realization. Some examples are shown and a framework for future research is presented.

Introduction

Uncertainty analysis and sensitivity analysis are closely related concepts. The former quantifies the uncertainty in the output variable that results from uncertainty in the input variables, while the latter quantifies the contribution of each input variable to the total uncertainty of the output variable [3].

In the natural resources industry, geostatistical models have traditionally been constructed to facilitate management decisions in the face of uncertainty. Many books and papers have been written on this subject [4, 6, 9, 15].

Sensitivity analysis, on the other hand, has been largely implemented in practice using a vary-one-at-a-time approach [3]. This essentially involves changing one input variable at a time and comparing the resultant change in the response to the base case. This is a straightforward and useful approach to assess the sensitivity of the response to each input variable.

Unfortunately, most resource studies consist of multiple input variables that can potentially take on a wide range of values. Many different factors can affect the final response or outcome. Applying the vary-one-at-a-time approach is inefficient. Efficiency in time and economics could be obtained if a set of realizations could be pre-determined that permits identification of the most important input variables. In this particular context, an important variable is one that greatly affects the response variable. The set of realizations to be processed is referred to as the "design matrix".

This paper proposes a methodology to solve for the design matrix that will optimize sensitivity analysis of the response variable(s) to the input variables. Based on this de-

sign matrix, the practitioner can then process the set of realizations and determine the sensitivities for each input variable.

Some background information on an increasingly popular area of statistics called experimental design is provided, along with a description of the notation used in this paper. The methodology is presented with some examples to show the preliminary results of this approach. This is followed by a discussion on some future considerations and research directions.

Background

Experimental design describes a growing field in statistics that aims to extract the most information from a set of observations in an efficient manner. In general, two main objectives are addressed via an experiment [2, 11]: (1) to test whether or not two or more input variables have different effects on the response, and (2) to estimate the magnitude of this difference.

The "design" is a set of experiments that reveals the input variable(s) that has the most effect on the response variable. These input variables are also known as predictor variables, or essentially variables that the practitioner can control. The effect of each predictor variable is referred to as the *main effect*. The design may also be set such that the influence of multiple predictors is considered; this influence is referred to as the *interaction* of the predictor variables.

Designs are commonly rated by the quality of information they provide; this is the *resolution* of the experiment. Three common levels of resolution are identified [5]: (1) Resolution III describes those experiments where all main effects can be estimated, (2) Resolution IV are those experiments that estimate all main effects and groups of interactions, and (3) Resolution V experiments estimate all main effects and two-factor interactions.

A complete factorial design permits consideration of all possible variables for all possible values that these variables can take. For a small number of predictor variables, this type of design may be feasible. However, suppose that there are N_i predictor variables, all of which can take N_c possible outcomes. For $N_c = 2$, the space of all combinations is 2^{N_i}; exploration of this space quickly becomes impractical for large N_i. In these cases, consideration of a fractional factorial design is more practical [1, 10] for time and economic constraints.

One such fractional factorial design was proposed by Plackett-Burman (PB) in 1946 [13]. Unlike the approach of independently changing one variable at a time, Plackett-Burman's optimum multifactorial approach changes multiple variables from their nominal values to their extreme values. Assessing the effect of these changes on a certain number of possible combinations can determine the main effect of each predictor variable [12, 13]. This assumes that all interactions are negligible relative to the main effects of the important variables [10, 13]. Estimating only the main effects makes this design a Resolution III experiment [5].

Determination of a Plackett-Burman design is not trivial; it is based on Group Theory, specifically on Galois fields [12, 13, 16], which is beyond the scope of this paper. Designs for the two-factor case ($2^k, k = 1, \ldots, N_i$), that is the case where each variable can take only two possible values, are available for up to 99 realizations, excluding the case for 92 realizations [7]. Only a few designs exist for select cases of other multiple factors.

Terminology in the literature on experimental design is variable. The notation used in this paper is described below with references to the corresponding terminology that would be found in statistical literature. The terminology in this paper is adopted specifically to facilitate communication of the concepts to practitioners.

Notation

- There are N_i input variables, V_i, $i = 1, \ldots, N_i$, each with a distribution $F_{V_i}(v)$, $i = 1, \ldots, N_i$. This is analogous to *factors* in experimental design terminology.
- There are N_r response variables, R_k, $k = 1, \ldots, N_r$, each with an associated function, $r_k = f(V_1, V_2, \ldots, V_{N_i})$.
- The base case value for the input variables is denoted by: V_i^0, $i = 1, \ldots, N_i$.
- The base case values for the response variables are denoted by: R_k^0, $k = 1, \ldots, N_r$.
- Each input variable, V_i, $i = 1, \ldots, N_i$, can take a number of values, N_c. This can be the number of discretizations of the cumulative distribution function (cdf), so a continuous variable can be assigned a discrete number of possible values corresponding to say, the quartiles (so $N_c = 3$).

  Each case is denoted by an integer, $d_i, i = 1, \ldots, N_c$ with 0 assigned to the base case. For example, if the quartiles present two other possible sets of values, then the 0.25 quantile will be assigned an integer of -1, and the 0.75 quantile will be assigned an integer of +1.

  This corresponds to what is referred to as *levels* in experimental design, which are essentially values that a factor can take.
- There are L realizations considered to optimize for sensitivity analysis. Each realization corresponds to a set of values for each input variable, V_i, $i = 1, \ldots, N_i$. For example, for $N_i = 5$, one realization may consist of {-1 0 1 1 -1}. Each realization is referred to as a *test run*.
- Realization values associated to the input and response variables are denoted by a superscript $l, l = 1, \ldots, L$ to represent the realization number. For example, V_i^l, $i = 1, \ldots, N_i$ or R_k^l, $k = 1, \ldots, N_r$.
- The design matrix is denoted by $\mathbf{D}$, which is an $L \times N_i$ matrix consisting of integers, $d_i, i = 1, \ldots, N_c$, that represent the different N_c cases each input variable can take. This is often referred to as either a *design* or a *layout*; these two terms are used interchangeably in statistical literature.

$$\mathbf{D} = \begin{bmatrix} d_1^1 & \cdots & d_{N_i}^1 \\ \vdots & \ddots & \vdots \\ d_1^L & \cdots & d_{N_i}^L \end{bmatrix}$$

Methodology

The problem is to calculate a design matrix, **D**, that permits optimal calculation of sensitivity terms with a fixed number of test runs or realizations.

The idea is to develop a general solution that does *not* require that the response function be known in advance - we only require the following information: the number of input and response variables, N_i and N_r; the distribution of the input variables, $f_{V_i}(v)$; the number of possible values that these variables can take, N_c; and the number of realizations, L, that the user would like to process. Specific knowledge of the response function, and hence distribution, should improve the solution.

The number of possible combinations posed by this problem is huge:

$$\left(\begin{array}{c} N_c^{N_i} \\ L \end{array} \right) = \frac{N_c^{N_i}!}{(N_c^{N_i} - L)!L!}$$

For example, for 4 input variables, 3 possible values (including the base case), 1 response variable, and 5 realizations, there are 25.6×10^6 possible sets of 5 realizations that can be chosen.

The challenge of choosing the best set of L realizations over the combinatorial is daunting. The choice of the "best" set will be based on optimizing an objective function. This function is devised such that the key parameters to be optimized are closeness to (1) the first order sensitivity of the response function(s) to the input variables, $\frac{\partial r_k}{\partial v_i}, i = 1, \ldots N_i$, and (2) the second order sensitivity of the response function(s), $\frac{\partial^2 r_k}{\partial v_i \partial v_j}, i, j = 1, \ldots N_i$. Note that the first order sensitivity term is the partial derivative of the response function taken with respect to the i^{th} input variable; similarly, the second order sensitivity term is the second order partial derivative of the response function taken with respect to the i^{th} and j^{th} input variables. The objective function that will be minimized is:

$$O = w_1 \cdot \left\| \frac{\partial r_k}{\partial v_i}^* - \frac{\partial r_k}{\partial v_i} \right\| + w_2 \cdot \left\| \frac{\partial^2 r_k}{\partial v_i \partial v_j}^* - \frac{\partial^2 r_k}{\partial v_i \partial v_j} \right\| \quad (1)$$

where

$$\begin{array}{rcl} \frac{\partial r_k}{\partial v_i} & = & \text{first order sensitivity taken with respect to input variable } i, i = 1, \ldots N_i \\ \frac{\partial^2 r_k}{\partial v_i \partial v_j} & = & \text{second order sensitivity with respect to input variables } i, j, i = 1, \ldots N_i \\ w_\alpha & = & \text{parameter for optimization, } \alpha = 1, \ldots, 2 \\ \|\cdot\| & = & \text{the norm function} \end{array}$$

and the superscript * denotes an estimate of the unknown true value. The term $\frac{\partial r_k}{\partial v_i}$ provides information on the rate of change of the k^{th} response variable, R_k, with respect to the i^{th} input variable, V_i. The second sensitivity term, $\frac{\partial^2 r_k}{\partial v_i \partial v_j}$, gives information about the shape of the surface of the response function; it can also be interpreted as how fast or slow the slope or gradient is changing.

The solution to such an optimization problem is not trivial. The response function is unknown, so the real or true first and second order sensitivity coefficients, $\frac{\partial r_k}{\partial v_i}$ and $\frac{\partial^2 r_k}{\partial v_i \partial v_j}$, are unknown. Note, however, that if the sensitivity terms are known, then the response variable can be approximated by a Taylor series expansion expressed up to the second order terms:

$$r_k^l = r_k^0 + \sum_{i=1}^{N_i} \frac{\partial r_k}{\partial v_i} \cdot \Delta V_i^l + \frac{1}{2} \sum_{i=1}^{N_i} \sum_{j=1}^{N_i} \frac{\partial^2 r_k}{\partial v_i \partial v_j} \cdot \Delta V_i^l \cdot \Delta V_j^l \quad , l = 1, \ldots, L \tag{2}$$

where $\Delta V_i^l = V_i^l - V^0, i = 1, \ldots, N_i$. The response can be calculated for each realization of the set of input variables.

For any response variable, notice that:

- If the response function is continuous, then

$$\frac{\partial^2 r_k}{\partial v_i \partial v_j} = \frac{\partial^2 r_k}{\partial v_j \partial v_i}$$

  This means that the number of different sensitivity terms is really $N_i + N_i(N_i + 1)/2$ and *not* $N_i + N_i^2$.

- The set of first order partial derivatives is commonly referred to as the gradient of the response function r_k, and is commonly denoted by $\nabla r_k(v_1, \ldots, v_{N_i}) = \left[\frac{\partial r_k}{\partial v_i}\right], i = 1, \ldots, N_i$.

- The set of second order partial derivatives is commonly referred to as the Hessian matrix of the response function r_k of size $N_i \times N_i$:

$$\mathbf{H}_{r_k}(v_1, \ldots, v_{N_i}) = \nabla^2 r_k(v_1, \ldots, v_{N_i}) = \left[\frac{\partial^2 r_k}{\partial v_i \partial v_j}\right], i, j = 1, \ldots, N_i$$

Since the true sensitivity terms are unknown, one way to solve this problem in the general case is to treat these sensitivity terms as random variables (RVs). The design matrix, $\mathbf{D}$, can then be determined for a set of sensitivity terms that are considered to be *possible truths*. To do this, $\frac{\partial r_k}{\partial v_i}$ and $\frac{\partial^2 r_k}{\partial v_i \partial v_j}$ are randomly drawn from an arbitrarily chosen standard normal distribution. Note that even for a simple response function, there are no constraints on the value that these two sensitivity terms can take, that is if $\frac{\partial r_k}{\partial v_i} > 0$, $\frac{\partial^2 r_k}{\partial v_i \partial v_j}$ is not bounded by some minimum or maximum. In fact, if $\frac{\partial^2 r_k}{\partial v_i \partial v_j} < 0$, then the slope of the response surface is decreasing and the surface is levelling or flattening off. If the signs are the same, that is $\frac{\partial^2 r_k}{\partial v_i \partial v_j} > 0$), then the slope of the response surface is increasing and continues to become more steep. To allow for complex response surfaces, these two terms are drawn independently.

Once the "true" value for each sensitivity term is drawn and an initial design matrix is chosen, the corresponding "true" response value can be calculated using Equation 2.

Given that the true response is known and a design matrix has been proposed, we can solve for the set of sensitivity terms that will yield the response values. This now assumes that the sensitivity terms are no longer available and must now be determined. This is a straightforward matrix computation *if* the number of equations, L, is equal to the number of unknowns, $N_i + N_i(N_i + 1)/2$; unfortunately, this is usually not the case. The more common scenario is that there are more variables of interest and fewer realizations, that is, $N_i + N_i(N_i + 1)/2 > L$ resulting in what is commonly referred to as the under-determined system [14]. There is, of course, the other possibility of an over-determined system in which there are more equations than there are unknowns. For both these types of systems, the singular value decomposition (SVD) algorithm [14, 8] is robust in solving for a solution of the system. Note that although the solution is non-unique, the algorithm can still solve for a set of different solutions [14]; however, this was not implemented in the methodology presented here since this will seriously impact the computational time required to solve the system, which is in itself only one component of the main methodology.

The overall methodology can be summarized by the following steps:

1. Draw a large number of values from the RVs for the $N_i + N_i \cdot (N_i + 1)/2$ first and second order sensitivity terms, $\frac{\partial r_k}{\partial v_i}$ and $\frac{\partial^2 r_k}{\partial v_i \partial v_j}$, respectively.

2. Draw an initial design matrix ($\mathbf{D}$) by Monte Carlo simulation (MCS) of the N_c cases for each input variable.

$$\mathbf{D} = \left[\begin{array}{ccc|ccc} \Delta V_1^1 & \cdots & \Delta V_{N_i}^1 & \Delta V_1^1 \Delta V_1^1 & \cdots & \Delta V_1^1 \Delta V_{N_i}^1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \Delta V_1^L & \cdots & \Delta V_{N_i}^L & \Delta V_{N_i}^1 \Delta V_1^1 & \cdots & \Delta V_{N_i}^1 \Delta V_{N_i}^1 \end{array}\right]$$

3. Calculate the objective function, O, in Equation 1:

   (a) Perform SVD on the design matrix.

   (b) For each set of values for the sensitivity terms:

      i. Calculate the response associated to the set of $N_i + N_i \cdot (N_i + 1)/2$ terms for the L realizations at the base case values.

      ii. Solve for estimates of the first and second order sensitivities, $\frac{\partial r}{\partial v_i}$ and $\frac{\partial^2 r}{\partial v_i \partial v_j}$, by back substitution of the SVD matrix with its associated vector of response values. The estimates will not be equal to the truth since the solution will likely not be unique (for the case of $L \neq N_i + N_i \cdot (N_i + 1)/2$).

      iii. Calculate the difference between the estimate and the true values for the sensitivity terms, and calculate the objective function (Equation 1).

      iv. Repeat until all sets of RVs have been solved.

4. Perturb this design matrix, $\mathbf{D'}$, by randomly choosing a realization and a variable to change. Recalculate the objective function, O' (See Step 3).

5. If $O' < O$, then set $\mathbf{D'}$=$\mathbf{D}$. Repeat Step 4, until the number of perturbations (set by the user) is reached.

Once the maximum number of perturbations is reached, the $L \times N_i$ design matrix is output. The first row of the design matrix is reserved for the base case scenario; this row consists of zeroes to denote the base case value for each input variable. The subsequent $L - 1$ rows are written out in an order that is sorted on a by-column basis in ascending order, that is, column 1 is sorted in ascending order, "ties" are broken by sorting column 2, and so on.

Implementation

The proposed methodology is implemented in a prototype program called `dmatrix`. Details on the required parameters to execute this program are given in the Appendix.

The current implementation of the algorithm considers equal minimization of the difference in both the first and second order sensitivity terms, that is, the weights in Equation 1 are equal. As well, the existing algorithm is implemented for only one response variable; future implementation of this algorithm will allow for multiple response variables.

The following examples show the application of the algorithm to a couple of different cases: (1) number of input variables is low, but the number of cases is high, (2) number of input variables is high and number of cases is low. The specific number of variables and cases are arbitrarily chosen to show the flexibility in user specification.

A comparison case is also provided that compares the objective function value resulting from the Plackett-Burmann design and the design matrix from `dmatrix` for the case of nine input variables, three possible outcomes, and nine assemblies or realizations ($N_i = 9, N_c = 3, L = 9$). In all examples, the distribution for each input variable is arbitrarily chosen to be standard normal, and the base case values are set at the median.

Example 1 - $N_i = 3, N_c = 9, L = 10$

Ten different random seed numbers are chosen to initialize the design matrix in each of ten different runs. This allows us to assess convergence of the objective function given ten different initial design matrices. The design matrix that gave the lowest objective function is shown in Table 1, while the convergence of the objective function is given in Figure 1. In any one run, the objective function value decreases quickly in the first 500 perturbations; after 1000 perturbations, the objective function values remain fairly steady. The convergence of the objective function value to just under 10 000 is quite good given that for all runs, the initial objective function value started at just under 1×10^6.

Example 2 - $N_i = 15, N_c = 3, L = 10$

Ten different random seed numbers are also used in this case for the same purpose of testing convergence of the objective function. Table 2 shows the design matrix that gave the lowest objective function, and Figure 2 shows how quickly this objective function converged. Again, we see that convergence is fairly quick and occurs at only 200 perturbations.

An interesting result in Table 2 shows that input variables 8 and 13(corresponding to columns 8 and 13) are assigned the same values for all 10 realizations. This essentially prevents determination of the sensitivity of the response variable to these two variables

0	0	0
-4	-4	1
-4	4	-4
-4	4	2
-3	-4	3
-3	3	4
-2	-3	4
1	4	4
2	-4	4
4	4	-4

Table 1: Design matrix from `dmatrix` that gave the lowest objective function values for $N_i = 3$, $N_c = 9$ and $L = 10$. Note that the first row denotes the base case scenario. Each row represents a realization while each column corresponds to each input variable.

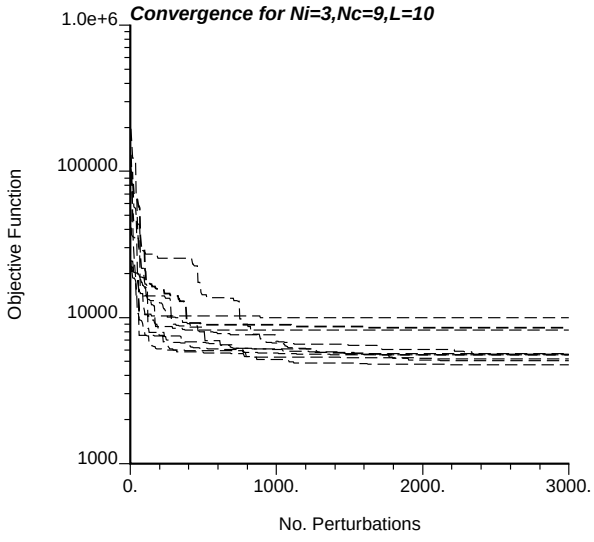

Figure 1: Convergence of objective function value from ten different initial matrices for $N_i = 3$, $N_c = 9$ and $L = 10$.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-1	-1	0	0	0	1	0	0	-1	-1	0	-1	-1	-1
0	-1	-1	0	0	-1	0	0	1	-1	-1	-1	-1	-1	-1
0	-1	-1	1	-1	0	-1	0	0	-1	-1	-1	-1	-1	-1
0	-1	1	-1	0	0	-1	0	-1	-1	0	1	-1	0	-1
0	-1	1	-1	1	0	-1	0	-1	-1	0	-1	-1	-1	1
0	1	-1	-1	-1	0	-1	0	-1	-1	-1	-1	-1	-1	0
0	1	1	-1	-1	1	0	0	0	0	1	1	-1	0	0
0	1	1	-1	0	-1	1	0	0	-1	-1	1	-1	-1	0
1	-1	1	-1	-1	1	-1	0	-1	-1	0	1	-1	1	-1

Table 2: Design matrix from `dmatrix` that gave the lowest objective function values for $N_i = 15$, $N_c = 3$ and $L = 10$. Note that the first row denotes the base case scenario. Each row represents a realization while each column corresponds to each input variable.

from being considered, and suggests that a penalty function should be considered to avoid this type of scenario.

Example 3 - Comparison to PB Design

The idea for this third example is to compare a known Plackett-Burman (PB) design to that obtained from the proposed methodology. The two designs are compared based on the objective function value, and also on the design matrix.

The choice of the PB design for comparison is arbitrary. A PB design exists for the case of $N_i = 9, N_c = 3, L = 9$ (see Table 3). Notice that the design is cyclic, this is characteristic of a Hadamard matrix in which the first column or values (minus the last row which is reserved for the base case) is required. The rest of the design is obtained by cyclically moving the row of values along the $N_i - 1$ columns [5].

For easier comparison, the PB design in Table 3 is sorted in the same manner as the output designs from `dmatrix`, that is, by successively sorting each column in ascending order. This is shown in Table 4. Processing this matrix through the algorithm and calculating the objective function in Equation 1 yields a function value of 54949.660 for random seed number 455411, and 54740.285 for seed number 69069.

On the other hand, the program `dmatrix` was executed for the same case of $N_i = 9, N_c = 3, L = 9$ for 2000 perturbations. For the same seed numbers, the objective function value is 54918.80 for 455411 (Table 5), and 54742.29 for seed 69069 (Table 6).

The objective function values from both the PB and the `dmatrix` designs are close in magnitude, but the designs from `dmatrix` vary depending on the seed number chosen. For this particular scenario, the algorithm took 40m39s to run once on a P4, 2.2GHz computer for 2000 perturbations. Increasing the number of perturbations (to several tens of thousands of trials rather than just a couple thousand) may lead to convergence of the design matrix to the PB design. The other possibility is that a true simulated annealing approach may be required to prevent against converging to a design that represents a local minima in the

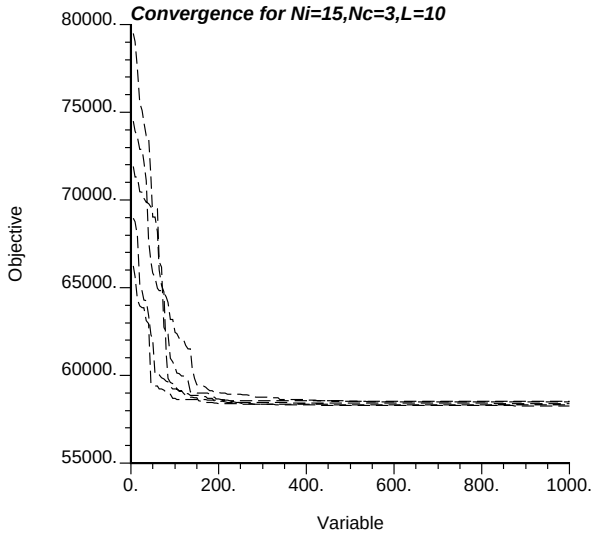

Figure 2: Convergence of objective function value from ten different initial matrices for $N_i = 15$, $N_c = 3$ and $L = 10$.

objective function.

Discussion

Research in this area is still preliminary. There are several issues related to this work that remains to be addressed.

Assumptions. Once a design matrix is available, the "truth" values of the sensitivity terms, $\frac{\partial r_k}{\partial v_i}$ and $\frac{\partial^2 r_k}{\partial v_i \partial v_j}$, are used along with the design matrix to calculate the corresponding "true" response according to Taylor series expansion in Equation 2. All ΔV_i^l are calculated relative to the base case. For the first order terms in Equation 2, this amounts to assuming a linearization of the response function at the base case input variable values.

Note that in this particular implementation, all calculations are performed using numerically derived values. If the response function was second-order derivable (as in the case of a third or higher order polynomial), then implementation of the algorithm would still proceed by calculating the effect of the sensitivity terms at the base case. Consequently, the assumed linearization effect would still be present.

CPU Time. Implementation of the algorithm requires making some decisions that affect the time to actually execute the program. Firstly, a choice must be made on the number of random numbers to draw for each of the first and second order sensitivity terms in the first step of the proposed methodology. In general, drawing 10 000 random values for each sensitivity term would amount to drawing a total of $(N_i \cdot (N_i + 1)/2) \cdot 10000$ values. For $N_i = 5$, this is 150 000 values, for $N_i = 10$, this is 550 000 values, etc. This step of actually

0	-1	-1	1	0	1	1	-1	0
-1	0	-1	-1	1	0	1	1	-1
1	-1	0	-1	-1	1	0	1	1
1	1	-1	0	-1	-1	1	0	1
0	1	1	-1	0	-1	-1	1	0
1	0	1	1	-1	0	-1	-1	1
-1	1	0	1	1	-1	0	-1	-1
-1	-1	1	0	1	1	-1	0	-1
0	0	0	0	0	0	0	0	0

Table 3: Plackett-Burman design for $N_i = 9$, $N_c = 3$ and $L = 9$. In this design, the bottom row denotes the base case scenario. Note that in most experimental design literature, this design consists of 0, 1, and 2 with 0 denoting the base case. This was recoded as -1 and 1 for PB codes 1 and 2, respectively; 0 still denotes the base case.

-1	-1	1	0	1	1	-1	0	-1
-1	0	-1	-1	1	0	1	1	-1
-1	1	0	1	1	-1	0	-1	-1
0	-1	-1	1	0	1	1	-1	0
0	1	1	-1	0	-1	-1	1	0
1	-1	0	-1	-1	1	0	1	1
1	0	1	1	-1	0	-1	-1	1
1	1	-1	0	-1	-1	1	0	1
0	0	0	0	0	0	0	0	0

Table 4: Plackett-Burman design from Figure 3, but sorted by successive columns in ascending order to facilitate comparison with `dmatrix` results.

-1	-1	-1	-1	0	-1	-1	-1	0
-1	-1	0	-1	-1	1	0	0	-1
-1	0	1	-1	0	-1	0	-1	-1
0	0	-1	0	0	-1	0	0	1
0	1	0	-1	1	1	1	-1	-1
0	1	0	0	-1	0	0	0	0
1	0	1	-1	0	0	1	0	-1
1	1	-1	-1	-1	-1	-1	-1	-1
0	0	0	0	0	0	0	0	0

Table 5: Design matrix from `dmatrix` with random seed number 455411 for $N_i = 9$, $N_c = 3$ and $L = 9$. Objective function value for this matrix is 54918.8 compared to 54949.7 using PB design in Table 4.

-1	-1	-1	0	0	0	-1	-1	0
-1	0	0	0	0	-1	0	1	-1
-1	0	1	1	-1	-1	-1	-1	1
0	-1	1	0	0	-1	-1	-1	1
0	0	-1	0	1	0	0	1	-1
0	0	1	0	0	0	-1	0	-1
1	-1	-1	-1	0	0	-1	-1	1
1	-1	-1	0	-1	0	1	0	-1
0	0	0	0	0	0	0	0	0

Table 6: Design matrix from `dmatrix` with random seed number 69069 for $N_i = 9$, $N_c = 3$ and $L = 9$. Objective function value for this matrix is 54742.3 compared to 54740.3 using PB design in Table 4.

drawing the 150 to 550 thousand random values is not really a CPU expensive task, the additional time to draw more is only a fraction of a second.

Instead, the effect of drawing 10 000 random values is evident later in the methodology when the objective function must be calculated. Essentially, each perturbation of the design matrix involves (1) solving the SVD system once, and (2) back substitution of the SVD solution 10 000 times. Thus, for 1000 perturbations, back substitution of the system will occur 10×10^6 or 10 million times. For a relatively small system, this in itself is also not that CPU expensive. However, when the size of the design is fairly large, the entire algorithm can take some time to run. For example, a system of 15 input variables, 3 possible outcomes including the base case, and 10 realizations ($N_i = 15, N_c = 3, L = 10$, as in Example 2), the algorithm takes 93m23.4s on a P4, 2.2GHz computer. Alternatively, if we set the number of random values to draw at 1 000, this same system takes only 9m32.6s to run on the same computer.

Differences with Plackett-Burman Design. The current algorithm does not pose a constraint on the number of times a particular case value (or *level*) can be chosen for each input variable. For example, for 10 realizations ($L = 10$) the first input variable can potentially take the same value in all but the first case (recall that this corresponds to the base case realization). Of course, the potential for this to occur is low, but there are no explicit controls implemented to prevent this from happening.

Most other experimental designs for fractional factorial experiments impose a constraint that for each input variable, each case appears in the design an equal number of times. For nine realizations, $L = 9$, each case will appear $(L)/N_c$ times. If the number of possible values is 3, that is $N_c = 3$, then each of the three cases can occur 3 times in the design. This type of design satisfies a property of orthogonality [1].

Convergence of the design matrix from the proposed approach is expected to satisfy this property of orthogonality if a large number of perturbations were permitted. Note that unlike simulated annealing where the number of perturbations may exceed 80 000, this algorithm implements only a fraction of this number for this to be computationally efficient. This warrants further investigation.

Calculating main effects. The main effect of each input variable is the average effect of that variable on the response value taken over the various values of the other input variables [17]. It can be estimated by considering only the effect of the input on the response [10, 17]:

$$M(V_i) = \sum_{l=1}^{L} d_i \cdot r^l, \qquad \forall i = 1, \ldots, N_i$$

where $M(\cdot)$ is the main effect of variable $(\cdot)$. Prior to calculating the main effects, the realizations specified in the design matrix must be processed to obtain the response value for each realization, $r^l, l = 1, \ldots, L$.

Future Work

Implementation of this type of experimental design approach is efficient from the perspective of both professional and computational time. Further, this methodology is flexible in accounting for different combinations of input variables, cases and response variables. This gain in efficiency and flexibility from traditional experimental design for sensitivity analysis merits further work.

Based on initial runs of the algorithm, there are a number of considerations that must be addressed in future implementations. These may include: (1) using simulated annealing to optimize the objective function and to avoid convergence on local minimas of the response surface, (2) imposing a penalty function for the absence of any case or level of an input variable, and (3) ensuring that each sensitivity coefficient can be determined equally well. The orthogonal property of conventional factorial designs may be closely related to the latter two issues, and thus warrant closer examination.

In addition to addressing the above issues, future work in this area will extend this research to allow for additional realizations to be computed given an already optimized design matrix. This essentially will build on the documented approach by allowing for additional test cases to be executed given that the practitioner has already run the initially specified L realizations. There are several reasons to consider directing research in this direction: more time may be available in the project schedule to allow for more extensive sensitivity analysis than originally budgeted, or there may be a few variables whose main effects may be sufficiently close to warrant further study.

References

[1] G. Box, W. Hunter, and J. Hunter. *Statistics for Experimenters.* John Wiley & Sons Inc., New York, 1978.

[2] W. Cochran and G. Cox. *Experimental Designs.* John Wiley & Sons Inc., New York, second edition, 1960.

[3] C. Deutsch, M. Monteiro, S. Zanon, and O. Leuangthong. Procedures and guidelines for assessing and reporting uncertainty in geostatistical reservoir modeling. Technical report, Centre for Computational Geostatistics, University of Alberta, Edmonton, AB, March 2002.

[4] C. V. Deutsch. *Geostatistical Reservoir Modeling.* Oxford University Press, New York, 2002.

[5] W. Diamond. *Practical Experiment Designs for Engineers and Scientists.* Van Nostrand Reinhold, New York, 1989.

[6] P. Goovaerts. *Geostatistics for Natural Resources Evaluation.* Oxford University Press, New York, 1997.

[7] J. Johnson. Plackett-burman designs using galois fields. *Annals of Eugenics?*, pages 1–5, 2001.

[8] R. Johnson and D. Wichern. *Applied Multivariate Statistical Analysis.* Prentice Hall, New Jersey, 1998.

[9] A. G. Journel and C. J. Huijbregts. *Mining Geostatistics.* Academic Press, New York, 1978.

[10] R. Mason, R. Gunst, and J. Hess. *Statistical Design and Analysis of Experiments with Applications to Engineering and Science.* John Wiley & Sons Inc., New York, 1989.

[11] R. Petersen. *Design and Analysis of Experiments.* Marcel Dekker, Inc., New York, 1985.

[12] R. Plackett. Some generalizations in the multifactorial design. *Biometrika*, 33(4):328–332, 1946.

[13] R. Plackett and J. Burman. The design of optimum multifactorial experiments. *Biometrika*, 33(4):305–325, 1946.

[14] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes.* Cambridge University Press, New York, 1986.

[15] A. Sinclair and G. Blackwell. *Applied Mineral Inventory Estimation.* Cambridge University Press, United Kingdom, 2002.

[16] W. Stevens. The completely orthogonalized latin square. *Annals of Eugenics*, pages 82–93, 1939.

[17] G. Taguchi. *System of Experimental Designm, Volume 1.* UNIPUB/Kraus International Publications, New York, 1987.

Appendix

An example parameter file for `dmatrix` is shown in Figure 3 and are explained below:

- **niv**: number of input or predictor variables.
- **biv(i), i=1,...,niv**: base case values for each predictor variable.
- The next three lines are repeated niv times, once for each predictor variable:
  - **transfl(i)**: file with input data for determination of data distribution.
  - **icol(i), iwt(i)**: column number for variable i, and corresponding weights.
  - **tmin(i), tmax(i)**: trimming limits to filter out variable i.
- **nrv**: number of response variables.
- **brv(i),i=1,..., nrv**: base case values for each response variable.
- **nreal**: number of desired realizations for processing.

- **ixv(1)**: random number seed.
- **MAXPERT**: number of perturbations to run.
- **outfl**: file for output. This file contains the optimum design matrix.
- **sumfl**: file with summary information about the convergence of the objective function.

```
                Parameters for DMATRIX
                **********************

START OF PARAMETERS:
2                              - number of input variables
0.25  3.01                     - base case values for input variables
3                              - number of outcome cases (excluding base case)
datafile1.out                  - file with input variable 1
5     3                        - column for variable 1 and weight
-1.0e21    1.0e21              - trimming limits for variable 1
datafile1.out                  - file with input variable 2
5     3                        - column for variable 2 and weight
-1.0e21    1.0e21              - trimming limits for variable 2
1                              - number of response variables
5.0                            - base case values for response variables
5                              - number of realizations
69069                          - random number seed
1000                           - number of perturbations
dmatrix.out                    - output file for design matrix
dmatrix.sum                    - summary file to report objective functions
```

Figure 3: Parameters for `dmatrix`.