Commun. Comput. Phys.
doi: 10.4208/cicp.230909.160310s

Vol. 9, No. 3, pp. 542-567
March 2011

Variance-Based Global Sensitivity Analysis via Sparse-Grid Interpolation and Cubature

Gregery T. Buzzard and Dongbin Xiu*

Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA.

Received 23 September 2009; Accepted (in revised version) 16 March 2010

Available online 17 September 2010

To the memory of David Gottlieb

Abstract. The stochastic collocation method using sparse grids has become a popular choice for performing stochastic computations in high dimensional (random) parameter space. In addition to providing highly accurate stochastic solutions, the sparse grid collocation results naturally contain sensitivity information with respect to the input random parameters. In this paper, we use the sparse grid interpolation and cubature methods of Smolyak together with combinatorial analysis to give a computationally efficient method for computing the global sensitivity values of Sobol'. This method allows for approximation of all *main effect* and *total effect* values from evaluation of f on a single set of sparse grids. We discuss convergence of this method, apply it to several test cases and compare to existing methods. As a result which may be of independent interest, we recover an explicit formula for evaluating a Lagrange basis interpolating polynomial associated with the Chebyshev extrema. This allows one to manipulate the sparse grid collocation results in a highly efficient manner.

AMS subject classifications: 41A10, 41A05, 41A63, 41A55

Key words: Stochastic collocation, sparse grids, sensitivity analysis, Smolyak, Sobol'.

1 Introduction

The growing popularity of computational models in various fields of science and engineering has led to a corresponding growth in interest in methods to understand parameter spaces. A common task in developing a model is to find parameters $p=(p_1,\cdots,p_n)$ to minimize some cost function, $C(p)$, often a sum of squared differences between model output and experimental data. This is a particularly difficult task when the dimensionality of the parameter space is large and the dependence of C on p is nonlinear. Hence, the development of efficient tools for reducing the size of the search space is vital.

*Corresponding author. *Email addresses:* buzzard@math.purdue.edu (G. T. Buzzard), dxiu@math.purdue.edu (D. Xiu)

Local sensitivity analysis, which computes partial derivatives $\partial C/\partial p_j$, may be used to determine which parameters are relatively more important than others at a single point in the parameter space. In contrast, global sensitivity analysis (GSA) seeks a measure of relative importance over the entire parameter space. One approach to GSA is variance based. With an appropriate measure, we may regard the parameter space as a probability space and C as a random variable on this space. The sensitivity of C to a given parameter p_j is then

$$\frac{\mathrm{Var}_{p_j}\big(E[C|p_j]\big)}{\mathrm{Var}(C)},$$

where $E[C|p_j]$ is the expected value obtained by fixing a given value of p_j and integrating over the remaining variables, and Var_{p_j} is the variance as a function of p_j only. As noted by Saltelli et al. (see [6] and the references therein), this quantity has appeared in various formulations in many places in the literature.

Sobol′ [10] provides a particularly appealing formulation of this quantity as a special case of a more general approach to GSA for arbitrary subsets of parameters. The measure above, which is meant to capture the effect of the single parameter p_j, is often called the *main effect* due to p_j. By considering other subsets containing p_j, the sensitivity values of Sobol′ can be used to identify interaction effects among multiple variables. For instance, by considering the sum of all sensitivity values in which p_j takes part, we obtain what is often called the *total effect* due to p_j. Sobol′ [10] describes a method for evaluating the main effect values using Monte Carlo or quasi-Monte Carlo methods. Saltelli et al. [6] give a Fourier analysis based method for computing both the main and total effects. This is often called the Extended FAST method.

The problem we address in this paper is to use the values of a function computed at the nodes of sparse grids to compute these sensitivity values efficiently. In practice, these function values are obtained through an expensive computation such as solving a large ODE system or a PDE, and these values are often used to construct an interpolating polynomial that approximates the function itself. Here we use both interpolation and cubature with these values to obtain all the main effect and total effect values without further simulation effort. To this end, we demonstrate that even though multi-dimensional Lagrange interpolation polynomials are not easy to manipulate in general, a highly efficient sensitivity analysis based on the Sobol′ decomposition can be accomplished by utilizing the properties of sparse grids and Chebyshev polynomials. Moreover, the algorithm presented here retains the high order approximation accuracy from the sparse-grid simulations and can be carried out completely offline.

For a detailed discussion of many approaches to sensitivity analysis as well as a discussion of the Extended FAST method and many additional references, see [8]. For other approaches to improve the efficiency of computing sensitivity values, see [2,4,7,11] and references therein. In contrast to these methods, our method is based on the idea of stochastic collocation (see, e.g., [16]), which has become popular since [16] introduced the use of sparse grids in the stochastic collocation method. The use of sparse grids alleviates (to some extent) the problem of the curse of dimensionality and allows us to approximate even the total effect sensitivity values in relatively high dimensional parameter spaces. For a detailed review of stochastic collocation methods and their relation to polynomial chaos expansion, see [15].

Some advantages of our method over existing methods include greater accuracy, computational efficiency, provable convergence rates for differentiable functions, the fact that there are no choices required to tune the algorithm to a given application, the fact that any of the sensitivity values may be approximated using the values of the function on a single set of sparse grids, and the fact that these values may be used to construct an interpolating polynomial that can act as a surrogate for the original function to guide optimization. Here we focus on the main effect and total effect values, but our approach may be applied in principle to any of the sensitivity values.

We note here that there are many choices of grids in the stochastic collocation approach. Here we focus on the sparse grids constructed using the Chebyshev extrema because of their popularity in practical computations, high order accuracy, and ease of construction. For a comparison of integration via sparse grids using various underlying quadrature rules, see [3].

As a result which may be of independent interest, we give a formula that generalizes the standard discrete orthogonality formula for Chebyshev polynomials evaluated at their extrema (Proposition 4.1). We use this to recover an explicit formula for evaluating a Lagrange basis interpolating polynomial associated with the Chebyshev extrema (Corollary 4.1). This latter formula may be derived from Lemma 6.4 of [5], but perhaps deserves greater attention since it allows for evaluation at any point in the interval $[-1,1]$ and has computational cost $\mathcal{O}(1)$, independent of the degree of the interpolating polynomial. In Section 2, we review the orthogonal decomposition of Sobol'. In Section 3, we use this decomposition to define the main and total effect sensitivity values and then prove an integral formula (Proposition 3.1) that may be seen as a generalization of Parseval's Theorem for this decomposition. This proposition also allows us to derive formulas for the desired sensitivity values. In Section 4, we review Chebyshev interpolation and the associated quadrature method of Clenshaw-Curtis, then state and derive a closed-form formula for the value at any point in $[-1,1]$ of the Lagrange interpolating polynomial associated with the Chebyshev extrema (Corollary 4.1). In Section 5, we review the ideas of sparse grid interpolation and cubature, and in Section 6, we apply these ideas in two different ways to approximate the main and total effect sensitivity values. Finally, in Section 7, we apply our method to several test functions and compare to quasi-Monte Carlo and Extended FAST.

2 Orthogonal decomposition

Here we review the key ideas of [10]. The results here are all contained in [10]; however, we provide somewhat different proofs than those contained in that paper. Also, we use the interval $[-1,1]$ for our calculation instead of the original $[0,1]$ in order to dovetail more easily with techniques of sparse grid cubature.

Let $K=[-1,1]$ and$f:K^n\to\mathbb{R}$. Let x denote coordinates on K^n, α denote a multi-index in $\{0,1\}^n$, α_j denote the j-th entry in α, $\mathbf{1}$ denote the multi-index with each entry equal to 1, $\mathbf{0}$ denote the multi-index with each entry equal to 0 and $\mathbf{1}^j$ denote the multi-index such that $\mathbf{1}^j_k=\delta_{jk}$. Also, let $\alpha'=\mathbf{1}-\alpha$ and $|\alpha|=\sum_j\alpha_j$. Define a partial order on such multi-indices by $\alpha\le\beta$ if $\alpha_j\le\beta_j$ for all j, with the obvious notion of equality. With this order, the set of γ satisfying $\alpha\le\gamma\le\beta$ is exactly the set of γ with $\alpha_j\le\gamma_j\le\beta_j$, for all j. Finally, for purposes of integration, let $d\mu_j=dx_j/2$ be normalized Lebesgue measure, let $d\mu=d\mu_1\cdots d\mu_n$, and for a multi-index $\alpha=\sum_{j=1}^k\mathbf{1}^{i_j}$, let $d\mu_\alpha=d\mu_{i_1}\cdots d\mu_{i_k}$.

For the method of Sobol′, we assume that f is L^2 integrable on K^n and look for a functional decomposition

$$f=\sum_{\mathbf{0}\le\alpha\le\mathbf{1}}f_\alpha \tag{2.1}$$

satisfying

(A) f_α is independent of x_j if $\alpha_j=0$.

(B) $\int_K f_\alpha(x)d\mu_j=0$ if $\alpha_j=1$.

(C) $\int_{K^n}f_\alpha f_\beta d\mu=0$ if $\alpha\ne\beta$.

We show below that this decomposition is achieved via the formula

$$f_\alpha(x)=\int_{K^{n-|\alpha|}}f(x)d\mu_{\alpha'}-\sum_{\mathbf{0}\le\beta<\alpha}f_\beta(x). \tag{2.2}$$

Here we make the convention that

$$\int_{K^0}f(x)d\mu_{\mathbf{1}'}=f(x).$$

Note also that

$$f_{\mathbf{0}}(x)=\int_{K^n}f(x)d\mu$$

is a constant function with value equal to the expected value of f in K^n, assuming uniform distribution in each variable. From now on we use f_0 for this expected value. The formula in (2.2) agrees with the definition given in [10], although the presentation is more explicit here.

Proposition 2.1. *With f_α as in* (2.2)*, the decomposition* (2.1) *holds, and statements* (A), (B) *and* (C) *are true.*

Proof. We show first that

$$f=\sum_{0\leq\alpha\leq\mathbf{1}} f_\alpha.$$

By definition of $f_{\mathbf{1}}$, we have

$$\begin{aligned}\sum_{\mathbf{0}\leq\alpha\leq\mathbf{1}} f_\alpha &= f_{\mathbf{1}}+\sum_{\mathbf{0}\leq\alpha<\mathbf{1}} f_\alpha\\ &=\Big(f-\sum_{\mathbf{0}\leq\alpha<\mathbf{1}} f_\alpha\Big)+\sum_{\mathbf{0}\leq\alpha<\mathbf{1}} f_\alpha=f.\end{aligned}$$

To prove statement (A), we induct on $|\alpha|$. Note that f_0 is constant, hence independent of all x_j. Also, if $\alpha_j=0$, for some $\alpha\neq\mathbf{0}$, then $\beta_j=0$, for all $\beta<\alpha$, and hence $f_\beta(x)$ is independent of x_j by induction. Moreover, in this case, α' has $\alpha'_j=1$, so $\int_{K^{n-|\alpha|}} f(x)d\mu_{\alpha'}$ is independent of x_j. Using this with the definition of f_α in (2.2) verifies statement (A).

For statement (B), again we induct on $|\alpha|$. If $\alpha=\mathbf{0}$, statement (B) is vacuously true. Next suppose that $\alpha_j=1$, and let $\hat{\alpha}=\alpha-\mathbf{1}^j$. Note that if $\mathbf{0}\leq\beta<\alpha$, then either $\beta<\hat{\alpha}$, or $\beta=\hat{\alpha}$, or $\mathbf{1}^j\leq\beta<\alpha$, and exactly one of these three cases holds. Using this with (2.2), we have

$$\begin{aligned}\int_K f_\alpha(x)d\mu_j &= \int_K\int_{K^{n-|\alpha|}} f(x)d\mu_{\alpha'}d\mu_j-\sum_{0\leq\beta<\alpha}\int_K f_\beta(x)d\mu_j\\ &=\int_{K^{n-|\hat{\alpha}|}} f(x)d\mu_{\hat{\alpha}'}-\sum_{0\leq\beta<\hat{\alpha}}\int_K f_\beta(x)d\mu_j\\ &\quad-\int_K f_{\hat{\alpha}}(x)d\mu_j-\sum_{\mathbf{1}^j\leq\beta<\alpha}\int_K f_\beta(x)d\mu_j.\end{aligned}$$

Note that the first two terms in the final expression are exactly the definition of $f_{\hat{\alpha}}(x)$. Also, since $\hat{\alpha}_j=0$, we see that $f_{\hat{\alpha}}(x)$ is independent of x_j by statement (A). Hence the third term is also exactly $f_{\hat{\alpha}}(x)$, thus cancelling the first two terms. Moreover, each integral in the final summation is 0 by induction since $\mathbf{1}^j\leq\beta<\alpha$ implies $|\beta|<|\alpha|$ and $\beta_j=1$. Thus the terms of the final expression cancel to verify statement (B).

Statement (C) is true exactly as presented in Sobol'. That is, if $\alpha\neq\beta$, then without loss, there is some j so that $\alpha_j=1$ while $\beta_j=0$. Integrating $f_\alpha f_\beta$ with respect to x_j and applying statement (B) gives 0. Hence the entire integral is 0. □

3 Variance and sensitivity values

In this section, we define the variance and sensitivity values as in Sobol', then state and prove Proposition 3.1, which is a generalization of Parseval's Theorem to partial sums of the Sobol' orthogonal decomposition. This result is the foundation for our method of computing sensitivity coefficients. While the case $|\alpha|=1$ of this result appears in [10], this generalized form is, to our knowledge, novel.

Definition 3.1. *Let α be a multi-index and let $1 \le j \le n$. Define*

$$D_\alpha = \int_{K^n} f_\alpha^2 d\mu - \Big(\int_{K^n} f_\alpha d\mu \Big)^2,$$

and

$$D_{T_j} = \sum_{\mathbf{1}^j \le \alpha \le \mathbf{1}} D_\alpha, \quad S_\alpha = \frac{D_\alpha}{D}, \quad S_{T_j} = \frac{D_{T_j}}{D}.$$

Here D_α is the variance of f_α; denote the variance of f by D (replace f_α by f in the expression for D_α). Also, D_{T_j} is called the total variance for x_j, S_α is called the sensitivity for α and S_{T_j} is called the total sensitivity of x_j (S_{T_j}).

By abuse of notation, let $D_j = D_{\mathbf{1}^j}$ and $S_j = S_{\mathbf{1}^j}$. Note that S_j is often called the main effect due to x_j and S_{T_j} is called the total effect due to x_j.

The next proposition is the main tool for computing sensitivity values via cubature. Here and below, $\|f\|_2$ denotes the L^2 norm of f over K^n with measure μ.

Theorem 3.1. *Let α be a multi-index. Then*

$$\sum_{\mathbf{0} \le \beta \le \alpha} \|f_\beta\|_2^2 = \int_{K^{|\alpha|}} \Big(\int_{K^{n-|\alpha|}} f(x) d\mu_{\alpha'} \Big)^2 d\mu_\alpha.$$

In the case $\alpha = \mathbf{1}$, this statement is a version of Parseval's Theorem for the orthogonal decomposition given by the f_α. Hence this equality may be viewed as a generalization of Parseval's Theorem to the case of partial sums of the functions in the decomposition.

Proof. For $\alpha = \mathbf{0}$, the statement is true by definition. Hence, we assume $\alpha \ne \mathbf{0}$. In this case, property (B) implies that

$$\int_{K^n} f_\alpha d\mu = 0.$$

Combining this with property (A) and the fact that μ_j is a probability measure, we see that

$$D_\alpha = \int_{K^n} f_\alpha^2 d\mu - \Big(\int_{K^n} f_\alpha d\mu \Big)^2 = \int_{K^{|\alpha|}} f_\alpha^2(x) d\mu_\alpha.$$

Using this together with (2.2) and the definition of $\|f\|_2$, we have

$$\begin{aligned} \|f_\alpha\|_2^2 &= \int_{K^{|\alpha|}} \Big(\int_{K^{n-|\alpha|}} f(x) d\mu_{\alpha'} - \sum_{0 \le \beta < \alpha} f_\beta(x) \Big)^2 d\mu_\alpha \\ &= \int_{K^{|\alpha|}} \Big[\Big(\int_{K^{n-|\alpha|}} f d\mu_{\alpha'} \Big)^2 - 2 \sum_{0 \le \beta < \alpha} f_\beta \int_{K^{n-|\alpha|}} f d\mu_{\alpha'} + \Big(\sum_{0 \le \beta < \alpha} f_\beta \Big)^2 \Big] d\mu_\alpha. \end{aligned}$$

Note that if $\beta < \alpha$, then f_β depends only on the variables in x_α and hence is independent of the variables in $x_{\alpha'}$. Since μ_j is a probability measure, we may integrate the final term over K^n instead of $K^{|\alpha|}$. Using this, expanding the square and applying the orthogonality result of property (C) and the independence result of (A) gives

$$\int_{K^{|\alpha|}}\Big(\sum_{0\le\beta<\alpha} f_\beta\Big)^2 d\mu_\alpha=\int_{K^n}\sum_{0\le\beta<\alpha}\sum_{0\le\gamma<\alpha} f_\beta f_\gamma d\mu=\sum_{0\le\beta<\alpha}\|f_\beta\|_2^2.$$

This same idea, plus the functional decomposition of (2.1) implies that if $\beta<\alpha$, then

$$\int_{K^{|\alpha|}} f_\beta\Big(\int_{K^{n-|\alpha|}} f(x)d\mu_{\alpha'}\Big)d\mu_\alpha=\int_{K^n} f_\beta\sum_{0\le\gamma\le\mathbf{1}} f_\gamma d\mu=\|f_\beta\|_2^2.$$

Applying these previous two results in the expression for $\|f_\alpha\|_2^2$, we see that

$$\|f_\alpha\|_2^2=\int_{K^{|\alpha|}}\Big(\int_{K^{n-|\alpha|}} f(x)d\mu_{\alpha'}\Big)^2 d\mu_\alpha-\sum_{0\le\beta<\alpha}\|f_\beta\|_2^2.$$

Moving the final sum to the left hand side gives the proposition. □

Using this result, we have the following four corollaries. These results appear also in [10], but the previous proposition provides a simple, direct proof for each. First we express D in terms of the D_α, then express each of D_j and D_{T_j} as an integral, where D is the variance of f, D_j is the variance for the main effect for x_j, and D_{T_j} is the variance for the total effect for x_j.

Corollary 3.1.

$$D=\sum_{\mathbf{0}\le\alpha\le\mathbf{1}} D_\alpha=\sum_{\mathbf{0}<\alpha\le\mathbf{1}}\|f_\alpha\|_2^2.$$

Proof. Note that property (B) implies that $E[f_\alpha]=0$ when $\alpha\neq\mathbf{0}$, in which case $D_\alpha=\|f_\alpha\|_2^2$. Also, since $f_{\mathbf{0}}(x)$ is constant, $D_{\mathbf{0}}=0$.

Applying this with the previous proposition and the formula for f_0 gives

$$\sum_{\mathbf{0}\le\alpha\le\mathbf{1}} D_\alpha=\Big(\sum_{\mathbf{0}\le\alpha\le\mathbf{1}}\|f_\alpha\|_2^2\Big)-\|f_0\|_2^2=\int_{K^n} f^2 d\mu-\Big(\int_{K^n} f d\mu\Big)^2.$$

The final expression is exactly D. □

Corollary 3.2.

$$D_j=\int_K\Big(\int_{K^{n-1}} f d\mu_{\mathbf{1}_j'}\Big)^2 d\mu_j-f_0^2.$$

Proof. Let I denote the integral in the expression above. Theorem 3.1 implies that

$$I=\sum_{0\leq\alpha\leq\mathbf{1}^j}\|f_\alpha\|_2^2.$$

Since only 0 and $\mathbf{1}^j$ satisfy the bounds on α in the sum, we have

$$I=\|f_0\|_2^2+\|f_{\mathbf{1}^j}\|_2^2=f_0^2+D_j,$$

hence $D_j=I-f_0^2$. □

Corollary 3.3.

$$D_{T_j}=D+f_0^2-\int_{K^{n-1}}\Big(\int_K f d\mu_j\Big)^2 d\mu_{\mathbf{1}^{j'}}.$$

Proof. By definition and properties of D_α,

$$D_{T_j}=\sum_{\mathbf{1}^j\leq\alpha\leq\mathbf{1}}D_\alpha=\sum_{0\leq\alpha\leq\mathbf{1}}\|f_\alpha\|_2^2-\sum_{0\leq\alpha\leq\mathbf{1}^{j'}}\|f_\alpha\|_2^2.$$

From Corollary 3.1, the first of the final two sums is $D+f_0^2$. From the proposition, the second of the final two sums is exactly the integral in the statement of this corollary. □

The following corollary follows directly from the previous corollaries using the standard proof for showing that

$$E[(X-\overline{X})^2]=E[X^2]-\overline{X}^2,$$

where X is a random variable and $\overline{X}=E[X]$.

Corollary 3.4. *As before, let*

$$f_0=\int_{K^n}f d\mu$$

be the mean of f *and*

$$D=\int_{K^n}f^2 d\mu-f_0^2$$

be the variance. Then the main effect sensitivity for x_j *on* f *is*

$$S_j=\frac{1}{D}\int_K\Big(\int_{K^{n-1}}(f-f_0)d\mu_{\mathbf{1}^{j'}}\Big)^2 d\mu_j,$$

and the total effect sensitivity for x_j *on* f *is*

$$S_{T_j}=1-\frac{1}{D}\int_{K^{n-1}}\Big(\int_K(f-f_0)d\mu_j\Big)^2 d\mu_{\mathbf{1}^{j'}}.$$

4 Chebyshev interpolation and Clenshaw-Curtis quadrature

In this section, we review some results about interpolation and quadrature in one dimension, then give a formula that generalizes a standard discrete orthogonality formula for Chebyshev polynomials. We use this to recover an explicit formula for evaluating a Lagrange basis interpolating polynomial associated with the Chebyshev extrema. Much of this section is well-known and based on "Chebyshev Polynomials" by J. C. Mason and D. C. Handscomb [5]. However, Proposition 4.1 and Corollary 4.1, which are consequences of the Christoffel-Darboux identity for orthogonal polynomials and the abstract formula for the value of a Lagrange interpolating polynomial, do not appear to be well-known in this explicit form. For a fascinating account of the convergence properties of Clenshaw-Curtis quadrature and Gaussian quadrature, see [13].

Recall that the Chebyshev polynomial of the first kind of degree $d \geq 0$ is denoted T_d and satisfies

$$T_d(\cos\theta) = \cos d\theta.$$

Within the interval $[-1,1]$, the extrema of T_d occur at the points

$$x_{d,k} = \cos\left(\frac{k\pi}{d}\right), \qquad k = 0,\cdots,d,$$

where $d > 0$. By convention, let $x_{0,0} = 0$. For sparse grid cubature as in the next section, we use Lagrange interpolation on these extrema of T_d. Let $L_{d,i}(x)$ denote the degree d polynomial, which satisfies $L_{d,i}(x_{d,k}) = \delta_{i,k}$ for $k = 0,\cdots,d$. Here and below, $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise. Given a function f on $[-1,1]$, the degree d polynomial interpolating f on the points $x_{d,k}$ is then

$$J_d f(x) = \sum_{j=0}^{d} f(x_{d,j}) L_{d,j}(x). \tag{4.1}$$

This can be rewritten in terms of T_j and applied to $f(x) = L_{d,i}(x)$ (see [5], pp. 189) to obtain

$$L_{d,i}(x) = \frac{2}{d(1+\delta_{i,0}+\delta_{i,d})} \sum_{j=0}^{d}{}'' T_i(x_{d,j}) T_j(x). \tag{4.2}$$

To approximate

$$I(f) = \int_{-1}^{1} f(x)dx,$$

first, let

$$U_d(f) = \int_{-1}^{1} J_d f(x)dx.$$

By integrating (4.1), we obtain

$$U_d(f) = \sum_{j=0}^{d} w_{d,j} f(x_{d,j}), \tag{4.3}$$

where

$$w_{d,j} = \int_{-1}^{1} L_{d,j}(x)dx.$$

The formula (4.3) gives $I(f)$ exactly for polynomials of degree at most d. For d even, which we apply later in calculating weights for the sensitivity analysis, there is a fairly simple and well-known formula (see, e.g., [5]). Here, the notation $\sum''$ means that the first and last terms in the series are divided by 2 before being added to the sum

$$\int_{-1}^{1} L_{2d,j}(x)dx = \frac{2}{d}\sum_{k=0}^{d}{}'' \frac{1}{1-4k^2}\cos\left(\frac{jk\pi}{d}\right). \tag{4.4}$$

In order to approximate the variances of the previous section, we need to integrate a product of interpolated functions, hence we need to integrate products of the Lagrange interpolants $L_{d,j}$. First some preliminary lemmas on sums of products of T_j. The first is a standard result on a kind of discrete orthogonality.

Lemma 4.1. (see, e.g., [5, (4.45)]) *Let d be a positive integer and $0 \le i, j \le d$. Then*

$$\sum_{k=0}^{d}{}'' T_i(x_{d,k})T_j(x_{d,k}) = d\frac{\delta_{i,j}(1+\delta_{0,i}+\delta_{i,d})}{2}.$$

Next, we need a similar result in which the point of evaluation is different for T_j and T_k. As pointed out by one of the referees, the formula in this proposition is a consequence of the Christoffel-Darboux identity for orthogonal polynomials [14]. However, the explicit form given here for Chebyshev polynomials does not appear to be widely known and is not immediately apparent from [14]. We give an elementary, self-contained proof below.

Proposition 4.1. *Let d_1, d_2, j_1, j_2 be integers, with $0 \le j_1 \le d_1 < d_2$ and $0 \le j_2 \le d_2$. Let $x_1 = x_{d_1,j_1}$, $x_2 = x_{d_2,j_2}$ and suppose $x_1 \ne x_2$. Then*

$$\sum_{k=0}^{d_1}{}'' T_{j_1}(x_{d_1,k})T_{j_2}(x_{d_2,k}) = \frac{(-1)^{j_1}\sqrt{1-x_2^2}\sin(j_2 d_1 \pi/d_2)}{2(x_1-x_2)}.$$

Note that Lemma 4.1 fills in the gap left here by the assumption that $x_1 \ne x_2$. That is, if $x_1 = x_2$, then we may replace j_2 with j_1 and d_2 with d_1 without changing the sum, in which case we may apply Lemma 4.1.

The following immediate corollary gives a method to evaluate the Lagrange basis interpolating polynomial, $L_{d,j}$ at any point $x \in [-1,1]$ in constant time (independent of the degree) without calculating the polynomial itself or applying (4.2). This formula may also be derived from Lemma 6.4 of [5] using the fact that the extrema of T_d are the zeros of the polynomial $(1-x^2)U_{d-1}(x)$, where U_{d-1} is the Chebyshev polynomial of the second kind.

Corollary 4.1. *Let d,j be integers with $d>0$ and $0\leq j\leq d$, let $x_0=\cos(j\pi/d)$, let*

$$C_{d,j}=\frac{(-1)^{j+1}}{d(1+\delta_{0,j}+\delta_{d,j})},$$

and let $x\in[-1,1]$. Then

$$L_{d,j}(x)=C_{d,j}\frac{\sqrt{1-x^2}\sin\left(d\cos^{-1}(x)\right)}{x-x_0},\quad x\neq x_0, \tag{4.5}$$

and $L_{d,j}(x_0)=1$.

Note that by replacing x with $\cos\theta$, we obtain an expression analogous to $T_d(\cos\theta)=\cos(d\theta)$:

$$L_{d,j}(\cos\theta)=C_{d,j}\frac{\sin\theta\sin d\theta}{\cos\theta-\cos\left(j\pi/d\right)},\quad \theta\neq\frac{j\pi}{d}.$$

Note also that the appearance of sin, $\cos^{-1}$, and square root in the formula for $L_{d,j}$ means that (4.5) is a less efficient way to calculate $L_{d,j}(x)$ for small d (by a factor of five or so in some brief tests using Matlab) than using the explicit polynomial representation of $L_{d,j}$ obtained by expanding (4.2) and collecting coefficients. However, this advantage is erased in our tests if $L_{d,j}(x)$ is evaluated directly as in (4.2). Moreover, (4.5) is very easy to implement in the general case, and since the calculation of the values of $L_{d,j}(x)$ takes a relatively small proportion of the total computational time, we use (4.5) for $d\geq 2$.

Proof of Corollary 4.1. Apply (4.2) to the result of Proposition 4.1 to get a formula for $L_{d,j}(x_{d_2,j_2})$. Then note that

$$\sin\left(\frac{j_2 d\pi}{d_2}\right)=\sin\left(d\cos^{-1}(x_{d_2,j_2})\right)$$

to obtain the formula in this corollary with $x=x_{d_2,j_2}$. Finally, note that the points x_{d_2,j_2} over all $d_2>d$ and $0\leq j_2\leq d_2$ are dense in $[-1,1]$, so that the left and right hand functions must be equal on all of $[-1,1]$ by continuity. The continuity of the right hand side at $x=x_0$ follows from the fact that

$$\sin\left(d\cos^{-1}(x_0)\right)=\sin\left(\frac{dj\pi}{d}\right)=0,$$

which implies that the right hand side has a removable singularity at x_0. □

Note that (4.3) together with this corollary gives an efficient method for evaluating the L^2 inner product of two Lagrange interpolating functions. That is,

$$\int_{-1}^{1}L_{d_1,j_1}(x)L_{d_2,j_2}(x)dx=\sum_{j=0}^{2d_2}w_{2d_2,j}L_{d_1,j_1}(x_{2d_2,j})L_{d_2,j_2}(x_{2d_2,j}), \tag{4.6}$$

which may be evaluated using Corollary 4.1 with a number of operations which is linear in the maximum degree. For this application, we may avoid the use of $\cos^{-1}$ since

$$\cos^{-1}(x_{2d_2,j}) = \frac{j\pi}{2d_2}.$$

To prove the proposition, we use a function S_a so that

$$S_a(x+1) - S_a(x) = \cos ax$$

in order to convert the sum to a telescoping sum.

Definition 4.1. *For $a \in \mathbb{R}$ with $\sin(a/2) \neq 0$, let*

$$S_a(x) = \frac{\sin\left(ax - a/2\right)}{2\sin\left(a/2\right)}.$$

Lemma 4.2. *For $a \in \mathbb{R}$ with $\sin(a/2) \neq 0$,*

$$\sum_{k=0}^{d} \cos(ak) = S_a(d+1) + \frac{1}{2}.$$

Proof. Using the formula for the difference of two sines, we have

$$\sin\left(a(x+1) - \frac{a}{2}\right) - \sin\left(ax - \frac{a}{2}\right) = 2\sin\left(\frac{a}{2}\right)\cos(ax),$$

so

$$S_a(k+1) - S_a(k) = \cos ak.$$

Using this in place of $\cos(ak)$ in the sum, canceling adjacent terms, and using $S_a(0) = -1/2$ gives the result. □

Proof of Proposition 4.1. Let

$$a^{\pm} = \pi\left(\frac{j_1}{d_1} \pm \frac{j_2}{d_2}\right).$$

Since j_1/d_1 and j_2/d_2 are both in $[0,1]$, the only way that $\sin(a^{\pm}/2)$ could be 0 is if $x_1 = x_2$, which is excluded by assumption. Hence

$$\sin\left(\frac{a^{\pm}}{2}\right) \neq 0.$$

Using the formula for the product of cosines, we have

$$T_{j_1}(x_{d_1,k}) T_{j_2}(x_{d_2,k}) = \frac{\cos(a^+ k) + \cos(a^- k)}{2}.$$

Applying the previous lemma and adjusting for the factors of $1/2$ for $k=0$ and $k=d_1$ gives

$$\sum_{k=0}^{d_1}{}'' T_{j_1}(x_{d_1,k})T_{j_2}(x_{d_2,k}) = \Big(\frac{S_{a^+}(d_1+1)+S_{a^-}(d_1+1)}{2}\Big) - \frac{(-1)^{j_1}}{2}\cos\Big(\frac{j_2 d_1 \pi}{d_2}\Big). \tag{4.7}$$

Let S denote the numerator of the first term. We expand using the definition of S_a, combine fractions, and use $2\sin a \sin b = \cos(a-b)-\cos(a+b)$. Writing

$$a_k^{\pm} = a^{\pm} d_1 + \frac{j_k \pi}{d_k}$$

we have

$$S = \frac{\cos(a_2^+) - \cos(a_1^+) + \cos\big(a^- d_1 - j_2\pi/d_2\big) - \cos(a_1^-)}{\cos\big(j_2\pi/d_2\big) - \cos\big(j_1\pi/d_1\big)}.$$

Rewriting the denominator as $x_2 - x_1$ and combining the first and third terms of the numerator and likewise the second and fourth terms, we get

$$S = \frac{\cos(j_1\pi)\cos\big(j_2 d_1 \pi/d_2 + j_2\pi/d_2\big) - \cos\big(j_1\pi + j_1\pi/d_1\big)\cos\big(j_2 d_1\pi/d_2\big)}{x_2 - x_1}.$$

Applying the usual angle addition formula for cos, then rewriting in terms of x_1 and x_2 gives

$$S = (-1)^{j_1}\Big[T_{d_1}(x_2) - \Big(\sqrt{1-x_2^2}\sin\big(j_2 d_1 \pi/d_2\big)\Big)(x_2 - x_1)^{-1}\Big].$$

Dividing by 2 and subtracting the second term in (4.7) cancels $T_{d_1}(x_2)$ and leaves the expression in the statement of the proposition. □

Remark 4.1. Note that if $d_1 = d_2$ but $\sin(a^{\pm}/2)$ is not 0, then the expression in Proposition 4.1 is 0, which agrees with the result in Lemma 4.1.

5 Sparse grid interpolation and cubature

As seen in Section 3, the primary step in computing the sensitivity values of Sobol′ is estimating an integral of the form

$$G = \int_A \Big(\int_B f(x,y)\, d\mu_B(y)\Big)^2 d\mu_A(x), \tag{5.1}$$

where

$$A = K^m, \quad B = K^{n-m}, \quad f : A \times B \to \mathbb{R}.$$

Here we assume f to be smooth.

In this section, we recall the basic ideas for approximating f on $[-1,1]^n$ using the sparse grid techniques of Smolyak, then apply this approximation to estimate G. See [1] for discussion of and references for sparse grid interpolation and cubature. As in [1], we use the extrema of the Chebyshev polynomials as nodes, with the coarsest level containing a single node and the remaining levels nested. For consistency with the literature, we largely adopt the indexing in [1] as well.

Let $M_1=1$, $M_i=2^{i-1}+1$ for $i>1$. Let $x_1^1=0$, and for $i>1$ and $1\le j\le M_i$, let

$$x_j^i=\cos\Big(\frac{(j-1)\pi}{(M_i-1)}\Big).$$

Also, with notation as in the previous section, let

$$L_j^i=L_{M_i-1,j-1} \quad \text{and} \quad w_j^i=w_{M_i-1,j-1}.$$

To describe the range of multi-indices corresponding to a given multi-index α with all entries positive, let

$$M(\alpha)=(M_{\alpha_1},\cdots,M_{\alpha_n}).$$

For $\mathbf{1}\le\beta\le M(\alpha)$, define

$$x_\beta^{\otimes\alpha}=(x_{\beta_1}^{\alpha_1},\cdots,x_{\beta_n}^{\alpha_n}),\quad w_\beta^\alpha=w_{\beta_1}^{\alpha_1}\cdots w_{\beta_n}^{\alpha_n},\quad L_\beta^\alpha=L_{\beta_1}^{\alpha_1}\cdots L_{\beta_n}^{\alpha_n}.$$

Note that

$$w_\beta^\alpha=\int_{K^n}L_\beta^\alpha d\mu.$$

For $n>1$, define

$$\mathcal{A}^{\otimes\alpha}(f)=\sum_{0\le\beta\le M(\alpha)}f(x_\beta^{\otimes\alpha})L_\beta^\alpha.$$

For an integer $q\ge n$, the Smolyak interpolation formula is then

$$\mathcal{A}(q,n)(f)=\sum_{q-n+1\le|\alpha|\le q}(-1)^{q-|\alpha|}\left(\begin{array}{c} n-1 \\ q-|\alpha| \end{array}\right)\mathcal{A}^{\otimes\alpha}(f). \tag{5.2}$$

Here $k=q-n$ describes the maximum degree of the interpolating polynomial used in a single variable; when α has all components but α_j equal to 1 and $\alpha_j=k+1$, then L_β^α has degree $M_{k+1}-1$ in x_j and is independent of the remaining variables. Moreover, it is known (see, e.g., [1]) that $A(n+k,n)(p)$ reproduces p exactly whenever p is a polynomial of degree k or less. By integrating this formula as was done to obtain (4.3), we obtain the standard Smolyak cubature formula to approximate the integral of f over K^n.

To describe the accuracy of this interpolation more generally, let $\Theta(q,n)$ be the set of all points of evaluation in (5.2), and let N denote the number of points in $\Theta(q,n)$. From [1], if k is fixed, n is large, and $q=n+k$, then

$$N\approx\frac{2^kn^k}{k!}.$$

Moreover, assuming that $D^\alpha f$ is continuous for all $|\alpha|\le\ell$, we may define

$$\|f\|=\max\left\{\|D^\alpha f\|_\infty:\ \alpha\in\mathbb{N}_0^n,\ \alpha_i\le\ell\right\}.$$

Then [1, Theorem 8] gives the error bound

$$\|A(q,n)(f)-f\|_0\le C_{n,\ell}N^{-\ell}(\log N)^{(n-1)(\ell+2)+1}\|f\|. \tag{5.3}$$

By integrating, this bound also gives an estimate for the error in approximating the integral of f.

6 Cubature for sensitivity values

There are several possible approaches to approximating the integral in (5.1). Here we describe two methods, each of which requires evaluation of f only on the set Θ, independent of the subsets of coordinates used to determine A and B. The first method applies the ideas used to develop integral approximations as in the previous two sections. That is, first replace the function, f, by an approximation using Lagrange interpolating polynomials, then integrate. This method is used to calculate the main effect coefficients, S_j. The second method is to use sparse-grid cubature directly on a higher dimensional space, but use projection and the symmetry of the support nodes to require evaluation of f on the original set Θ only. This method is used to calculate the total effect coefficients, S_{T_j}.

For clarity of notation, in this section, we drop $d\mu_A(x)$ in favor of dx, etc., with the assumption that the normalizing factors remain implicit.

Main effect values

For the main effect coefficients, S_j, we need to approximate (5.1) in the case when A is 1-dimensional. To do this, we first approximate f via $\mathcal{A}(q,n)$ on a Smolyak cubature set, Θ, to get an approximating function $\hat{f}$. Suppressing the precise bounds on α and β, we obtain

$$\hat{f}(x,y)=\sum_{\alpha,\beta}C_\alpha f(x_\beta^{\otimes\alpha})L_\beta^\alpha(x,y),$$

where $x\in A$, $y\in B$ as in (5.1) and C_α is the coefficient in (5.2). Note that a given point in Θ generally appears more than once in this sum due to the nonuniqueness of the representation $x_\beta^{\otimes\alpha}$. In practice, we collect terms associated with the same point, but for exposition we keep the form above. Replacing f by $\hat{f}$ and expanding the square in G, we get the approximation

$$\begin{aligned}\hat{G}&=\int_A\int_B\int_B\hat{f}(x,y)\hat{f}(x,z)dydzdx\\&=\sum_{\alpha,\beta}\sum_{\alpha',\beta'}C_\alpha C_{\alpha'}f(x_\beta^{\otimes\alpha})f(x_{\beta'}^{\otimes\alpha'})\int_{A\times B\times B}L_\beta^\alpha(x,y)L_{\beta'}^{\alpha'}(x,z)dydzdx.\end{aligned}$$

To compute the final integral, note that the product form of L^{α}_{β} means that this integral is actually a product of one dimensional integrals. For the integral over each coordinate in A, we obtain an integral of a product of two interpolating polynomials as in (4.6), which may be evaluated as in Corollary 4.1. For the integral over each coordinate in B, we obtain an integral of a single interpolating polynomial, which may be evaluated as in (4.4). In each case, this evaluation is linear in the degree of the polynomials. Moreover, each of these one dimensional integrals may be precomputed and then multiplied as needed to compute each multidimensional integral in the expression above.

To make this more explicit, let $\alpha(A)$ be the projection of α to the coordinates in A, and likewise for $\alpha(B)$. Also, let

$$(L^{\alpha}_{\beta},L^{\alpha'}_{\beta'})_A=\int_A L^{\alpha(A)}_{\beta(A)}(x)L^{\alpha'(A)}_{\beta'(A)}(x)dx,$$

and note that

$$w^{\alpha(B)}_{\beta(B)}=\int_B L^{\alpha(B)}_{\beta(B)}(y)dy.$$

Then

$$\hat{G}=\sum_{\alpha,\beta}\sum_{\alpha',\beta'}C_{\alpha}C_{\alpha'}f(x^{\otimes\alpha}_{\beta})f(x^{\otimes\alpha'}_{\beta'})w^{\alpha(B)}_{\beta(B)}w^{\alpha'(B)}_{\beta'(B)}(L^{\alpha}_{\beta},L^{\alpha'}_{\beta'})_A.$$

As noted above, A is one-dimensional when computing S_j; without loss, we may assume that A is the first coordinate direction. In this case, the expression above, which is a quadratic form, is nearly a diagonal quadratic form, except for the appearance of the inner product. In practice, for moderately large dimension, most entries in a multi-index α are trivial. Hence for computational efficiency, we may represent a point in Θ as the nontrivial entries in α and β together with a choice of coordinate direction for each nontrivial entry. With this representation, the coefficients in the expression above depend only on the nontrivial entries in α, α', β, and β' plus whether or not the first coordinate in α and α' is trivial or not. Keeping track of this accounting and summing up over all equivalent representations of points in Θ, we obtain a block representation of this quadratic form. That is, collecting terms corresponding to identical points in Θ produces

$$\hat{G}=v^TWv,$$

where v is a column vector obtained by evaluating f at the points of Θ. However, by using the combinatorial representation of points in Θ, we identify sets of pairs of points with a common value for the corresponding coefficient in the quadratic form. Reordering the rows and columns, we identify a block structure in W. Replacing W by the matrix $\hat{W}$ formed from the coefficients in this block and replacing v by $\hat{v}$ formed by summing the elements of v in each block, we obtain

$$\hat{G}=\hat{v}^T\hat{W}\hat{v}.$$

Computational efficiency

Note that, for fixed $n=\dim(A\times B)$, the algorithm requires that we evaluate f only on the points in Θ regardless of $m=\dim(A)$. Assuming an upper bound on the time for a single evaluation of f, the total time required for functional evaluation is bounded by a constant times the number of points in Θ. As mentioned near the end of Section 5, the number of points in Θ is $N\approx 2^k n^k/k!$, for fixed $k=q-n$ and large dimension n. Hence the time required for functional evaluation is

$$\mathcal{O}(N)=\mathcal{O}\Big(\frac{2^k n^k}{k!}\Big).$$

In practice, f may be expensive to compute, so this represents the bulk of the computational time. For completeness, we analyze the time required to compute all the S_j. Note that the weights w^{α}_{β} and integrals of interpolating functions may be precomputed in time $\mathcal{O}(d^2)$, where $d=2^k$ is the maximum degree of the interpolating polynomials L^i_j.

After collecting terms corresponding to multiple occurrences of a given point in Θ, the double sum for $\hat{G}$ may then be computed at first glance in time $\mathcal{O}(N^2)$. However, we may reduce this estimate using the block structure described above. For a given pair, α and β, define the corresponding minimal representative to be the nontrivial entries of α, with order preserved, and the corresponding entries in β. For a given norm $n<|\alpha|$, we need to determine the nontrivial entries in α. This can be done by distributing $t=|\alpha|-n$ balls into some number, s, of nonempty slots, where $1\le s\le n$, which can be done in $\binom{t-1}{s-1}$ ways. For a given such choice, the number in a given slot, t_j, gives $M_{t_j+1}=2^{t_j}+1$ corresponding points in that coordinate in the sparse grid. One of these points is 0, which may be ignored since it will have been counted for smaller values of t_j. This gives 2^{t_j} points, for each slot independently, for a total of 2^t points corresponding to this choice of α. This estimate is quite crude since it double counts many points, but it will suffice here. Let R be the total number of representatives. Summing over t and s gives

$$R\le\sum_{t=1}^{k}2^t\sum_{s=1}^{t}\binom{t-1}{s-1}.$$

Recognizing the inner sum as 2^{t-1} (the total number of subsets of $t-1$ elements) and using the partial geometric sum in the outer sum gives

$$R\le 2(4^k-1)/3.$$

Since the sum for $\hat{G}$ is different for multi-indices which are trivial in the first slot versus those that are nontrivial in the first slot, we see that $2R$ gives an upper bound for the number of rows in $\hat{W}$. Since we can compute $\hat{v}$ from the values of f on Θ in time $\mathcal{O}(N)$, we see that after precomputation of the weights and the function evaluations, we may compute $\hat{G}$ in time $\mathcal{O}(R^2)=\mathcal{O}(8^k)$, which is $\mathcal{O}(N)$ for fixed k and large n. Finally, the mean, f_0, and the variance, D, may be computed using the standard sparse grid cubature rule, which gives time $\mathcal{O}(N)$. Hence the total time for a single value S_1 is $\mathcal{O}(N)$. To obtain all the S_j, we create a cyclic permutation of the points in Θ by cyclically permuting the coordinate directions:

$$\sigma(x_1,\cdots,x_n)=(x_2,\cdots,x_n,x_1).$$

By the symmetry of the sparse grid, we have

$$\sigma(\Theta)=\Theta.$$

Computing $\hat{G}$ for $f\sigma$ allows us to compute S_2, and iterating gives $S_1,\cdots,S_n$. Hence we obtain

Theorem 6.1. *For a fixed degree of approximation k and large dimension n, the main effect sensitivity coefficients S_j, $j=1,\cdots,n$, for f may be computed in time*

$$\mathcal{O}(nN)=\mathcal{O}\Big(\frac{2^k n^{k+1}}{k!}\Big).$$

The function f need be evaluated only on points in the sparse grid Θ.

Total effect values

For S_{T_j}, the total effect values, the set A in (5.1) is $(n-1)$-dimensional, and the accounting as described above for $\hat{G}$ is much more involved since now the inner product $(L^{\alpha}_{\beta},L^{\alpha'}_{\beta'})_A$ depends on the overlap pattern of the nontrivial entries in the multi-indices. In contrast, B is only 1-dimensional in this case, so the integral

$$\int_A\int_B\int_B f(x,y)f(x,z)dydzdx \tag{6.1}$$

is only $(n+1)$-dimensional. Moreover, the symmetry of the nested sparse grid implies that if $(x,y,z)\in K^{n-1}\times K\times K$ is a point in the sparse grid of dimension $n+1$ and maximum norm $q+1$, then (x,y) and (x,z) are both points in the sparse grid of dimension n and maximum norm q; that is, they are both points in $\Theta(q,n)$. Hence we may approximate the integral (6.1) by first evaluating f on the points in Θ, then using appropriate projections from $\Theta(q+1,n+1)$ to $\Theta(q,n)$ to define $g(x,y,z)=f(x,y)f(x,z)$ evaluated on $\Theta(q+1,n+1)$. Given g, we may then approximate (6.1) using the standard sparse grid cubature weights. As before, f_0, D, and the evaluations of f may be done in time $\mathcal{O}(N)$. Also as before, for fixed $k=q-n$ and large n, the construction of g and the subsequent cubature may be done in time

$$\mathcal{O}\Big(\frac{2^k(n+1)^k}{k!}\Big)=\mathcal{O}(N).$$

Using cyclic permutations as before, we have the corresponding theorem for S_{T_j}.

Theorem 6.2. *For a fixed degree of approximation k and large dimension n, the total effect sensitivity coefficients S_{T_j}, $j=1,\cdots,n$, for f may be computed in time*

$$\mathcal{O}(nN)=\mathcal{O}\Big(\frac{2^k n^{k+1}}{k!}\Big).$$

The function f need be evaluated only on points in the sparse grid Θ.

Convergence

Note that (5.3) guarantees convergence of the computed values of (5.1) to the true values as $q\to\infty$ (hence $N\to\infty$) under the assumption that f is C^ℓ smooth, $\ell\geq 1$. For the method used to compute S_{T_j}, this follows directly by replacing $f(x,y)$ by $g(x,y,z)=f(x,y)f(x,z)$ in (5.3). For the method used to compute S_j, we need to show the convergence of

$$\|A(q+1,n+1)(f_1)A(q+1,n+1)(f_2)-f_1f_2\|\to 0 \quad \text{as } q\to\infty,$$

where

$$f_1(x,y,z)=f(x,y), \quad f_2(x,y,z)=f(x,z).$$

But this follows immediately by adding and subtracting $A(q+1,n+1)(f_1)f_2$, using the triangle inequality, then using (5.3) together with the sup norm bound on f.

Since the calculation of S_j and S_{T_j} requires dividing (5.1) by D, the error in these values may be sensitive to the details of the function, particularly for functions with small variance. As implemented below, D is calculated by using the values of f on $\Theta(q,n)$ to determine the values of $f-E[f]$, then squaring these values and using the cubature rule for Θ to approximate the integral. An alternative method, which is generally more accurate, is to use the values of f on $\Theta(q,n)$ to determine the interpolating polynomial, then use the values of this polynomial on $\Theta(q+1,n)$ to determine D. This method requires some extra computational effort, but does not require additional function evaluations.

However, in practice, only the magnitude of one sensitivity in comparison to the others is used to guide exploration of parameter space. Since all the values S_j and S_{T_j} include the same factor $1/D$, we may directly compare the integrals used to determine the sensitivity values, in which case the convergence is given by (5.3).

7 Numerical results

In order to evaluate this method, we computed S_j and S_{T_j} for a variety of standard test functions. As noted above and in [1], the convergence for sparse grid cubature using Clenshaw-Curtis nodes depends strongly on the differentiability of the function being integrated. For discontinuous functions or functions whose first derivatives are discontinuous, the estimates from sparse grid cubature are poor to nearly useless. At the opposite extreme, for low degree polynomials, sparse grid cubature will give answers which are correct to round-off error for correspondingly low values of q. Since the output of many models of interest for sensitivity analysis lies between these two extremes, we have chosen a set of test functions that lie between these extremes. The test functions are the functions 1 through 4 of [1]:

1. OSCILLATORY: $f_1(x)=\cos\Big(2\pi w_1+\sum_{i=1}^{n} c_i x_i\Big),$
2. PRODUCT PEAK: $f_2(x)=\prod_{i=1}^{n}\big(c_i^{-2}+(x_i-w_i)^2\big)^{-1},$
3. CORNER PEAK: $f_3(x)=\Big(1+\sum_{i=1}^{n} c_i x_i\Big)^{-(n+1)},$
4. GAUSSIAN: $f_4(x)=\exp\Big(-\sum_{i=1}^{n} c_i^2 (x_i-w_i)^2\Big).$

We used $n=10$ and chose values for c_i and w_i at random as indicated in [1].

We compared our method to the Extended FAST method of [6] as implemented in the software package Simlab 3.2.6 [9] and to the quasi-Monte Carlo integration method labeled the ″Richtmyer sequence″ in [12]. To apply quasi-Monte Carlo, we expanded the integral as in Eq. (6.1) and partitioned the points so that the total number of points used to calculate all of the main effect values was roughly the same as the number of points used to calculate all of these values using the sparse grid method, and likewise for the total effect values; some version of this method is often called ″Sobol's method″ [10]. We show a representative sample of the computed values plotted as functions of the number of points evaluated in Figs. 1 (main effect value) and 3 (total effect values). Each line corresponds to the estimated sensitivity value for one coordinate direction plotted against the number of function evaluations. The coordinate direction for a particular line is indicated in the legend in the first box of Figs. 1 and 3. All plots in these figures use the same coordinate directions as indicated in the legend. All computations were performed on a 3.2 GHz Pentium 4 with 2 GB of RAM.

Note that the sparse grid method and to a lesser extent the Extended FAST method provide relatively consistent size and order relationships among the various factors, even for small numbers of model evaluations (this is true even when the additional coordinate directions are added). That is, a common application for sensitivity analysis is to determine which parameters are the most sensitive. These results show that sparse grid and Extended FAST generally provide reasonable answers even when the number of model evaluations is relatively low. However, the implementation of Extended FAST in [9] puts a lower bound (depending on the dimension) on the number of points that must be evaluated. This is the reason for the gaps in the graphs for Extended FAST.

Where possible, we also computed main effect and total effect sensitivity values analytically (using the Integrate function of Mathematica with numerical evaluation). For these cases, we show the average differences (taken over each of the 10 coordinate directions) between the approximated and exact values for both main and total effect values,

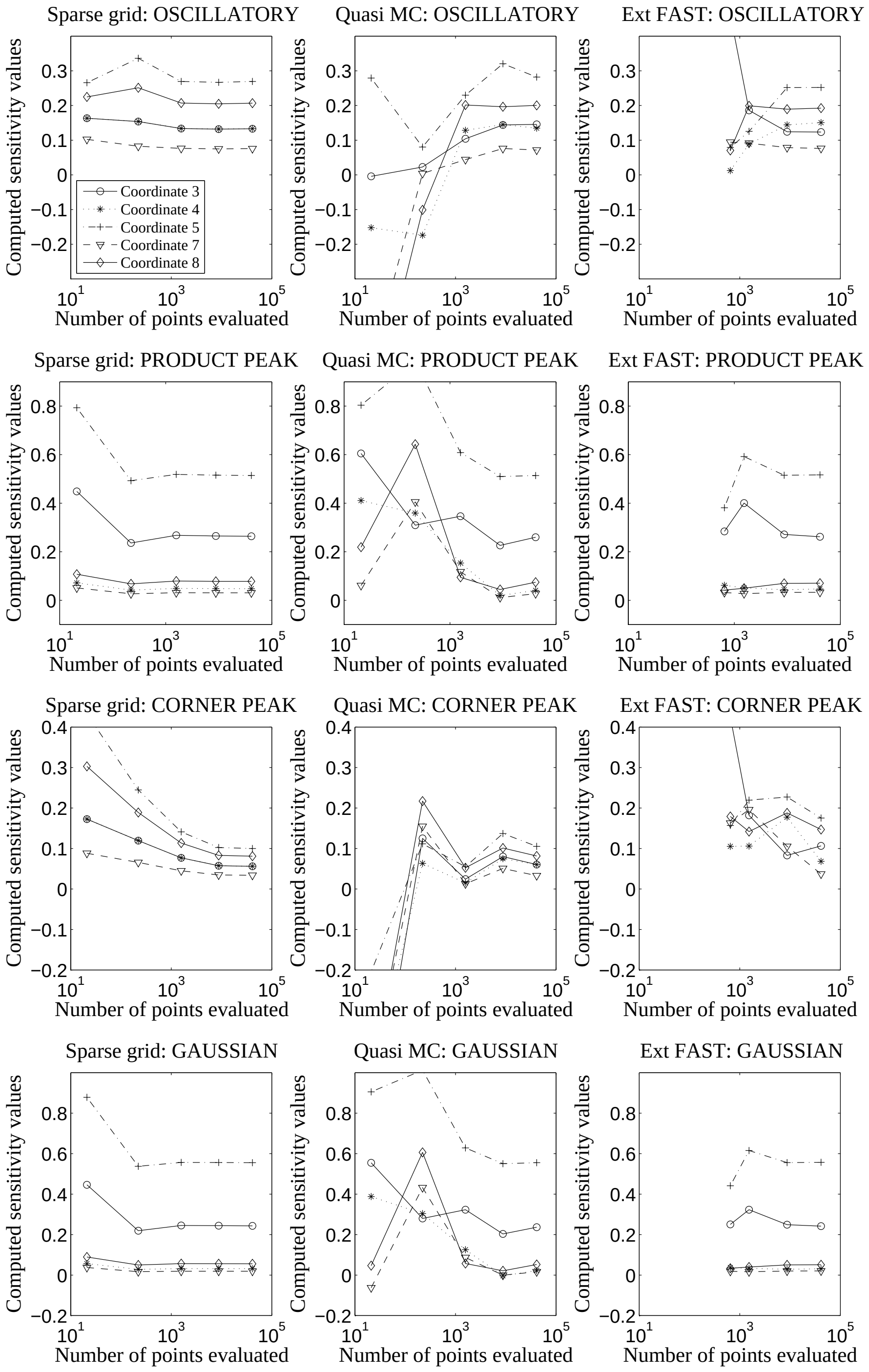

Figure 1: Computed main effect values plotted against number of points evaluated. Left: Sparse grid method. Middle: Quasi-Monte Carlo method. Right: Extended FAST method. The sparse grid method generally shows the correct ordering of the sensitivity values, even for a small number of evaluations.

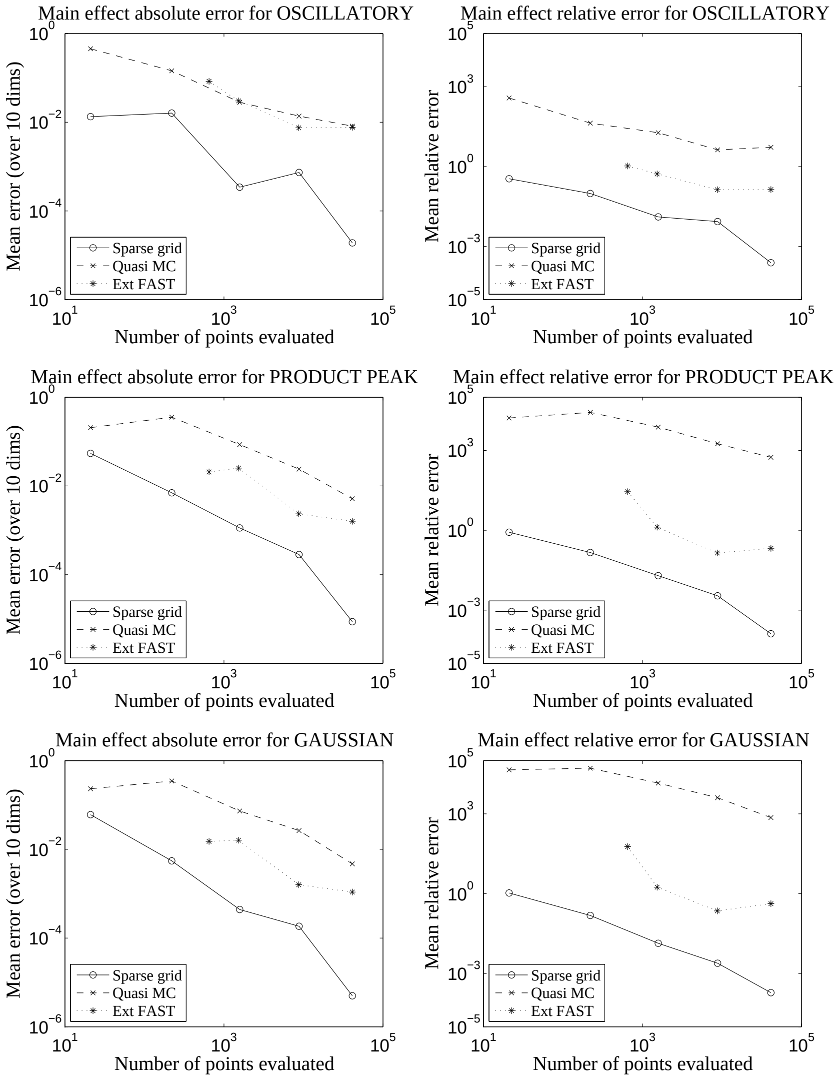

Figure 2: Average differences (averaged over all 10 coordinate dimensions) between approximated and true main effect values. Left: Absolute errors. Right: Relative errors.

plotted as functions of the number of points evaluated. We also show the average relative errors, $|S-\hat{S}|/S$, where S is the exact sensitivity value and $\hat{S}$ is the computed value. These results are shown in Figs. 2 and 4. Note that for a given value S_j, the plot of the relative error is simply a rescaled version of the plot of the absolute error. However, in averaging over all 10 coordinate directions, a small S_j may yield a fairly large relative error even with a small absolute error, thus resulting in a significantly different contribution to the average relative error versus the average absolute error. For the function Corner Peak, we were not able to compute analytic values in a reasonable amount of time, so we show only the values obtained rather than error values.

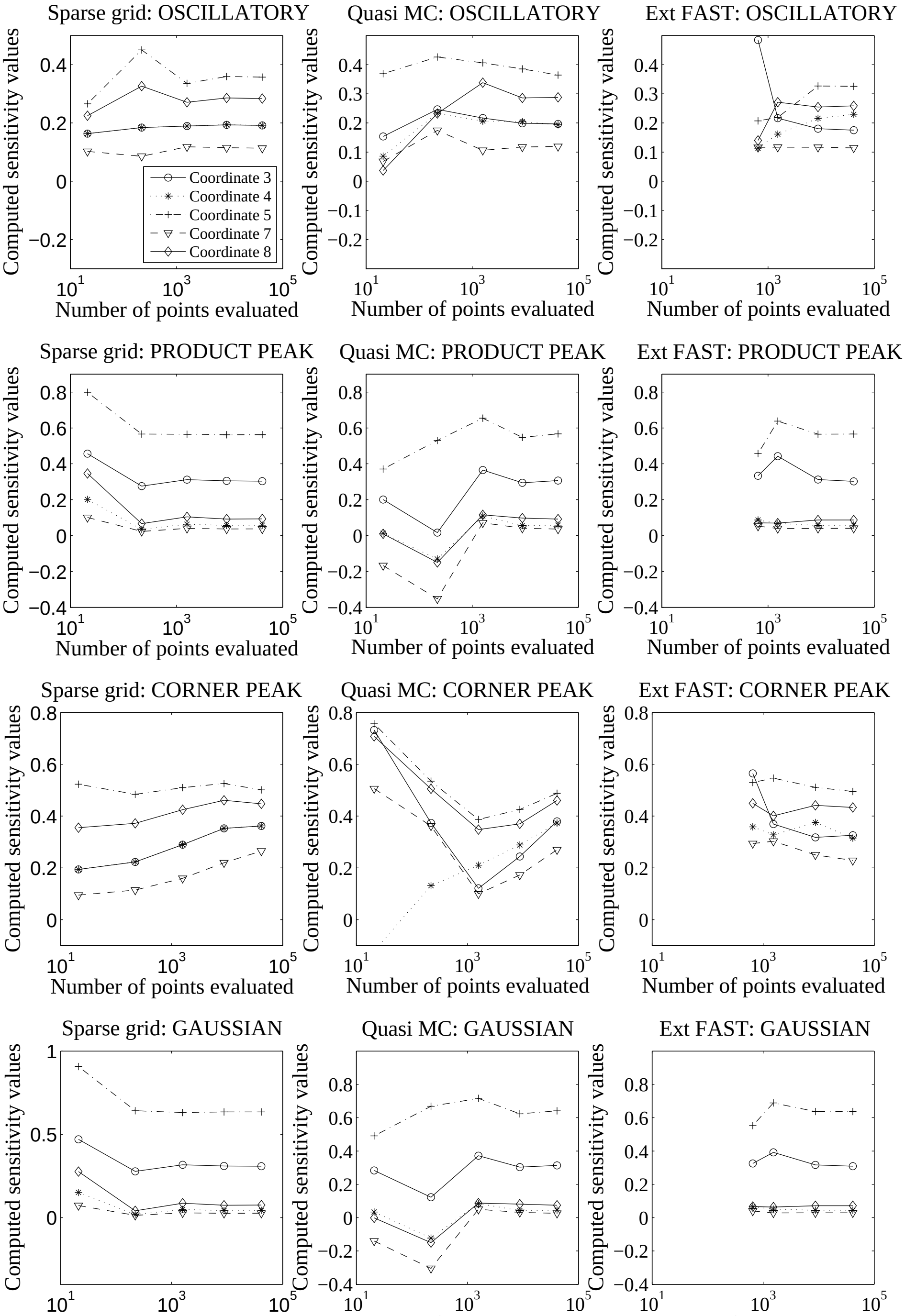

Figure 3: Computed total effect values plotted against number of points evaluated. Left: Sparse grid method. Middle: Quasi-Monte Carlo method. Right: Extended FAST method. The sparse grid method generally shows the correct ordering of the sensitivity values, even for a small number of evaluations.

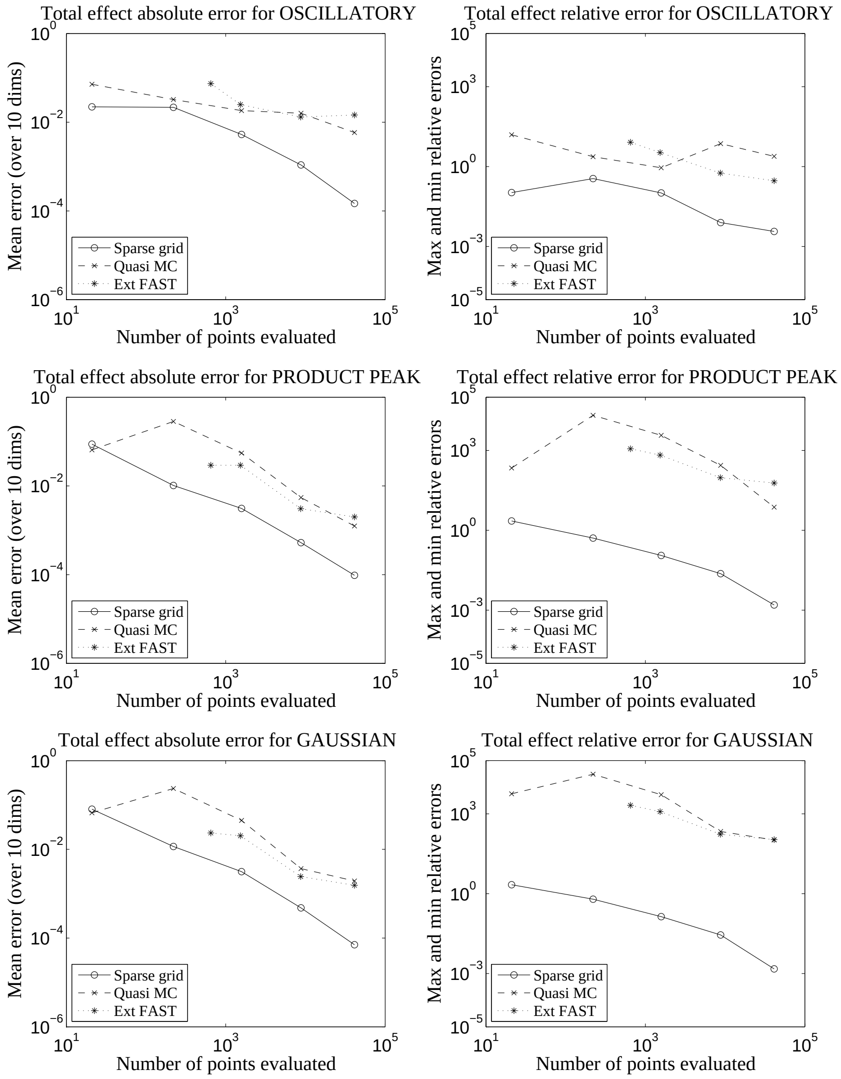

Figure 4: Average differences (averaged over all 10 coordinate dimensions) between approximated and true total effect values. Left: Absolute errors. Right: Relative errors.

The error plots show that the sparse grid has good accuracy, even for small numbers of function evaluations, and good convergence as the number of function evaluations increases. The quasi-Monte Carlo method is generally less accurate but still has reasonably good convergence. The Extended FAST method is often reasonably accurate but doesn't display much improvement with the number of function evaluations. It may be that different choices in the particulars of the Extended FAST algorithm would produce better convergence; we have not tried to address this question since it represents a completely different line of inquiry.

For each of the four functions, the sparse grid method shows the best overall accuracy and convergence rates. This accuracy is especially pronounced for the relative error with a small number of function evaluations, which is in turn reflected in the plots of the sensitivity values themselves; the sparse grid values are generally in the correct order and the relative magnitudes $S_j/S_{\max}$ are close to the true values even for a small number of function evaluations, where $S_{\max}$ is the largest sensitivity value. We note here that the first two moments, f_0 and D, display similar convergence rates, although the convergence of D is slower than that of f_0 due to the fact that f is more accurately approximated by polynomials of a fixed degree than is f^2.

8 Conclusions

We have used sparse grid interpolation and cubature to produce a numerically accurate and efficient method for computing the main and total effect variance-based global sensitivity coefficients. This method displays good accuracy and convergence properties on functions which are known to be globally differentiable, compares favorably with existing methods, and allows for the computation of all sensitivity values from the evaluation of a function on a single set of sparse grids. In practice, this method allows one to conduct high-order sensitivity analysis as a postprocessing step of a sparse-grid based stochastic collocation simulation without extensive additional computational efforts. Alternatively, this method could be used in a preprocessing step since it provides a reasonable estimate of the relative sensitivity of the output to each coordinate with a small number of function evaluations. This information could be used to fix the values of relatively insensitive coordinates and focus more resources on the remaining coordinates.

Acknowledgments

We are grateful to Maia Donahue and Ann Rundell for help in applying the Extended FAST method to our test functions and to the referees for several helpful suggestions.

References

[1] V. Barthelmann, E. Novak and K. Ritter, High dimensional polynomial interpolation on sparse grids, Adv. Comp. Math., 12 (2000), 273–288.

[2] T. Crestaux, O. Le Matre and J. M. Martinez, Polynomial chaos expansion for sensitivity analysis, Reliab. Eng. Syst. Safe., 94 (2009), 1161–1172.

[3] T. Gerstner and M. Griebel, Numerical integration using sparse grids, Numer. Algorithms., 18 (1998), 209–232.

[4] D. Lewandowski, R. M. Cooke and R. J. D. Tebbens, Sample-based estimation of correlation ratio with polynomial approximation, ACM Trans. Model. Comp. Simul., 18 (2007), 1–17.

[5] J. C. Mason and D. C. Handscomb, Chebyshev Polynomials, Chapman and Hall/CRC, Boca Raton, 2003.

[6] A. Saltelli, S. Tarantola and K. P. S. Chan, A quantitative model-independent method for global sensitivity analysis of model output, Technometrics., 41 (1999), 39–54.

[7] A. Saltelli, Making best use of model evaluations to compute sensitivity indices, Comp. Phys. Comm., 145 (2002), 280–297.

[8] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana and S. Tarantola, Global Sensitivity Analysis, The Primer, John Wiley & Sons, Ltd, 2008.

[9] Simlab 3.2.6, `http://simlab.jrc.ec.europa.eu/`, accessed 7/21/2009.

[10] I. M. Sobol', Sensitivity estimates for nonlinear mathematical models, Math. Model. Comp. Exp., 1 (1993), 407–414, [Translation of Sensitivity estimates for nonlinear mathematical models, Matematicheskoe Modelirovanie, 2 (1990), 112–118 (in Russian)].

[11] I. M. Sobol' and E. E. Myshetskaya, Monte Carlo estimators for small sensitivity indices, Monte. Carlo. Methods. Appl., 13 (2007), 455–465.

[12] G. Takhtamysheva, B. Vandewoestyneb and R. Cools, Quasi-random integration in high dimensions, Math. Comput. Simulat., 73 (2007), 309–319.

[13] L. N. Trefethen, Is Gauss quadrature better than Clensh-Curtis?, SIAM Rev., 50 (2008), 67–87.

[14] E. W. Weisstein, Christoffel-Darboux Identity, MathWorld–A Wolfram Web Resource, `http://mathworld.wolfram.com/Christoffel-DarbouxIdentity.html`.

[15] D. Xiu, Fast numerical methods for stochastic computations: A review, Commun. Comput. Phys., 5 (2009), 242–272.

[16] D. Xiu and J. S. Hesthaven, High-order collocation methods for differential equations with random inputs, SIAM J. Sci. Comput., 27 (2005), 1118–1139.