Advances in Experimental Design Matrix for Establishing Sensitivity

Oy Leuangthong

Centre for Computational Geostatistics
Department of Civil and Environmental Engineering,
University of Alberta

Abstract

The use of experimental design for optimal sensitivity analysis was proposed in the 2003 CCG report. A methodology was developed to determine a design matrix of realizations that should be processed through to a transfer function such as flow simulation or engineering design. This presents an efficient means of performing sensitivity analysis with a minimum of realizations. The methodology consisted of minimization of an objective function that characterizes the difference between reference and observed first and second order sensitivity terms. Flexibility in the algorithm permitted consideration of any number of input variables, response variables and case values.

This paper presents the latest advances in determining this experimental design matrix for optimal sensitivity analysis. Considerations in the recent work include (1) maximization of entropy as an objective, (2) implementation of simulated annealing for optimization, and (3) extension of the current algorithm to incrementally optimize the design matrix for additional realizations. Application of this methodology to a synthetic petroleum reservoir is illustrated.

Introduction

Uncertainty and sensitivity analysis are closely related; the former considers uncertainty in the response variable as a result of uncertainty in the input variables, while the latter quantifies the contribution of each variable to the total uncertainty of the response variable. The use of Monte Carlo simulation (MCS) is ubiquitous in uncertainty assessment; however, the efficiency of MCS can be improved by applying a stratified sampling approach, such as Latin Hypercube Sampling (LHS). These methods are simple, straightforward and commonly used for uncertainty assessment.

Sensitivity analysis has been largely implemented in practice using a vary-one-at-a-time approach[2]. This involves changing one input variable at a time and comparing the resultant change in the response to the base case. Although, this approach is straightforward to assess the sensitivity of the response to each input variable, it presents inefficiencies in both time and economics.

In the case of multiple input variables that affect the response outcomes, the vary-one-at-a-time approach is particularly inefficient. The idea proposed was to determine a "design matrix", consisting of the set of realizations that should be processed for sensitivity analysis. In this context, sensitivity results will indicate the input variable(s) that greatly affects the response variable. The inspiration for this idea was Plackett-Burman's optimal

multifactorial design [8] which was theoretically derived for only a limited set of scenarios. Rather than follow the theoretical approach, the methodology proposed in 2003 was to *numerically* derive a design matrix that permitted greater flexibility in the number of input variables, the possible cases these variables could take, the number of response variables and the number of realizations that the practitioner wanted to process. While a design matrix methodology was implemented in the 2003 annual CCG report, a number of issues were identified for further evaluation. Orthogonality of the matrix was one such issue. **Need definition of orthogonal design**. The implication of a non-orthogonal design is that certain variables would be over-represented with too many cases, while others would be under-represented with little to no change in the case values that it could take. Thus, construction of a more orthogonal design should lead to equal representation of all cases over all variables. This should allow for a fairer assessment of sensitivity.

Another issue was the chosen optimization method. In any complex, non-linear setting, there usually the lingering question of whether the algorithm converges to a local optimal objective rather than the global optimal value. The methodology proposed in 2003 consisted of a rather simple rejection algorithm. Simulated annealing [3] was proposed as a means to address this issue; this would take more time, yet the complexity of the problem and the large solution space lends itself to this more sophisticated optimization algorithm.

Another suggestion was to implement this algorithm sequentially, that is, allow for more realizations to be processed after an initial batch. Suppose the original design matrix had been processed and additional computational time and resources were available to enhance the sensitivity analysis by processing another set of realizations. We decide on the next set of realizations to be processed. This issue of incremental optimization is fairly straightforward, however optimization in this context should also account for the sensitivity results corresponding to the original design matrix.

This paper reviews the background presented in 2003. This is followed by the above considerations and results in development of this latest design matrix methodology. Implementation considerations are discussed and an application to a synthetic petroleum reservoir is presented.

Background

Experimental design describes a growing field in statistics that aims to extract the most information from a set of realizations. The "design" is a set of experiments that reveals how the input variables affect the response variable. These input variables are also known as predictor variables. 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.

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. For most problems, however, a fractional factorial design is more practical [1, 6] for time and economic constraints. One such fractional factorial design was proposed by Plackett-Burman (PB) in 1946 [8]. 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 ([7, 8]. This assumes that all interactions are negligible rel-

ative to the main effects of the important variables [6, 8].

Determination of a Plackett-Burman design is not trivial; it is based on Group Theory, specifically on Galois fields [7, 8, 9], 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 [4]. Only a few designs exist for select cases of other multiple factors.

A recall of the notation adopted in this paper is described below with references to the corresponding terminology that would be found in statistical literature.

Recall of Notation

- There are N_i input variables, V_j, $j = 1, \ldots, N_i$, each with a distribution $F_{V_j}(v)$, $j = 1, \ldots, N_i$. 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_j^0, $j = 1, \ldots, N_i$.
- The base case values for the response variables are denoted by: R_k^0, $k = 1, \ldots, N_r$.
- Each input variable, V_j, $j = 1, \ldots, N_i$, 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_j, $j = 1, \ldots, N_i$. For example, for $N_i = 5$, one realization may consist of {-1 0 1 1 -1}. Each realization is also 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_j^l, $j = 1, \ldots, N_i$ 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.

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

The Objective

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, respectively; the distribution of the input variables, $f_{V_i}(v), i = 1, \ldots, N_i$; 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. Note that in the case of continuous variables, N_c can represent the number of discretizations of the cumulative distribution function (cdf). Specific knowledge of the response function, and hence distribution, should improve the solution; however, this is usually unavailable at the start of a project.

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 more than 25 million possible sets of 5 realizations that can be chosen.

The challenge of choosing the best set of L realizations over the combinatorial is daunting. Simulated annealing presents an ideal optimization algorithm for large, complex and non-linear problems [3]. This algorithm requires the explicit definition of an objective function, and employs an acceptance and conditional-rejection scheme to random perturbations of an initial system. Convergence depends on the solution space and the annealing schedule that determines the probability of acceptance/rejection in the case of a less favourable perturbation. This possibility of allowing less favourable changes to be accepted permits the chance to reach global optimums rather than accepting local optimums, which can often be the case with more "greedy" optimization algorithms like the steepest descent methods.

The focus is to assess sensitivity and since derivative terms of the response variable are commonly used measures of sensitivity, it is natural to expect that these terms be considered as part of the objective. The first order sensitivity of a response function to the input variables, $\frac{\partial r_k}{\partial v_i}, i = 1, \ldots N_i$, 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.

To account for the shape of the response surface, one may also wish to consider the second order sensitivity of the response function(s); $\frac{\partial^2 r_k}{\partial v_i \partial v_j}, i, j = 1, \ldots N_i$. This gives information about the shape of the surface of the response function; it can also be interpreted as how fast the slope or gradient is changing. A positive value of $\frac{\partial^2 r_k}{\partial v_i \partial v_j}, i, j = 1, \ldots N_i$ indicates that the response is a minimum (or one can imagine that the response surface at this point lies within a valley); that is, a change in the i^{th} and j^{th} input variables , V_i and V_j, will yield a response value that is larger than the current value. A negative value then indicates a surface that will decrease with a change in the input variables. While this is useful to determine whether we are at a maximum or minimum response value, it does ensure that this maximum/minimum is a global maximum/minimum; the employment of simulated annealing offsets this possibility.

One other consideration is that of an orthogonality of the design, that is, all cases (or levels) of each input variable (or factor) appears in the same number of realizations [11]. This property can be interpreted as achieving equal representation for all cases for

all variables. Table 1 shows an example of an orthogonal design for 7 input variables, 8 realizations and 2 cases for each variable (i.e. $N_i = 7, L = 8$ and $N_c = 2$) [8], where -1 represents the base value and +1 represents the extreme value. Note that this orthogonal design only requires knowing the first row (1,1,1,-1,1,-1,-1); the next realization is obtained by shifting the case value of variable 1 to variable 2, variable 2 to variable 3, and so on from the preceding realization. This shifting of case values for the subsequent realizations is referred as cycling across the realizations.

Realizations	**Input Variables**						
	1	2	3	4	5	6	7
1	1	1	1	-1	1	-1	-1
2	-1	1	1	1	-1	1	-1
3	-1	-1	1	1	1	-1	1
4	1	-1	-1	1	1	1	-1
5	-1	1	-1	-1	1	1	1
6	1	-1	1	-1	-1	1	1
7	1	1	-1	1	-1	-1	1
8	-1	-1	-1	-1	-1	-1	-1

Table 1: Example of an orthogonal design for 8 realizations and 7 input variables that can take 2 case values [8].

While this orthogonality property is quite important and common in a design, Lin and Chang note that it is not always possible to achieve orthogonality[5]. We can, however, add this property as an objective to strive for in the optimization. This can be achieved in two ways: (1) choosing an initial realization and then cycling this set of cases to the next realization, and (2) explicitly add this into the objective function by maximizing the entropy of the design matrix. Cycling across realizations is quite straightforward, however, this only works to achieve orthogonality for a specific number of realizations [11, 5]. For the general case, we can calculate the entropy of the system and use this to penalize the objective. A low entropy system corresponds to a design that deviates from an orthogonal design while an orthogonal design achieves the highest entropy system, thus the objective will be to maximize entropy. Entropy can be calculated as:

$$H = -\sum_{i=1}^{N_i}\sum_{j=1}^{N_c} ln[F_i(d_j)]F_i(d_j) \tag{1}$$

where $F_i(d_j)$ represents the probability of case j for the i^{th} variable, V_i determined over all the realizations within the design matrix. For instance, the entropy for the design in Table 1 is 4.852. This can be compared to the vary-one-at-a-time approach as represented by the design in Table 2, which has an entropy of 2.637.

Overall, the choice of the "best" set will be based on optimizing an objective function that considers all of the above objectives:

$$O = w_1 \cdot \left\| \frac{\partial r_k}{\partial v_i}^* - \frac{\partial r_k}{\partial v_i} \right\| + w_2 \cdot \left\| \frac{\partial^2 r_k}{\partial v_i \partial v_j}^* - \frac{\partial^2 r_k}{\partial v_i \partial v_j} \right\| + w_3 \cdot [H_{max} - H^*] \tag{2}$$

Realizations	Input Variables						
	1	2	3	4	5	6	7
1	1	-1	-1	-1	-1	-1	-1
2	-1	1	-1	-1	-1	-1	-1
3	-1	-1	1	-1	-1	-1	-1
4	-1	-1	-1	1	-1	-1	-1
5	-1	-1	-1	-1	1	-1	-1
6	-1	-1	-1	-1	-1	1	-1
7	-1	-1	-1	-1	-1	-1	1
8	-1	-1	-1	-1	-1	-1	-1

Table 2: Design matrix corresponding to the vary-one-at-a-time approach for 8 realizations and 7 input variables that can take 2 case values.

where

$\frac{\partial r_k}{\partial v_i}$ = first order sensitivity taken with respect to input variable $i, i = 1, \ldots N_i$
$\frac{\partial^2 r_k}{\partial v_i \partial v_j}$ = second order sensitivity with respect to input variables $i, j, i = 1, \ldots N_i$
w_α = parameter for optimization, $\alpha = 1, \ldots, 2$
$\|\cdot\|$ = the norm function
H = entropy of a system as given by Equation 1
$*$ = denotes estimate of the unknown true value

Methodology

The solution to such an optimization problem is not trivial. The response function is unknown: the 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. One solution 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*.

The overall methodology can be summarized by the following steps:

1. Draw a large number of values from the RVs for the $N_i \cdot (N_i + 1)/2$ 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. Cycle the realizations wherever possible to try to achieve orthogonality of the resulting system:

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

3. Calculate the objective function, O, in Equation 2:

(a) Perform singular value decomposition (SVD) on the design matrix. [1]

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

i. Calculate the response associated to the L realizations at the base case values; this can be approximated by a Taylor series expansion expressed up to the second order terms:

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

where $\Delta V_i^l = V_i^l - V_i^0, i = 1, \ldots, N_i$.

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 $L \neq N_i \cdot (N_i + 1)/2$).

iii. Calculate the difference between the estimate and the true values for the sensitivity terms, and calculate the objective function (Equation 2).

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

5. If $O' < O$, then accept the change and set $\mathbf{D'}$=$\mathbf{D}$; otherwise, accept the change with probability, $p = e^{-(O'-O)/T}$, where T is the temperature parameter based on the annealing schedule. Repeat Step 4, until the number of perturbations (set by the user and controlled by the annealing schedule) is reached.

Following a post-processing of the set of realizations, as determined by the design matrix, the main effect or impact of each input variable can be calculated. The main effect is the average effect of that variable on the response value taken over the various values of the other input variables [10], and can be estimated by [6, 10]:

$$M(V_i) = \sum_{l=1}^{L} d_i \cdot r^l, \qquad \forall i = 1, \ldots, N_i \tag{3}$$

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, $r^l, l = 1, \ldots, L$.

For example, let's consider the design matrix in Table 1 and say the first variable. Suppose that the response value is NPV and that it is available for each of the eight realizations (i.e. they have been processed through to the transfer function): \$5, \$10, \$7, \$8, \$24, \$2, −\$8, and −\$4 for the eight runs, respectively. The main effect of variable one is calculated as:

[1] While a matrix solution can be obtained using any number of decomposition methods (such as Gaussian elimination or Cholesky decomposition), these other methods require certain conditions, such as symmetry of a square matrix and/or non-redundancy in the system, to be true in order to obtain a solution. Redundancy in the system is often referred to as singularity of the matrix; this is precisely the case where SVD can be used. Given the flexibility of the algorithm and although the solution may be non-unique, SVD is robust in handling under- and over-determined systems.

$$
\begin{aligned}
M(V_1) &= \sum_{l=1}^{8} d_1 \cdot r^l \\
&= (1)(5) + (-1)(10) + (-1)(7) + (1)(8) + (-1)(24) + (1)(2) + (1)(-8) + (-1)(-4) \\
&= -30
\end{aligned}
$$

The units of the response are inconsequential in this type of calculation. As well, it is the absolute magnitude of the main effect that is important to determine the impact of a particular variable. Using the above example calculation, one could now calculate the main effect of the other six variables in the Table 1. The result of such a calculation shows that variables 5 and 6 tie for the largest main effect and NPV is least affected by variable 3.

Extension to Already Processed Realizations

Now consider the case where the L realizations from the design matrix (obtained using the above approach) are processed and resources remain to allow processing of additional realizations, say L_{add} realizations. Note that the first set of L realizations have already been jointly optimized. Now we must consider a design matrix that is effectively $L + L_{add}$ realizations, and not L realizations that we initially optimized. One could consider running the above methodology and simply change the user specified number of realizations to $L + L_{add}$; however, this will likely result in an entirely new set of realizations with little duplication of the initial run). This would effectively negate the professional and computational effort expended in the original sensitivity work.

We should capitalize on the fact that some preliminary sensitivities have already been evaluated and use this information to influence the selection of the optimal design for the L_{add} realizations. For this task, we could use the main effects determined in Equation 3 to weight the selection of variables to change during the design matrix perturbation (in Step 4). Depending on the response variable (e.g. NPV), the units of the main effects can be rather large. We can simply restandardize the main effect of each variable by:

$$
M(V_i) = \frac{\sum_{l=1}^{L} d_i \cdot r^l}{\sum_{i=1}^{N_i} \sum_{l=1}^{L} d_i \cdot r^l}, \qquad \forall i = 1, \ldots, N_i \tag{4}
$$

This type of weighting permits the variables that were initially deemed to be influential to have a larger probability of being selected to test different cases; conversely, the least important variables will have a lower probability to be selected for perturbation. Further, the realizations possibly selected for perturbation are also limited to only the L_{add} realizations.

The main caveat with this approach lies in the size of the initial design matrix that is used to calculate the main effects. If too few realizations are initially processed; the corresponding main effects may give artificially high influence indicators for certain variables and vice versa. The new design matrix will reflect the preferential selection of these variables, and lead to a less optimal design matrix for improved sensitivity results.

Application

The latest methodology is implemented in Version 2.006 of the prototype program called `dmatrix`. Details on the required parameters to execute this program are given in the Appendix.

Consider the synthetic reservoir example [12] where there are six uncertain input variables including reservoir area, production index (PI), reservoir surface, horizontal permeability, vertical permeability, and porosity. Consider only the horizontal well type scenario, and that all six inputs can take one of three possible values: a pessimistic, base and an optimistic value. The size of this problem is $3^6 = 729$ possible combinations; however, there is only time to process say 7 realizations through to flow simulation. The following table shows the resulting design matrix based on the proposed methodology:

Realization	**Variables**					
	Area	**PI**	**Surface**	**VPerm**	**HPerm**	**Porosity**
1	0	0	0	0	0	0
2	0	-1	1	-1	0	1
3	1	0	-1	1	-1	0
4	0	1	0	-1	1	-1
5	-1	0	1	0	-1	1
6	1	-1	0	1	0	-1
7	-1	1	-1	0	1	0

Table 3: Design matrix for the case of 6 variables, 3 possible outcomes and 7 realizations.

For this finite example, the flow simulation results of the full combination of scenarios are available; all scenarios were put through to flow simulation and the net present value (NPV) was calculated [12]. The true sensitivities are shown in Figure 1. For the design matrix in Table 3, calculation of the main effects showed the three main variables that impact NPV are (in descending order) the Reservoir Area, vertical permeability and the reservoir surface. An assessment of the main effects based on the full factorial shows the three main variables are (in descending order) the Reservoir Area, PI and porosity. Table 5 compares the ranking of the six variables based on the main effects, and clearly shows that we are only able to correctly predict one of the ranked sensitivities (the most important variable no less).

Using the sensitivity results of the initial 7 realizations, suppose we want to add another 5 realizations to the sensitivity analysis. Application of `dmatrix` for this case yields the design matrix shown in Table 4. Running this set of realizations and retrieving the response values, the main effects of each variable can be calculated. This showed that the NPV was most sensitive to the reservoir area (variable 1), followed by the horizontal permeability, top surface alteration, porosity, vertical permeability and least affected by the production index. Comparing these results to the reference sensitivities shows that considering an additional 5 realizations permits the correct ranking of another variable (see Table 5).

As the number of realizations increases towards the full factorial scenario, this rank ordering should become more stable; however, the number of realizations required to reach this stability using this design matrix approach will vary depending on the random number seed and the annealing schedule.

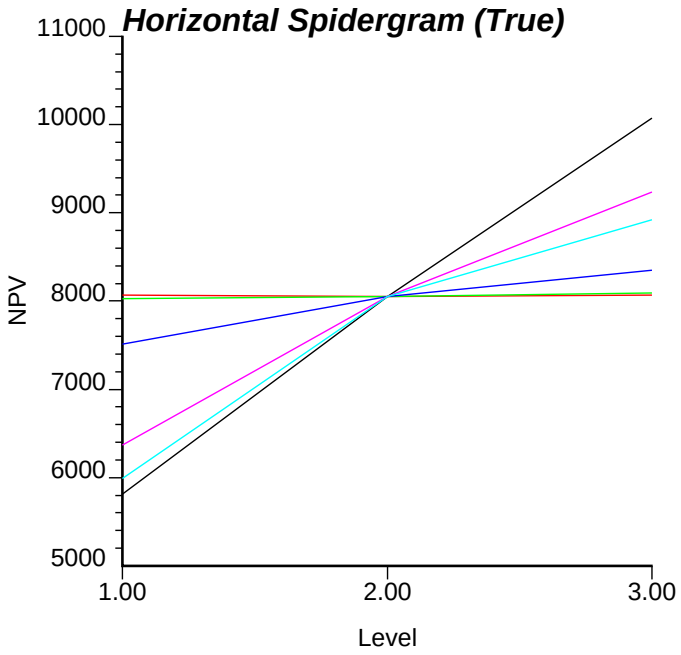

Figure 1: Spider graph showing sensitivity results from the full factorial analysis of the synthetic reservoir setting. (Source: Zanon et.al., 2005)

Realization	**Variables**					
	Area	**PI**	**Surface**	**VPerm**	**HPerm**	**Porosity**
1	0	0	0	0	0	0
2	0	-1	1	-1	0	1
3	1	0	-1	1	-1	0
4	0	1	0	-1	1	-1
5	-1	0	1	0	-1	1
6	1	-1	0	1	0	-1
7	-1	1	-1	0	1	0
8	-1	-1	1	0	0	1
9	1	-1	-1	1	0	0
10	0	1	-1	-1	1	0
11	0	0	1	-1	-1	1
12	1	0	0	1	-1	-1

Table 4: Design matrix for the case of 6 variables, 3 possible outcomes extended to 12 realizations.

Variable	Reference	DMatrix with $L = 7$	DMatrix with $L = 12$	Vary One at a Time
Reservoir Area	1	**1**	**1**	**1**
Porosity	2	4	4	**2**
Prod. Index	3	6	6	6
Hor. Perm.	4	5	2	5
Vert. Perm	5	2	**5**	4
Surface	6	3	3	3

Table 5: Comparison of sensitivity ranking of NPV to the input variables. Matches between the true ranking and the `dmatrix` results are shown in bold font.

Realization	Variables					
	Area	**PI**	**Surface**	**VPerm**	**HPerm**	**Porosity**
1	0	0	0	0	0	0
2	-1	0	0	0	0	0
3	1	0	0	0	0	0
4	0	-1	0	0	0	0
5	0	1	0	0	0	0
6	0	0	-1	0	0	0
7	0	0	1	0	0	0
8	0	0	0	-1	0	0
9	0	0	0	1	0	0
10	0	0	0	0	-1	0
11	0	0	0	0	1	0
12	0	0	0	0	0	-1
13	0	0	0	0	0	1

Table 6: Design matrix for the vary-one-at-a-time approach.

The vary-one-at-a-time approach was also examined and produced the rankings shown in Table 5. In order to determine these rankings, 13 realizations were required (see corresponding matrix in Table 6). Similar to the extended 12 realizations case, only two of the six rankings are correctly predicted, however, this required processing an extra realization over the design matrix approach.

Figure 2 shows the instability of this rank ordering if a Monte Carlo drawing of different sets of realizations is used (rather than the design matrix approach).

Discussion

This algorithm is flexible in terms of the user's problem specification: number of input and response variables, number of cases each variable can take, and the number of realizations desired for processing. As such, the degrees of the freedom imposed on the solution space for an optimal design matrix is immensely large; the search for a unique solution is a challenge. The proposed methodology is a stochastic approach to this optimization, the resulting matrix is not a unique solution; however, the use of simulated annealing presents

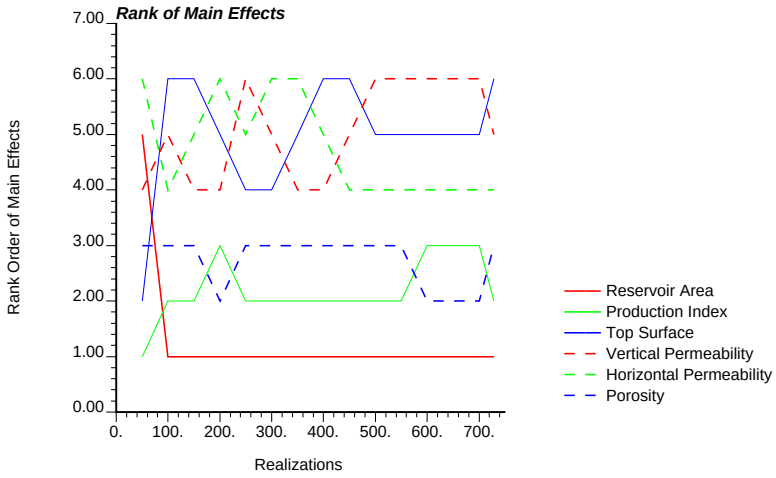

Figure 2: Sensitivity of rank order of the main effects based on MCS.

some promise in determining a more globally optimal design.

The computational time required to execute the program is directly related to the simulated annealing (SA) algorithm employed for optimization. Convergence of the SA algorithm depends on the annealing schedule; a balance must be struck between computational time and optimality of the resulting design matrix. A large design matrix will also contribute to the computational time required. Although some preliminary testing was conducted to obtain reasonable computational effort and convergence, more testing and analysis are required for thorough documentation of this issue. Future documentation should also focus on the convergence of the design matrix rank order sensitivities as a function of the number of realizations, and a comparison of this rank order to other conventional sensitivity assessment approaches.

Extension of this algorithm to allow for additional realizations to be computed given an already optimized design matrix was also implemented. This essentially builds 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. The main effects of the previously run scenarios are used to determine the significant variables, and thus can be used to preferentially select certain variables for perturbation. Future work in this area should consider the response results from the initial run. These results can be used to estimate a response surface, using kriging or some other interpolation method. The reference sensitivity terms, that are stochastically attained in the initial run, can now be numerically determined using this response surface. This should permit a more efficient design.

In both the initial and extension cases, evaluating the main effects shows that the sensitivity results can be highly variable. Use of the Taylor series expansion in determining the response values can be refined by using the distribution of response values; however, this presumes that one is readily available prior to post-processing of simulation results.

Sensitivity analysis remains a challenge - especially given that post-processing for natural resource management requires time-intensive (flow) simulations. This ultimately leaves the practitioner with few options but to run only a handful of models through sensitivity analysis. The proposed design matrix methodology presents a means of selecting the realizations to be processed and evaluated.

References

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

[2] 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.

[3] C. V. Deutsch. *Annealing Techniques Applied to Reservoir Modeling and the Integration of Geological and Engineering (Well Test) Data*. PhD thesis, Stanford University, Stanford, CA, 1992.

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

[5] D. Lin and J. Chang. A note on cyclic orthogonal designs. *Statistica Sinica*, 11(2):549–552, 2001.

[6] 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.

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

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

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

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

[11] C. Wu and M. Hamada. *Experiments: Planning, Analysis and Parameter Design Optimization*. John Wiley & Sons Inc., New York, 2000.

[12] S. Zanon, L. Cunha, O. Leuangthong, and C. Deutsch. Short note on the quantification of the effect of input variable uncertainty in oil reservoir performance prediction. Technical report, Centre for Computational Geostatistics, University of Alberta, Edmonton, AB, September 2005.

Appendix

An example parameter file for `dmatrix` is shown in Figure 3 and the corresponding parameters 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.
- **outfl**: file for output. This file contains the optimum design matrix.
- **sumfl**: file with summary information about the convergence of the objective function.
- **maxpert, rreport**: maximum number of perturbations at a specified temperature; used to determine a stopping criteria.
- **maxnochange**: maximum iterations with no change in objective function; used to determine a stopping criteria.
- **iext, ladd**: flag to determine if additional realizations are desired (0=no, 1=yes); if yes, then specify number of additional realizations.
- **resfl**: file with previously determined design matrix, plus the corresponding response variable value (this should look like the dmatrix.out file with a results column appended to it).

```
                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
dmatrix.out               - output file for design matrix
dmatrix.sum               - summary file to report objective functions


SIMULATED ANNEALING PARAMETERS:
10  0.1                   - maximum number of perturbations, reporting
100                       - maximum perturbations without a change


EXTEND TO MORE REALIZATIONS:
0   5                     - more realizations(0=no,1=yes), no. realizations to add
results.out               - file with prev. dmatrix and extra column for response
```

Figure 3: Parameters for `dmatrix`.