

## Research Article

# Efficient Basis Change for Sparse-Grid Interpolating Polynomials with Application to T-Cell Sensitivity Analysis

**Gregery T. Buzzard**

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

Correspondence should be addressed to Gregery T. Buzzard; [buzzard@math.purdue.edu](mailto:buzzard@math.purdue.edu)

Received 28 November 2012; Accepted 18 March 2013

Academic Editor: Željko Bajzer

Copyright © 2013 Gregery T. Buzzard. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sparse-grid interpolation provides good approximations to smooth functions in high dimensions based on relatively few function evaluations, but in standard form it is expressed in Lagrange polynomials. Here, we give a block-diagonal factorization of the basis-change matrix to give an efficient conversion of a sparse-grid interpolant to a tensored orthogonal polynomial (or gPC) representation. We describe how to use this representation to give an efficient method for estimating Sobol' sensitivity coefficients and apply this method to analyze and efficiently approximate a complex model of T-cell signaling events.

## 1. Introduction

A common problem in many areas of computational mathematics is to approximate a given function based on a small number of functional evaluations or observations. This problem arises in numerical methods for PDE [1, 2], sensitivity analysis [3–5], uncertainty quantification [2], many areas of modeling [6, 7], and other settings. As a result, there are a large number of approaches to this problem, and the literature is large and growing quickly. In settings in which the points of evaluation may be chosen at will, two common approaches are generalized polynomial chaos (gPC) using cubature and sparse grid collocation [8]. In other settings in which the points of evaluation are given, common approaches include RS-HDMR, cut-HDMR, ANOVA decomposition, kriging, and moving least squares; see, for example, [9, Chapter 5] for a discussion of such methods and further references.

Sparse grid collocation has been used widely in recent years as a means of providing a reasonable approximation to a smooth function,  $f$ , defined on a hypercube in  $\mathbb{R}^n$ , based on relatively few function evaluations [2]. This method produces a polynomial interpolant using Lagrange interpolating polynomials based on function values at points in a union of product grids of small dimension [10, 11]. Using barycentric interpolation to evaluate the resulting polynomial [12], this method is a viable alternative to an expansion of  $f$  in

terms of a sum of products of one-dimensional orthogonal polynomials. This latter approach is known as generalized polynomial chaos (gPC) or spectral decomposition and is obtained via standard weighted  $L^2$  techniques. However, the orthogonality implicit in the gPC representation often provides many advantages over the Lagrange representation, particularly in applications to differential equations, in which the gPC representation is closely related to spectral methods. Other advantages of the gPC representation include the ability to estimate convergence as more points are added to the sparse grid, and the ability to estimate variance-based sensitivity coefficients quickly and accurately [4]. A common approach to obtain the gPC coefficients is to use numerical integration by applying a cubature rule. However, cubature rules to integrate the product of two polynomials up to the degree in the sparse grid interpolant typically require either more or different points than found in the sparse grid itself.

In this paper, we provide an efficient algorithm for converting from the Lagrange interpolating polynomial to an equivalent gPC polynomial using only the function values at the sparse grid points. The foundation of this algorithm is a matrix factorization based on the fact that the sparse grid is a union of small product grids. More precisely, let  $\Phi$  be the matrix obtained by evaluating each of the gPC basis functions (one per column) at each of the sparse grid points (one per row). This matrix produces the gPC coefficients,

$c$ , from the function values,  $f$ , by solving  $\Phi c = f$  for  $c$ . We show below that  $\Phi^{-1}$  factors into a product of block diagonal matrices in which each block corresponds to one of the small product grids composing the full sparse grid. We also demonstrate the utility of this method for calculating variance-based sensitivity coefficients on a model of T-cell signaling.

Methods for changing basis between sets of orthogonal polynomials are described in many places, including [13–15]. Most of these results focus on the case of polynomials of one variable, and none focuses on the case of polynomial bases for sparse grid interpolants. For further background, Boyd [16, Chapter 5] discusses the matrix multiplication transform for converting between Lagrange and spectral representations, while Chapter 10 discusses the special case of the Chebyshev polynomial expansion using the FFT.

*Organization.* In Section 2, we provide some background into Smolyak’s algorithm for sparse grid interpolation and the calculation of Sobol’ sensitivity coefficients. In Section 3.1, we give the factorization of  $\Phi^{-1}$  and in Section 3.2 we describe the algorithm to produce the gPC representation based on function values at the sparse grid points and give an upper bound on the complexity of the algorithm. In Section 4.1, we give numerical results on running time and accuracy of the algorithm for a simple test function. In Section 4.2, we describe the application of this algorithm to compute Sobol’ sensitivity coefficients for a model of T-cell signaling.

## 2. Background

*2.1. Sparse Grid Interpolation.* In this section, we provide some background on Smolyak’s algorithm for sparse grid interpolation. The discussion here is based on [10]. The foundation for sparse grid interpolation is interpolation in one dimension using Lagrange interpolation as follows:

$$U^i(f) = \sum_{j=1}^{m_i} f(x_j^i) L_j^i, \quad (1)$$

where  $i \in \mathbb{N}$ ,  $x_j^i \in [-1, 1]$ , and  $L_j^i$  is the Lagrange polynomial satisfying  $L_j^i(x_k^i) = \delta_{jk}$ . A common choice for sparse grid interpolation is to use the Chebyshev-Gauss-Lobatto (CGL) points, in which case  $m_i = 2^{i-1} + 1$  and  $x_j^i = -\cos((j-1)\pi/2^{i-1})$  for  $i > 1$ , since this choice provides a nesting property that is vital for efficient interpolation in higher dimensions. Moreover,  $m_1 = 1$  and  $x_1^1 = 0$  for this choice. Other choices, including Gauss-Patterson nodes and nodes on a (semi)infinite interval as in [17], are also possible; the methods below apply for these choices as well. These one-dimensional formulas may be tensored in dimension  $d > 1$  to yield

$$U^{\mathbf{i}}(f) = (U^{i_1} \otimes \cdots \otimes U^{i_d})(f) = \sum_{\mathbf{1} \leq \mathbf{j} \leq \mathbf{m}_{\mathbf{i}}} f(x_{\mathbf{j}}^{\mathbf{i}}) L_{\mathbf{j}}^{\mathbf{i}}, \quad (2)$$

where  $\mathbf{i}$  and  $\mathbf{j}$  are multi-indices with componentwise partial order,  $\mathbf{1}$  is the multi-index of all 1s,  $x_{\mathbf{j}}^{\mathbf{i}} = (x_{j_1}^{i_1}, \dots, x_{j_d}^{i_d})$ , and

$L_{\mathbf{j}}^{\mathbf{i}}(x) = L_{j_1}^{i_1}(x_1) \cdots L_{j_d}^{i_d}(x_d)$ . This formula requires  $m_{i_1} \cdots m_{i_d}$  function values sampled on a product grid. Note, however, that when some of the  $i_k$  are 1, then this grid is of dimension less than  $d$  (since  $m_1 = 1$ ).

Linear combinations of these formulas produce the Smolyak formulas. Let  $U^0 = 0$  and  $\Delta^i = U^i - U^{i-1}$  for  $i \in \mathbb{N}$ , and define  $|\mathbf{i}| = i_1 + \cdots + i_d$ . Then for  $q \geq d$ , we have

$$A(q, d) = \sum_{d \leq |\mathbf{i}| \leq q} \Delta^{\mathbf{i}}, \quad (3)$$

where  $\Delta^{\mathbf{i}}$  is the tensor product of the  $\Delta^{i_k}$ . A multinomial expansion and  $q = d + k$  produce

$$A(d+k, d) = \sum_{k+1 \leq |\mathbf{i}| \leq d+k} (-1)^{d+k-|\mathbf{i}|} \binom{d-1}{|\mathbf{i}|-k-1} U^{\mathbf{i}}. \quad (4)$$

This formula is known as the combination technique, which first appeared in connection with sparse grids in [18]. When  $d > k$ , as is common for large  $d$ , we can replace  $k+1 \leq |\mathbf{i}|$  by  $d \leq |\mathbf{i}|$ . An anisotropic version of this formula is described in [19]. The formula for this version is largely the same as (3) with the index set  $d \leq |\mathbf{i}| \leq q$  replaced by a more general index set  $I$ , characterized by the property that if  $\mathbf{i} \in I$  and  $i_k > 1$ , then  $\mathbf{i} - \mathbf{e}_k \in I$ , where  $\mathbf{e}_k$  has 1 in the  $k$ th position and zeros elsewhere.

Important results from [10] include that  $A(d+k, d)(P) = P$  for all polynomials of degree at most  $k$ , and that if the derivatives  $D^{\alpha} f$  are continuous for all multi-indices  $\alpha$  with  $\alpha_i \leq m$  for all  $i$ , then

$$\|f - A(q, d)(f)\|_{\infty} \leq \frac{c_{d,m}}{n^m} (\log n)^{(m+2)(d-1)+1} \|f\|_{\infty}, \quad (5)$$

where  $n = n(q, d)$  is the number of points in the sparse grid,  $X(q, d)$ , for  $A(q, d)$ . There is a similar error estimate for a weighted  $L^2$  norm of the Chebyshev expansion of  $f - A(q, d)(f)$ . They note also that  $n(d+k, d)$  is asymptotic to  $2^k d^k / k!$  as  $d$  tends to  $\infty$ , in the sense that the ratio tends to 1. More precise estimates on convergence may be found in [20].

*2.2. Global Sensitivity Analysis.* A common problem in analyzing complex biological models with unknown parameters is identifying which parameters play an important role in modulating the response of the model. The variance-based sensitivity coefficients of Sobol’ [21] can provide a great deal of insight into this problem.

The calculation of these coefficients for a function  $f(x) = f(x_1, \dots, x_n)$  defined on a hypercube in  $\mathbb{R}^n$  starts by writing  $f$  as a normalized sum of functions that depend on a specified subset of the following variables:

$$\begin{aligned} f(x) = & f_0 + \sum_{i=1}^n f_i(x_i) \\ & + \sum_{1 \leq i < j \leq n} f_{ij}(x_i, x_j) + \cdots + f_{12 \dots n}(x_1, \dots, x_n), \end{aligned} \quad (6)$$

with a weighted orthogonality condition imposed on pairs of functions in this decomposition and a 0 mean condition in each variable separately imposed on each function individually. Depending on the context, this decomposition is known as the Sobol' or ANOVA decomposition of  $f$  or the HDMR representation of  $f$ . One method for obtaining this decomposition is by calculating the gPC expansion of  $f$  [4, 7]. In fact, the gPC expansion is essentially a refinement of the above decomposition in which each component function in the sum above is written as a sum of a product of one-dimensional orthogonal polynomials.

The Sobol' or variance-based sensitivity coefficients for  $f$  are then defined in terms of the variance of the functions in the Sobol' decomposition. More precisely, let  $H^d$  denote the hypercube  $[-1, 1]^d$  and assume that the decomposition above takes place on this set. Let  $V_d = 2^d$  be the volume of  $H_d$ , and for a function  $g$  on this set, define the variance of  $g$  to be

$$D(g) = \frac{1}{V_d} \int_{H^d} g^2(x) dx - \left( \frac{1}{V_d} \int_{H^d} g(x) dx \right)^2. \quad (7)$$

In terms of the Sobol' decomposition given above, the main effect sensitivity coefficient for  $x_j$  is  $S_j = D(f_j)/D(f)$ . This is generalized to interaction coefficients,  $S_M$  by replacing  $f_j$  by  $f_M$ , where  $M$  is any subset of  $\{1, \dots, n\}$ . Also, the total effect sensitivity coefficient  $S_{T_j}$  is the sum of  $S_M$  over all  $M$  containing  $j$ .

As shown in [4], the gPC expansion of  $f$  leads to an efficient method for calculating these sensitivity coefficients:  $D(f_M)$  is simply the sum of squares of coefficients of the gPC polynomials in the expansion of  $f_M$ . Hence, the change of basis algorithm described above provides an efficient way to estimate Sobol' sensitivity coefficients based on function values at sparse grid nodes. We simply evaluate the function at sparse grid nodes, change to a gPC basis using Legendre polynomials, and find the appropriate sum of squares of coefficients to calculate the  $S_j$  and  $S_{T_j}$ .

### 3. Main Results

**3.1. Change of Basis Matrix and Factorization.** In this section, we describe the matrix factorization that is central to the efficient change to a gPC basis.

Let  $P_j$ ,  $j \geq 0$  be a set of polynomials, each of degree  $j$ , orthogonal on the interval  $[-1, 1]$  using inner product  $\int_{-1}^1 fgw dx$ , where  $w$  is a nonnegative weight function that is positive in the interior of the interval (straightforward modifications allow for unbounded intervals, but for clarity we restrict to the finite interval).

The expansion in (4) together with (1) gives a decomposition of  $A(q, d)$  into a sum of products of Lagrange polynomials, with each  $U^i$  corresponding to a full product grid,  $X_i$  (although usually  $X_i$  is less than  $d$ -dimensional since each entry of  $\mathbf{i}$ , that is 1, produces only a single point for the corresponding dimension). On this grid, each nontrivial

coordinate direction  $k$  has  $i = i_k > 1$ , and the number of points in this direction is  $m_i = 2^{i-1} + 1$ . Hence, the degree of each  $L_j^i$  is  $m_i - 1$ , so we write each  $L_j^i$  as a linear combination of  $P_0, \dots, P_{m_i-1}$ . For a fixed value of  $i$ , we have

$$L_j^i = \sum_{m=0}^{m_i-1} C_{j,m}^i P_m. \quad (8)$$

Taking tensor products as in (1), we obtain a basis in terms of tensor products of  $P_n$  for interpolation on the product grid  $X_i$ . Taking a union over the  $X_i$  that appear implicitly in (4) via  $U^i$ , we obtain a basis for interpolation on the entire sparse grid. The problem then is to determine the (gPC) coefficients in this basis for specified function values,  $\mathbf{f}$ , at the sparse grid nodes.

Conceptually, the simplest approach to finding the gPC coefficients starts by evaluating each basis function,  $P$ , at each point in  $X = X(q, d)$ . These values form a matrix,  $\Phi$ , with each basis function corresponding to one column and each point in  $X$  corresponding to one row.

*Definition 1.* Let  $P_j$ ,  $j = 1, \dots, n$  be the gPC basis polynomials for the sparse grid,  $X$ , having points  $x_1, \dots, x_n$ . Then the matrix  $\Phi$  is the matrix with entries  $\Phi_{i,j} = P_j(x_i)$ .

With this definition, and a gPC coefficient vector,  $c$ , the product  $\Phi c$  gives the value at each point in  $X$  of the polynomial determined by the coefficients in  $c$ . Hence, one can solve the interpolation problem by solving  $\Phi c = \mathbf{f}$  for  $c$ . Of course, simply constructing this matrix takes  $O(n^2)$  operations, while solving the system in this form using elimination takes  $O(n^3)$  operations.

The alternative presented here is to give a factorization of  $\Phi^{-1}$  based on (4).

**Theorem 2.** *The change-of-basis matrix  $\Phi^{-1}$  has a factorization*

$$\Phi^{-1} = \pi^T \widehat{D}_w \widehat{\Phi}^{-1} \pi, \quad (9)$$

where  $\widehat{\Phi}$  is block-diagonal, with each block corresponding to a subgrid,  $X_i$ ;  $\widehat{D}_w$  is diagonal;  $\pi$  is a matrix that includes  $\mathbb{R}^n$  into  $\mathbb{R}^N$  for some  $N > n$ .

*Proof.* To start, we order the columns of  $\Phi$  so that for each multi-index,  $\mathbf{i}$ , the set of column indices of basis functions for interpolation on  $X_i$  from (8) are the same as the set of row indices for points in  $X_i$ . Let  $n$  be the number of points in  $X$  and  $n_i$  the number in  $X_i$ . Let  $\pi_i$  be the projection from  $\mathbb{R}^n$  to  $\mathbb{R}^{n_i}$  obtained by restricting to the indices corresponding to  $X_i$ . In this case,  $\Phi_i = \pi_i \Phi \pi_i^T$  is the matrix obtained by evaluating basis functions at points, with the basis functions and points restricted to those associated with  $X_i$ . Hence, given  $\mathbf{f} = f(X)$ , we can interpolate  $f$  on  $X_i$  by solving  $\Phi_i c_i = \pi_i \mathbf{f}$  for  $c_i$ .

Since  $U^i(f)$  in (1) also interpolates  $f$  on  $X_i$ , we have

$$U^i(f)(X_i) = \Phi_i c_i = \pi_i \Phi \pi_i^T c_i. \quad (10)$$

Since the gPC basis functions associated with  $X_{\mathbf{i}}$  were obtained by changing basis from the Lagrange representation of  $U^{\mathbf{i}}$ , we see that  $U^{\mathbf{i}}(f)$  and the polynomial with gPC coefficients  $\pi_{\mathbf{i}}^T \mathbf{c}_{\mathbf{i}}$  are the same polynomial. Hence, we may extend the equality in (10) to all of  $X$  by replacing  $X_{\mathbf{i}}$  by  $X$  on the left and dropping  $\pi_{\mathbf{i}}$  on the right to obtain

$$U^{\mathbf{i}}(f)(X) = \Phi \pi_{\mathbf{i}}^T \mathbf{c}_{\mathbf{i}}. \quad (11)$$

Let  $w_{\mathbf{i}}$  denote the weight for  $U^{\mathbf{i}}$  in (4). Using (11) with (4), we have

$$\begin{aligned} \mathbf{f} &= A(f)(X) = \sum_{\mathbf{i} \in I} w_{\mathbf{i}} U^{\mathbf{i}}(f)(X) \\ &= \sum_{\mathbf{i}} w_{\mathbf{i}} \Phi \pi_{\mathbf{i}}^T \mathbf{c}_{\mathbf{i}} \\ &= \Phi \left( \sum_{\mathbf{i}} w_{\mathbf{i}} \pi_{\mathbf{i}}^T \mathbf{c}_{\mathbf{i}} \right). \end{aligned} \quad (12)$$

Recalling that  $\mathbf{c}_{\mathbf{i}} = \Phi_{\mathbf{i}}^{-1} \pi_{\mathbf{i}} \mathbf{f}$ , we have

$$\Phi^{-1} \mathbf{f} = \sum_{\mathbf{i} \in I} w_{\mathbf{i}} \pi_{\mathbf{i}}^T \Phi_{\mathbf{i}}^{-1} \pi_{\mathbf{i}} \mathbf{f}. \quad (13)$$

Let  $\mathbf{i}_1, \dots, \mathbf{i}_m$  be an enumeration of the indices in  $I$ , let  $\pi$  be the matrix obtained by vertical concatenation of  $\pi_{\mathbf{i}_1}, \dots, \pi_{\mathbf{i}_m}$ , let  $\widehat{\Phi}$  be the block diagonal matrix with  $\Phi_{\mathbf{i}_1}, \dots, \Phi_{\mathbf{i}_m}$  as blocks, and let  $\widehat{D}_w$  be the diagonal matrix with diagonal entries  $w_{\mathbf{i}_1}, \dots, w_{\mathbf{i}_m}$ , each repeated according to the size of the corresponding block. Then (13) gives the factorization

$$\Phi^{-1} = \pi^T \widehat{D}_w \widehat{\Phi}^{-1} \pi. \quad (14)$$

□

**3.2. Algorithm and Complexity.** Based on the previous section, the algorithm to find the coefficient vector,  $\mathbf{c}$ , to represent the sparse-grid interpolating polynomial in gPC basis is simply to use (14) to find  $\mathbf{c} = \Phi^{-1} \mathbf{f}$ . We avoid the matrix inverse by defining  $\widehat{\mathbf{f}} = \pi \mathbf{f}$ , solving  $\widehat{\Phi} \widehat{\mathbf{c}} = \widehat{\mathbf{f}}$  for  $\widehat{\mathbf{c}}$ , and then using  $\mathbf{c} = \pi^T \widehat{D}_w \widehat{\mathbf{c}}$ .

In this section, we show that for fixed  $k$  in (4) and large dimension,  $d$ , the complexity of this algorithm is linear in the number of points of evaluation. We frame this result as a corollary to the following proposition, which gives a bound on the total number of operations needed to compute on each subgrid,  $X_{\mathbf{i}}$ , in turn, as needed to solve  $\widehat{\Phi} \widehat{\mathbf{c}} = \widehat{\mathbf{f}}$  for  $\widehat{\mathbf{c}}$ .

In both of the following results,  $n(d+k, d)$  is the number of points in the sparse grid associated with  $A(d+k, d)$ .

**Proposition 3.** *Let  $m \geq 1$  and  $k \geq 1$  be fixed. Then there is  $c_{k,m} > 0$  so that if  $d > k$ , then*

$$\sum_{d \leq |\mathbf{i}| \leq d+k} |X_{\mathbf{i}}|^m \leq c_{k,m} n(d+k, d). \quad (15)$$

In this proposition, we use the fact mentioned above that when  $d > k$ , then the sparse grid  $X$  is a union of  $X_{\mathbf{i}}$  over  $\mathbf{i}$  as indicated in the sum given here. Given this result, the following theorem is nearly immediate.

**Theorem 4.** *For fixed  $k$ , there is  $c_k > 0$  so that for  $d > k$ , the coefficients in the gPC expansion of  $A(d+k, d)(f)$  may be found using (14) with computation time bounded by  $c_k n(d+k, d)$ ; that is, for fixed  $k$ , the running time is linear in the number of grid points.*

We have not attempted to calculate the best possible  $c_k$  in the proof of this result. In fact, numerical results given below show that in some cases, the coefficient  $c_k$  actually decreases as  $k$  increases due to the spread of fixed overhead costs over a small number of points when  $k$  and  $d$  are small. Additionally, the numerical results show that the time per point evaluated is roughly constant through  $k = 8$ .

Note that this algorithm computes the gPC coefficients of the interpolating polynomial rather than the original function itself. However, for a  $C^r$  function,  $f$ , with  $r \geq 1$ , the error estimate from (5) implies that the interpolating function converges uniformly to  $f$  as the depth,  $k$ , increases. Since the gPC coefficients are obtained by weighted integration of  $f$  against the orthogonal polynomials, the gPC coefficients for the interpolating polynomials converge to the gPC coefficients for  $f$ , and (5) provides a means to estimate the error in these coefficients based on the set of orthogonal polynomials and their corresponding weights.

*Proof of Proposition 3.* Given  $|\mathbf{i}| = d + \kappa$  between  $d$  and  $d + \kappa$ , consider the number of points in the corresponding grid  $X_{\mathbf{i}}$ . Let  $r = r(\mathbf{i})$  be the number of nontrivial entries in  $\mathbf{i}$  (i.e., entries larger than 1). Suppose without loss that the first  $r$  entries of  $\mathbf{i}$  are nontrivial with entries  $i_j = \kappa_j + 1 \geq 2$ . Then, the number of points in this grid is  $\prod_{j=1}^r (2^{\kappa_j} + 1)$ . If  $\kappa_1$  is the only entry larger than 1, then  $\kappa_1 = \kappa - (r - 1)$ , and so the number of points is exactly  $3^{r-1} (2^{\kappa-r+1} + 1)$ . A simple counting argument implies that any other distribution of nontrivial entries produces no more grid points, so

$$|X_{\mathbf{i}}| \leq 3^{r-1} 2^{\kappa-r+2}. \quad (16)$$

To construct  $\mathbf{i}$  with  $|\mathbf{i}| = d + \kappa$  and  $r$  nontrivial entries, we select  $r$  out of  $d$  nontrivial entries, then distribute  $\kappa$  elements to these entries and so that each entry gets at least one element. There are  $\binom{d}{r} \binom{\kappa-1}{r-1}$  ways to do this. Summing from  $r = 1$  to  $\kappa$ , and using  $\binom{d}{r} \leq d^r$ , we have

$$\begin{aligned} \sum_{|\mathbf{i}|=d+\kappa} |X_{\mathbf{i}}|^m &\leq \sum_{r=1}^{\kappa} \binom{d}{r} \binom{\kappa-1}{r-1} (3^m)^{r-1} (2^m)^{\kappa-r+2} \\ &\leq (2^m)^{\kappa+1} d \sum_{q=0}^{\kappa-1} \binom{\kappa-1}{q} \frac{(3^m)^q d^q}{(2^m)^q}. \end{aligned} \quad (17)$$

The summation on the right hand side is a binomial expansion of a power of  $\kappa - 1$ , and distributing a factor of  $(2^m)^{\kappa-1}$

into this power converts the right hand side to  $4^m d(3^m d + 2^m)^{\kappa-1}$ . Using this in place of the sum over  $\mathbf{i}$  and summing over  $\kappa$  gives

$$\begin{aligned} 4^m d \sum_{\kappa=1}^k (3^m d + 2^m)^{\kappa-1} &= 4^m d \frac{(3^m d + 2^m)^k - 1}{3^m d + 2^m - 1} \\ &\leq 2d^k 4^m \left(3^m + \frac{2^m}{d}\right)^{k-1}. \end{aligned} \quad (18)$$

As noted above,  $n(d+k, d)$  is asymptotically equal to  $2^k d^k / k!$  for large  $d$  in the sense that for fixed  $k$ ,  $\lim_{d \rightarrow \infty} n(d+k, d) / (2^k d^k / k!) = 1$ . In particular, there is  $a_k > 0$  so that  $d^k \leq a_k n(d+k, d)$  for all  $d \geq 1$ . Hence, the results just given imply

$$\sum_{\kappa=1}^k \sum_{|\mathbf{i}|=d+\kappa} |X_{\mathbf{i}}|^m \leq 2a_k n(d+k, k) 4^m (2 \cdot 3^m)^{k-1}. \quad (19)$$

Hence, taking  $c_{k,m} = 2a_k 4^m (2 \cdot 3^m)^{k-1}$ , we obtain the desired result.  $\square$

*Proof of Theorem 4.* Given the sparse grid,  $X$ , associated with  $A(d+k, k)$  and the vector of values,  $\mathbf{f}$ , we need first to form the matrices  $\pi$ ,  $\widehat{D}_w$ , and  $\widehat{\Phi}$  as in (13). From the construction of  $\pi$ , it has exactly one nonzero entry per row and is  $n \times N$ , where  $n = n(d, k)$  is asymptotic to  $2^k d^k / k!$  as  $d$  increases [10], and  $N = \sum_{d \leq |\mathbf{i}| \leq d+k} |X_{\mathbf{i}}|$ . Hence,  $\pi$  and the diagonal matrix,  $\widehat{D}_w$ , may be constructed and applied in time  $O(N)$ .

A given block of  $\widehat{\Phi}$  is obtained by evaluating products of one-dimensional polynomials on the points of a product grid,  $X_{\mathbf{i}}$ . For a nontrivial entry  $\mathbf{i}_j = 1 + \kappa_j$ , the polynomials in  $x_j$  have degree at most  $2^{\kappa_j}$ . Using a three-term recurrence, these polynomials may be evaluated at a given point in  $O(2^{\kappa_j})$ . Since the projection of  $X_{\mathbf{i}}$  to the  $x_j$  coordinate has  $2^{\kappa_j} + 1$  points, these polynomials are evaluated in time  $O(2^{2\kappa_j})$ . This is repeated for each nontrivial  $\kappa_j$  for a total of  $O(k2^{2\kappa_j})$ , which is  $O(k|X_{\mathbf{i}}|^2)$ . Then for each point in  $X_{\mathbf{i}}$ , we multiply at most  $k$  polynomials, which are  $O(k|X_{\mathbf{i}}|)$ . Summing over  $\mathbf{i}$  and applying Proposition 3, we see that  $\widehat{\Phi}$  may be constructed in time  $O(k n(d+k, d))$ .

As noted above, to find the gPC coefficients,  $c$ , we define  $\widehat{\mathbf{f}} = \pi \mathbf{f}$ , solve  $\widehat{\Phi} \widehat{c} = \widehat{\mathbf{f}}$  for  $\widehat{c}$ , and then use  $c = \pi^T \widehat{D}_w \widehat{c}$ . Since  $\widehat{\Phi}$  has blocks of size  $|X_{\mathbf{i}}| \times |X_{\mathbf{i}}|$ , we may solve each block in time at most  $O(|X_{\mathbf{i}}|^3)$ . Combining this with Proposition 3 and the estimates  $O(N)$  and  $O(kN)$  above, we obtain the theorem.  $\square$

A practical point is that many of the blocks in  $\widehat{\Phi}$  are identical up to a permutation of the rows. Hence, for each of these blocks, we may use a single LU decomposition and appropriate permutations of the entries in  $\mathbf{f}$  to reduce the total running time (although not, of course, the linear dependence described in the theorem).

We note also that the algorithm given here is compatible with the anisotropic adaptive sparse grids of [19]. The analysis

given here shows that the time per point depends only on the maximum size of a block in the factorization of  $\Phi^{-1}$ , and this size is bounded by the maximum depth,  $k$ , in both the isotropic and the anisotropic cases.

Finally, the analysis for fixed  $k$  and increasing  $d$  is relevant in that as models become more complex,  $d$  increases, while often the important behavior of a model can be captured by a relatively low-order approximation. An example of this is seen in the T-cell signaling model in Section 4.2.

## 4. Numerical results

*4.1. Efficiency of Conversion.* In this section, we provide results of numerical experiments using the algorithm described in the previous sections. All computations were performed in MATLAB 7.7.0 on a Dell Precision PWS690 with an Intel Xeon running at 3 GHz with 3 GB of RAM.

To evaluate the running time of the conversion algorithm, we used the test function labeled oscillatory in [10] as

$$f_1(x) = \cos\left(2\pi w_1 + \sum_{i=1}^d c_i x_i\right), \quad (20)$$

where  $c_i$  and  $w_1$  are chosen at random as indicated in [22]. The domain of definition is  $[0, 1]$ . The sparse grid for a given dimension and depth was created using the MATLAB package spinterp, version 5.1.1 [23]. The resulting functional values were then used as input to the revised conversion algorithm using the Legendre polynomials as basis.

In Figure 1, we plot the running time of the conversion algorithm for fixed order of accuracy,  $k$ , and increasing dimension. Figure 4(a) clearly shows the linear dependence on the number of points evaluated. Figure 4(b) shows the same data as a function of the number of dimensions. Here, the nonlinear increase for  $k$  bigger than 1 is due to the nonlinear dependence of the number of points of evaluation on dimension. Nevertheless, for  $k = 2$ , the algorithm is reasonably fast even up to 100 dimensions.

In Figure 2, we plot the running time of the conversion algorithm for fixed dimension and increasing order of accuracy,  $k$ . Here, the bound in (18) implies that the running time is bounded by  $c_1 c_2^k$  per point for some constants  $c_1$  and  $c_2$ . However, Figure 4(a) shows that the deviation from nonlinear is relatively small for small  $k$ . This is made more precise in Figure 4(b), which shows the time per point as a function of  $k$ . Perfect linear dependence would imply that the traces for different dimensions would coincide. While this is essentially true for  $k \geq 5$  in the data presented, there are significant deviations for small  $k$  and  $d$ . This is due to fixed overhead time that must average over fewer points in these cases, which gives rise to the decrease in time per point as dimension increases when  $k \leq 4$ . Somewhat more surprisingly, the time per point (after discounting the fixed overhead) is essentially constant up to  $k = 8$ .

*4.2. Sensitivity Analysis for T-Cell Signaling Model.* To illustrate the calculation of global sensitivity coefficients as described in Section 2.2, we consider a mathematical model

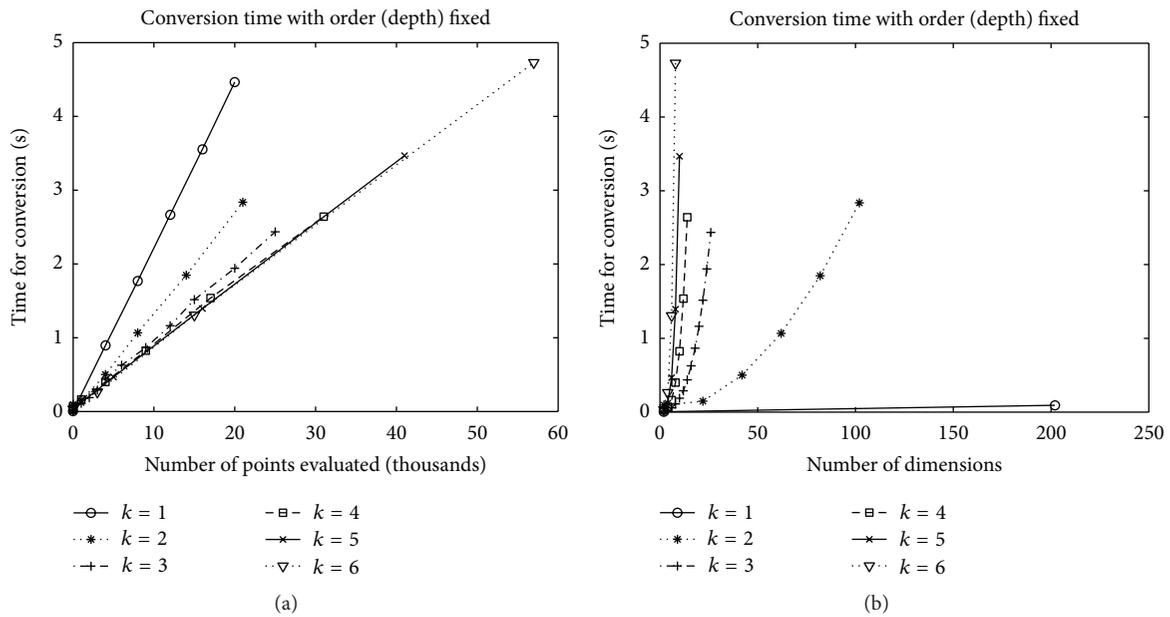


FIGURE 1: Time for conversion to Legendre basis as a function of number of points and number of dimensions. Each trace corresponds to a fixed order of accuracy (or depth,  $k$ ) with the number of dimensions increasing. (a) Time versus number of points showing linear dependence on the number of points evaluated. (b) The same data as a function of number of dimensions (except,  $k = 1$  is truncated for scaling reasons). Given the linear relationship on the right, this is essentially a scaled plot of the number of points as a function of the number of dimensions.

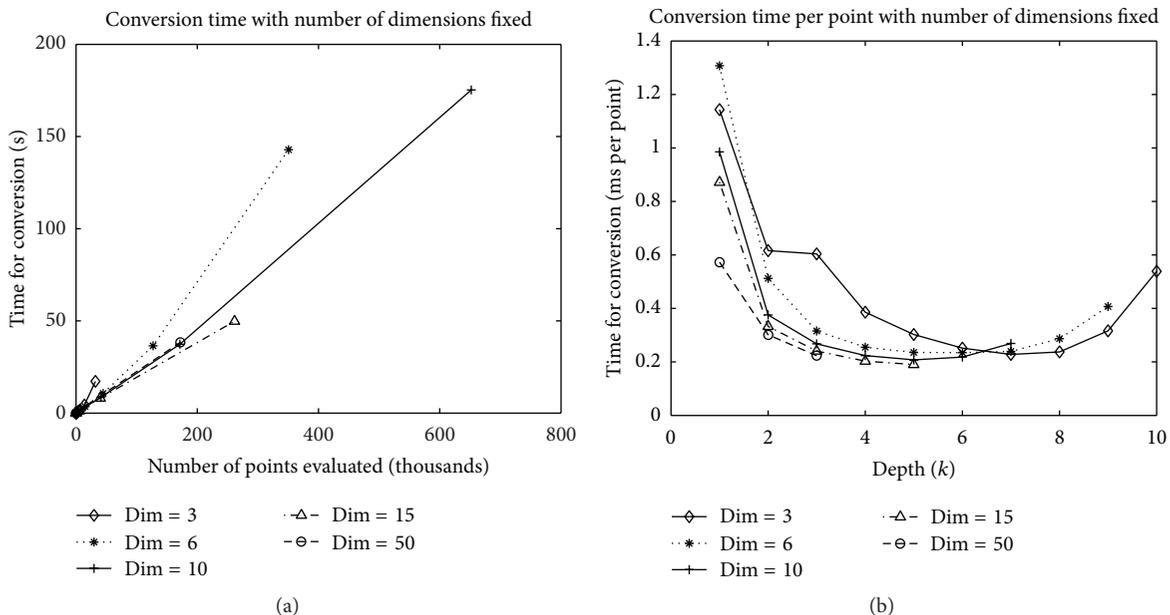


FIGURE 2: Time for conversion to Legendre basis as a function of number of points and order of accuracy (or depth,  $k$ ). Each trace corresponds to a fixed number of dimensions with the depth increasing. (a) Time versus number of points showing nonlinear dependence on  $k$ , particularly in dimensions 3 and 6, for which the maximum depth is 10 and 9, respectively. (b) The same data scaled to give time per point of evaluation and is shown as a function of  $k$ . Perfect linear scaling of time with number of points would imply that the time per point is the same for a given value of  $k$ , independent of dimension. Apart from fixed overhead costs that skew the results for small  $k$ , the time per point is nearly constant over a large range of feasible values of  $k$ .

of the T-cell signaling pathway proposed by Lipniacki et al. in [24] and examined further in [25]. This model describes the effect on T-cell activation of agonist and antagonist binding of major histocompatibility complex molecules, with both positive and negative feedback present in the signaling network. We use a form of this model that is a standard ODE system with 19 parameters, 37 state variables, and fixed initial conditions. We focus on the dynamics of phosphorylated ZAP, which plays an important role in positive feedback through the MEK/ERK kinase cascade. We take the uncertain parameter space to be a hypercube with sides given by  $[p_i/5, 5p_i]$ , where  $p_i$  is the nominal  $i$ th parameter value. For purposes of constructing the gPC expansion and calculating sensitivity coefficients, we transform the parameters to log space. Hence, the function that we evaluate has the form  $f(t; q)$ , where  $q$  is a vector of parameters given by  $q_i = p_i 5^{x_i}$ , with each  $x_i \in [-1, 1]$ , and  $t \in [0, 2000]$  is time.

More precisely, for this example, the nodes of a sparse grid are vectors in  $[-1, 1]^{19}$ , and each such node gives a corresponding parameter vector  $q$  using the transformation just described. For each such  $q$ , we evaluate the model at 200 evenly spaced time points between 0 and 2000. For one of these fixed  $t$ , say  $t_i$ , the model evaluations  $f(t_i; q)$  over all the  $q$  given by nodes of a sparse grid allow us to construct an interpolating polynomial, which can then be evaluated at any  $q$  in the parameter space, even those that do not arise from nodes of the sparse grid. Of course the resulting interpolation is only an approximation to the true model, but the evaluation of the interpolating polynomial is in this case on the order of 20 times faster than evaluation by integrating the system of ODEs directly, and after change of basis to gPC form, the interpolating polynomial can be used to compute sensitivity coefficients very quickly. Moreover, the remarks in Section 2 quantify the relationship between the number of nodes in the sparse grid and the accuracy of the approximation.

Over the selected parameter space, this model has dynamics that can be difficult to capture using interpolation methods. Depending on parameters, the dynamics can exhibit dramatically different time scales, and, moreover, there is a division of parameter space into regions that produce trajectories with relatively high terminal concentrations of pZAP and other regions with relatively low terminal concentrations; very few trajectories have terminal concentrations in an intermediate range. These features can be difficult to capture accurately using polynomial interpolation in the same way that step functions can be difficult to approximate with Fourier series. Figure 3 shows a set of sample trajectories of the full model along with trajectories obtained by interpolation using an increasing number of model evaluations. The parameter vectors for these trajectories were chosen randomly from the full parameter space and thus are not sparse grid nodes. These plots show both the difficulty in capturing the dynamics accurately (e.g., the overshoot in the trajectory that rises most quickly) and the fact that interpolation captures the main features of the dynamics even in this difficult case and with accuracy increasing with the number of points.

After evaluating the function on a sparse grid, we next computed the Sobol' sensitivity coefficients for each of the 19

parameters as a function of time. As described in Section 2.2, this is accomplished by first using the function values  $f(t_i; q)$  for fixed  $t_i$  to change to a gPC basis and thereby obtain an interpolating polynomial for  $f(t_i; \cdot)$ ; the sensitivity coefficients for this time point are then obtained by summing the squares of coefficients of appropriate terms in this polynomial. This is done for each  $t_i$  to obtain the sensitivity coefficients as a function of time. The resulting curves are shown in Figure 4, using interpolating polynomials based on 100 function values (a) and 10000 function values (b). As seen there, the estimates of the sensitivity coefficients change in magnitude between the two panels, but the ranking of sensitive versus insensitive is quite consistent between the two. Moreover, this ranking is nearly identical to that given by an independent estimation using a quasi-Monte Carlo method.

To demonstrate the utility of this sensitivity analysis, we note that all but 4 of the parameters have sensitivity coefficients that are below 0.1 for the entire time course. The 4 remaining parameters have 3 distinct patterns of influence on the sensitivity. The first pattern is due to the ZAP dephosphorylation rate, which gives rise to a sharp spike in sensitivity near the beginning of the time course (in black in Figure 4). This sensitivity at the start of the time course is related to the way in which the model departs from the fixed initial conditions; a larger dephosphorylation leads to a steeper initial decline in the level of pZAP. However, examination of trajectories arising from a range of values for this parameter shows that there is less than 2% variation in these trajectories over the part of the curve corresponding to the peak of sensitivity. Hence, although the sensitivity to this parameter is high in this region, the reason for this sensitivity is not that the trajectories vary greatly in response to changes in this parameter, but rather that changes to the other parameters lead to essentially no change to the output in the early part of the trajectory.

The next sensitivity pattern is a moderate hump in the time period of about 50 to about 500 (shown in green and red in Figure 4). These are due to the T-cell receptor (TCR) phosphorylation rate (the upper, green curve) and the ZAP phosphorylation rate (the lower, red curve). Not surprisingly, with the remaining parameters held fixed, increased values for either of these rates generally lead to significantly increased levels of pZAP from about time 50 onward.

The final sensitivity pattern is shown in Figure 4 in the blue curve that rises around time 500, which is due to the SHP phosphorylation rate. SHP plays a role in a negative feedback loop, and this nonlinear dependence gives rise to complex and often long-lasting influences on the trajectories.

To indicate the utility of the sensitivity analysis in capturing the dynamics of this model, we took the parameter values used in Figure 3, changed all but the 4 parameters discussed as most sensitive back to the nominal values of the original model, and then resimulated. The resulting curves in Figure 5(a) indicate that this simple procedure captures the majority of the dynamical features of the original trajectories of Figure 3(a). We then used sparse grid interpolation on the 4-dimensional grid corresponding to these 4 parameters and produced interpolated curves as in Figure 3. The smaller

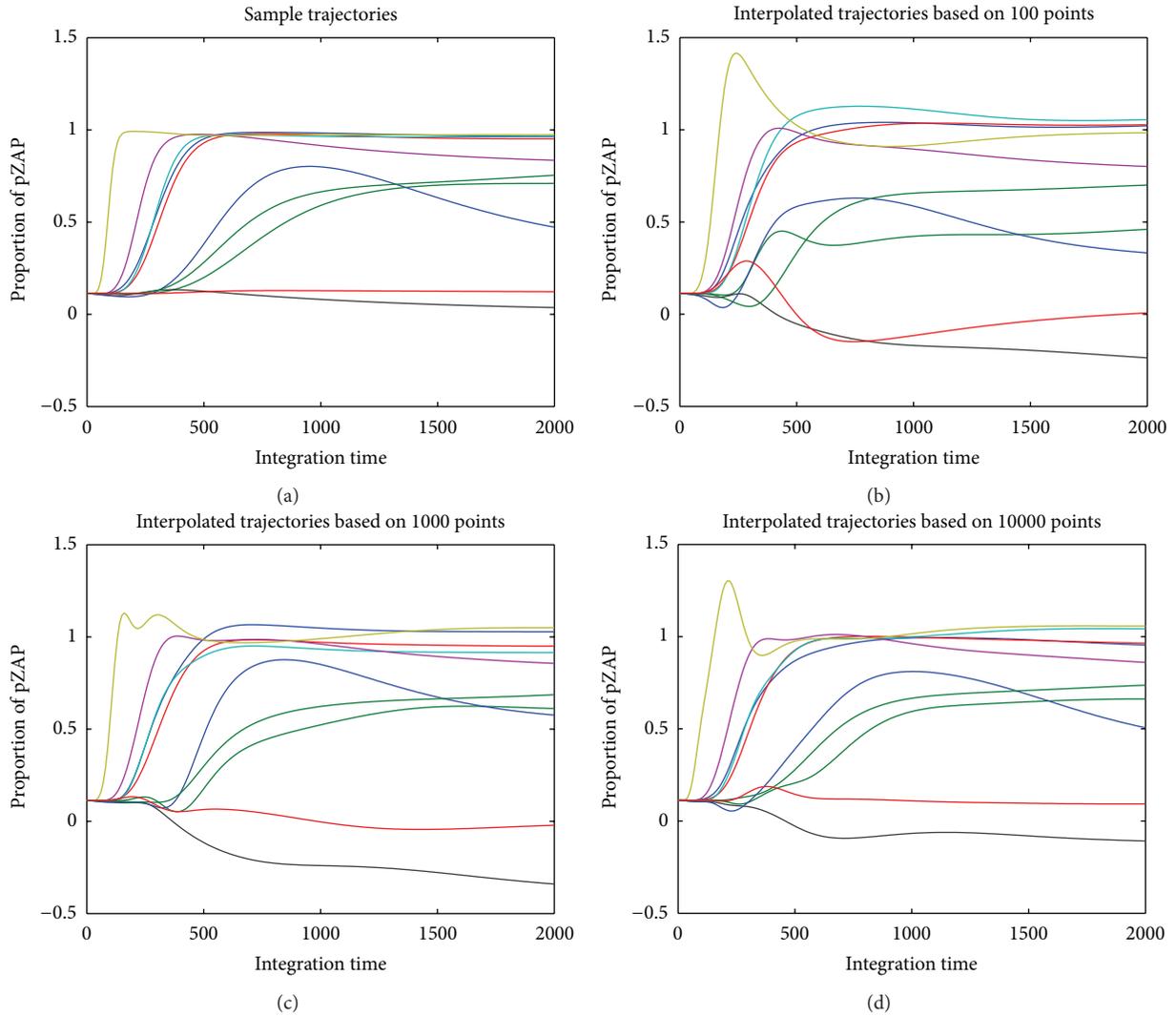


FIGURE 3: Size sample trajectories for the Lipniacki model, showing the proportion of total ZAP in phosphorylated state (pZAP) versus integration time. (a) shows the computed trajectories. (b)–(d) show interpolated trajectories based on the indicated number of sparse grid nodes.

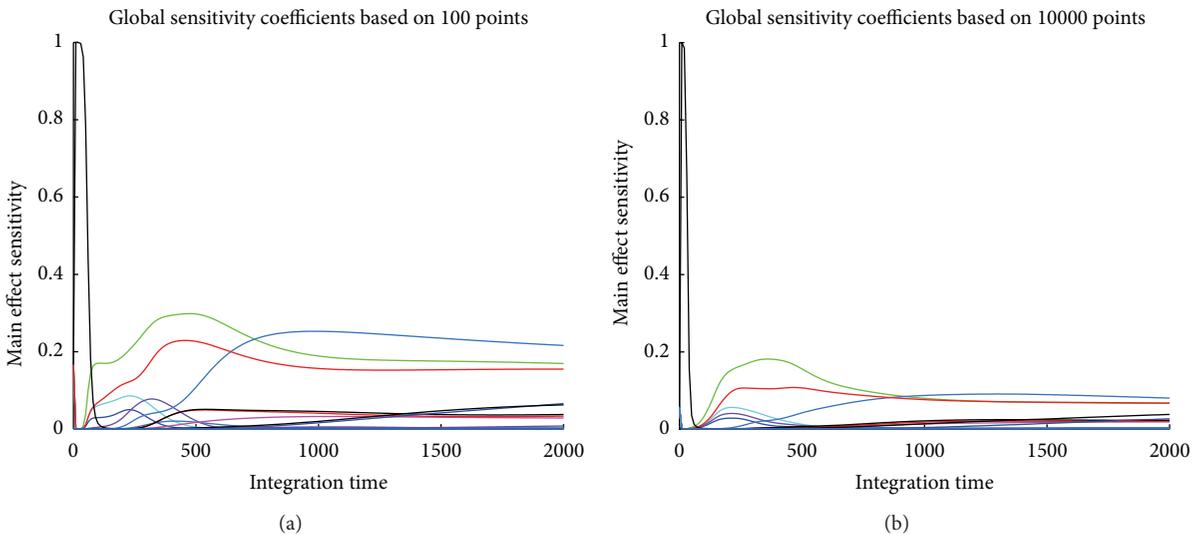


FIGURE 4: Sobol' sensitivity coefficients for each of the 19 parameters as a function of time, based on the indicated number of sparse grid nodes. Although the magnitude of the estimates change slightly between the two plots, the relative ranking is essentially the same.

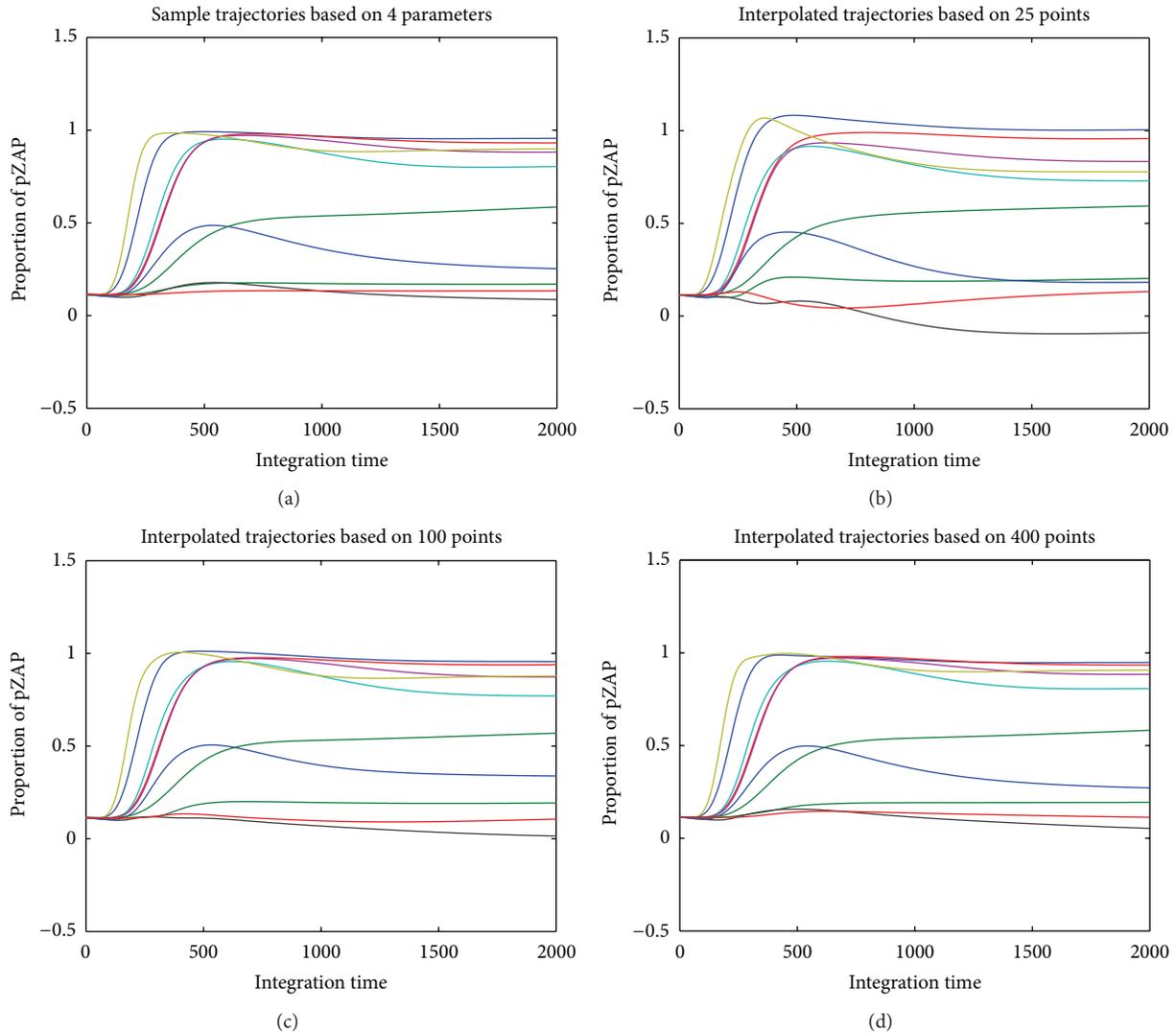


FIGURE 5: Sample trajectories corresponding to those in Figure 3 but with using nominal parameter values for all but the 4 parameters identified as the most sensitive. (a) shows the trajectories obtained by integrating the system of ODEs. (a)–(b) show interpolated trajectories based on the indicated number of points in 4-dimensional sparse grid.

dimension here means that far fewer points are needed to accurately capture the trajectories relative to the 19 dimensional cases examined earlier.

## 5. Conclusions

The algorithm described here provides an efficient method to convert function values to a gPC approximation to a given function and thereby to estimate global Sobol' sensitivity coefficients. The method relies on a block-diagonal factorization of the matrix for changing basis from Lagrange polynomials to orthogonal polynomials based on function values at a Smolyak sparse grid. For a fixed degree of accuracy and increasing dimension, this algorithm is linear in the number of points of evaluation. Moreover, for fixed dimension, the time per point of evaluation is nearly constant as the degree of accuracy increases up to about  $k = 8$ .

We applied this method to a time-varying model of T-cell signaling and showed that the majority of the dynamics of this model could be explained by variation in only 4 parameters: the ZAP dephosphorylation rate, and the ZAP, TCR, and SHP phosphorylation rates. Taken together, the combination of efficient sensitivity analysis and interpolation provides an easily applied and efficient method for understanding the main dynamics of a complex biological model.

## Acknowledgments

This research is partially supported under NSF Grant DMS-0900277. The author is grateful to Vu Dinh for the use of existing Matlab code to simulate the T-cell model presented here and to Ann Rundell and her group members for many helpful discussions.

## References

- [1] J. Shen and L. L. Wang, "Sparse spectral approximations of high-dimensional problems based on hyperbolic cross," *SIAM Journal on Numerical Analysis*, vol. 48, no. 3, pp. 1087–1109, 2010.
- [2] D. Xiu and J. S. Hesthaven, "High-order collocation methods for differential equations with random inputs," *SIAM Journal on Scientific Computing*, vol. 27, no. 3, pp. 1118–1139, 2006.
- [3] A. Saltelli, S. Tarantola, and K. P. S. Chan, "A quantitative model-independent method for global sensitivity analysis of model output," *Technometrics*, vol. 41, no. 1, pp. 39–56, 1999.
- [4] T. Crestaux, O. Le Maître, and J. M. Martinez, "Polynomial chaos expansion for sensitivity analysis," *Reliability Engineering and System Safety*, vol. 94, no. 7, pp. 1161–1172, 2009.
- [5] G. T. Buzzard and D. Xiu, "Variance-based global sensitivity analysis via sparse-grid interpolation and cubature," *Communications in Computational Physics*, vol. 9, no. 3, pp. 542–567, 2011.
- [6] M. M. Donahue, G. T. Buzzard, and A. E. Rundell, "Experiment design through dynamical characterisation of non-linear systems biology models utilising sparse grids," *IET Systems Biology*, vol. 4, no. 4, pp. 249–262, 2010.
- [7] G. Li, H. Rabitz, J. Hu, Z. Chen, and Y. Ju, "Regularized random-sampling high dimensional model representation (RS-HDMR)," *Journal of Mathematical Chemistry*, vol. 43, no. 3, pp. 1207–1232, 2008.
- [8] D. Xiu, "Fast numerical methods for stochastic computations: a review," *Communications in Computational Physics*, vol. 5, no. 2–4, pp. 242–272, 2009.
- [9] A. Saltelli, M. Ratto, T. Andres et al., *Global Sensitivity Analysis. The Primer*, John Wiley & Sons, 2008.
- [10] V. Barthelmann, E. Novak, and K. Ritter, "High dimensional polynomial interpolation on sparse grids," *Advances in Computational Mathematics*, vol. 12, no. 4, pp. 273–288, 2000.
- [11] S. A. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Soviet Mathematics Doklady*, vol. 4, pp. 240–243, 1963.
- [12] J. P. Berrut and L. N. Trefethen, "Barycentric Lagrange interpolation," *SIAM Review*, vol. 46, no. 3, pp. 501–517, 2004.
- [13] B. K. Alpert and V. Rokhlin, "A fast algorithm for the evaluation of Legendre expansions," *SIAM Journal on Scientific Computing*, vol. 12, pp. 158–179, 1991.
- [14] A. Bostan, B. Salvy, and É. Schost, "Power series composition and change of basis," in *Proceedings of the 21st Annual Meeting of the International Symposium on Symbolic Computation (ISSAC '08)*, pp. 269–276, ACM, New York, NY, USA, July 2008.
- [15] W. Gander, "Change of basis in polynomial interpolation," *Numerical Linear Algebra with Applications*, vol. 12, no. 8, pp. 769–778, 2005.
- [16] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, Dover, New York, NY, USA, 2nd edition, 2001.
- [17] J. P. Boyd, "Exponentially convergent Fourier-Chebyshev quadrature schemes on bounded and infinite intervals," *Journal of Scientific Computing*, vol. 2, no. 2, pp. 99–109, 1987.
- [18] M. Griebel, M. Schneider, and C. Zenger, "A combination technique for the solution of sparse grid problems," in *Methods in Linear Algebra*, P. de Groen and R. Beauwens, Eds., pp. 263–281, Elsevier, Amsterdam, The Netherlands, 1992.
- [19] T. Gerstner and M. Griebel, "Dimension-adaptive tensor-product quadrature," *Computing*, vol. 71, no. 1, pp. 65–87, 2003.
- [20] T. Ullrich, "Smolyak's algorithm, sampling on sparse grids and Sobolev spaces of dominating mixed smoothness," *East Journal on Approximations*, vol. 14, no. 1, pp. 1–38, 2008.
- [21] I. M. Sobol', "Sensitivity estimates for nonlinear mathematical models," *Mathematical Modeling and Computational Experiment*, vol. 1, pp. 407–414, 1993 (Russian), translation of Sensitivity estimates for nonlinear mathematical models. *Matematicheskoe Modelirovanie*, vol. 2, pp. 112–118, 1990.
- [22] A. C. Genz, "A package for testing multiple integration subroutines," in *Numerical Integration: Recent Developments, Software, and Applications*, P. Keast and G. Fairweather, Eds., pp. 337–340, Kluwer, Dodrecht, The Netherlands, 1987.
- [23] A. Klimke and B. Wohlmuth, "Algorithm 847: spinterp: piecewise multilinear hierarchical sparse grid interpolation in MATLAB," *ACM Transactions on Mathematical Software*, vol. 31, no. 4, pp. 561–579, 2005.
- [24] T. Lipniacki, B. Hat, J. R. Faeder, and W. S. Hlavacek, "Stochastic effects and bistability in T cell receptor signaling," *Journal of Theoretical Biology*, vol. 254, no. 1, pp. 110–122, 2008.
- [25] J. N. Bazil, G. T. Buzzard, and A. E. Rundell, "A global parallel model based design of experiments method to minimize model output uncertainty," *Bulletin of Mathematical Biology*, vol. 74, no. 3, pp. 688–716, 2012.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

