Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 225 (2007) 652–685
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Sparse grid collocation schemes for stochastic natural convection problems

Baskar Ganapathysubramanian, Nicholas Zabaras *

Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, 188 Frank H.T. Rhodes Hall, Cornell University, Ithaca, NY 14853-3801, USA

Received 17 August 2006; received in revised form 15 December 2006; accepted 20 December 2006
Available online 10 January 2007

Abstract

In recent years, there has been an interest in analyzing and quantifying the effects of random inputs in the solution of partial differential equations that describe thermal and fluid flow problems. Spectral stochastic methods and Monte-Carlo based sampling methods are two approaches that have been used to analyze these problems. As the complexity of the problem or the number of random variables involved in describing the input uncertainties increases, these approaches become highly impractical from implementation and convergence points-of-view. This is especially true in the context of realistic thermal flow problems, where uncertainties in the topology of the boundary domain, boundary flux conditions and heterogeneous physical properties usually require high-dimensional random descriptors. The sparse grid collocation method based on the Smolyak algorithm offers a viable alternate method for solving high-dimensional stochastic partial differential equations. An extension of the collocation approach to include adaptive refinement in important stochastic dimensions is utilized to further reduce the numerical effort necessary for simulation. We show case the collocation based approach to efficiently solve natural convection problems involving large stochastic dimensions. Equilibrium jumps occurring due to surface roughness and heterogeneous porosity are captured. Comparison of the present method with the generalized polynomial chaos expansion and Monte-Carlo methods are made.

Keywords: Collocation methods; Natural convection; Stochastic partial differential equations; Sparse grids; Adaptive sampling; Navier–Stokes equations; Stochastic Galerkin method

1. Introduction

With the rapid advances in computational power and easier access to high-performance computing platforms, it has now become possible to computationally investigate realistic multiscale problems. The coupling of various length and time scales in a deterministic solution give highly accurate predictions of the evolution and stability of the investigated system. One of the key objectives of computational mechanics would be to couple all the length scales from the atomistic through to the continuum, thus removing the need for any

* Corresponding author. Fax: +1 607 255 1222.
E-mail address: zabaras@cornell.edu (N. Zabaras).
URL: http://mpdc.mae.cornell.edu (N. Zabaras).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.12.014

constitutive models or physical parameters to represent the system. But the present state-of-art in computational mechanics has been to couple only up to three scales of physics [1]. So there exists the need to input constitutive relations and/or material properties to the model. These inputs are usually available or derived from experimental data. The presence of uncertainties/perturbations in experimental studies implies that these input parameters have some inherent uncertainties. To accurately predict the performance of the system, it then becomes essential for one to include the effects of these input uncertainties into the model system and understand how they propagate and alter the final solution.

The uncertainties in the physical model may also be due to inaccuracies or indeterminacies of the initial conditions. This may be due to variation in experimental data or due to variability in the operating environment. Uncertainties may also creep in due to issues in describing the physical system. This may be caused by errors in geometry, roughness and variable boundary conditions. Uncertainties could also occur due to the mathematical representation of the physical system. Examples include errors due to approximate mathematical models (e.g. linearization of non-linear equations) and discretization errors.

The presence of uncertainties can be modelled in the system through reformulation of the governing equations as stochastic partial differential equations (SPDEs). Solution techniques for SPDEs can be broadly classified into two approaches: statistical and non-statistical. Monte-Carlo simulations and its variants constitute the statistical approach to solve SPDEs. This approach gives access to the complete statistics. This method does not approximate the solution space (like the other methods discussed later). The other major attraction of these methods are that their convergence rates do not depend on the number of independent random variables. Further, they are very easy to implement. The only necessity of computing uncertainties using the statistical approach is the availability of a working deterministic code. However, the statistical approach becomes quickly intractable for complex problems in multiple random dimensions. This is because the number of realizations required to acquire good statistics is usually quite large. Furthermore, the number of realizations changes with the variance of the input parameters and the truncation errors in this inherently statistical method are hard to estimate. This has in part been alleviated by improved sampling techniques like Latin hypercube sampling and importance sampling among others [2].

A non-statistical approach consists of approximating and then modelling the uncertainty in the system. One example of this approach is the perturbation method. Here, the random field is expanded about its mean in a truncated Taylor series. This method is usually restricted to small values of uncertainties. A similar method is to expand the inverse of the stochastic operator in a Neumann series. This method is again restricted to small uncertainties and furthermore it is almost impossible to apply these methods to complex problems [3]. In the context of expanding the solution in terms of its statistical moments, there has been some recent work in reformulating the problem as a set of moment differential equations (MDE). The closure equations for these MDE's are then derived from Taylor expansions of the input uncertainty as well as perturbation and asymptotic expansions. The random domain decomposition method [4–7] coupled with the MDE method has been shown to be successful in solving complex fluid flow problems in porous media having large input variances.

A more recent approach to model uncertainty is based on the spectral stochastic finite element method (SSFEM) [3]. In this method, the random field is discretized directly, i.e uncertainty is treated as an additional dimension along with space and time and a field variable is expanded along the uncertain dimension using suitable expansions. The basic idea is to project the dependent variables of the model onto a stochastic space spanned by a set of complete orthogonal polynomials. These polynomials are functions of a set of random variables $\boldsymbol{\xi}(\theta)$ where θ is a realization of the random event space. Ghanem and Spanos first used SSFEM in linear elastic problems in [3]. Ghanem and co-workers subsequently applied it to transport in random media [8], structural dynamics applications [9] and heat conduction problems [10]. Karniadakis and co-workers applied a generalized polynomial chaos expansion to model uncertainty in diffusion [11], fluid flow applications [12] and transient heat conduction problems [13]. In the original work by Wiener, Gaussian random variables were used with Hermite polynomials. The scheme has been extended to include other random distributions [14–19]. Error bounds, and convergence studies [20–24] have shown that these methods exhibit fast convergence rates with increasing orders of expansions. These convergence studies assume that the solutions are sufficiently smooth in the random space.

Like the statistical methods, the SSFEM approach also reduces the problem to a set of deterministic equations. But unlike the statistical methods, the resulting deterministic equations are often coupled. This coupled

nature of the deterministic problems makes the solution of the stochastic problem extremely complex as the number of stochastic dimensions increases or the number of expansion terms is increased. In fact, computational complexity of the problem increases combinatorially with the number of stochastic dimensions and the number of expansion terms. This problem is especially acute when dealing with realistic problems in fluid mechanics. The number of random dimensions needed to represent surface fluctuations (roughness) or permeability variations is relatively high ($N \geqslant 8$). To solve these problems in high-dimensional spaces and to allow non-smooth variations of the solution in the random space, there have been recent efforts to couple the fast convergence of the Galerkin methods with the decoupled nature of Monte-Carlo sampling [25]. Babuska et al. [26] proposed a methodology wherein the Galerkin approximation is used to model the physical space while a collocation scheme is used to sample the random space. A tensor product rule was used to interpolate the variables in stochastic space using products of one-dimensional interpolation functions (double orthogonal polynomials) based on Gauss points. Though this scheme effectively decoupled the deterministic system of equations, the number of realizations required to build the interpolation scheme increased as power of the number of random dimensions (n_{pt}^N). Xiu and Hesthaven [27] recently used the Smolyak algorithm to build sparse grid interpolants in high-dimensional space. The sparse grid collocation and cubature schemes have been well studied and utilized in different fields [28–32]. Using this method, interpolation schemes can be constructed with orders of magnitude reduction in the number of sampled points to give the same level of approximation (up to a logarithmic factor) as interpolation on a uniform grid.

In the standard sparse grid collocation approach, all the dimensions are treated equally. In many problems usually encountered, not all the dimensions are important. That is, the solution varies much more smoothly in some particular dimension than in others. This brings up the possibility of reduction in the computational effort required to solve the stochastic problem by weighting the number of sampling points in the stochastic dimensions according to the solution smoothness in that dimension. But it is not possible to know a priori which dimensions are more important. It is however possible to develop heuristic methods to adaptively sample the stochastic space to reduce the number of function evaluations.

The major contributions of the present work are as follows: We utilize an adaptive extension to the sparse grid collocation method that can automatically detect the important dimensions and bias the sparse sampling towards those dimensions. This results in substantial savings in the computational effort required to solve large dimensional problems. We further utilize these methods to tackle significant problems in natural convection like investigating the effects of surface roughness on the dynamics of Rayleigh–Bénard convection and the effects of heterogeneous porous media on natural convection. We are able to accurately capture equilibrium shifts and efficiently scale to a large number of stochastic dimensions. This work further provides a road map to convert any deterministic code to include the effects of input uncertainty in a non-intrusive manner utilizing the adaptive sparse grid collocation method (ASGC). We illustrate this by analyzing the effect of input uncertainties in realistic natural convection problems.

The layout of this article is as follows: In the next section, we briefly describe the problem of interest and introduce the governing equations describing the system. Section 3 briefly describes the spectral Galerkin method as a prelude to introducing collocation based methods, which is described in Section 4. This section provides a natural progression of development from simple collocation to sparse collocation to adaptive sparse collocation. We have attempted to make the paper complete by including a brief of the mathematics that lead to the adaptive sparse grid collocation method. Sections 5 and 6 describe the conventional and adaptive sparse grid collocation methodologies. Section 7 provides a basic outline to construct a sparse grid collocation methodology in a non-intrusive way utilizing existing deterministic codes. The numerical examples are given in Section 8 which also includes some discussion of the relevant physics. We conclude in Section 9 with some comments and avenues of future work.

2. Problem definition

The main focus of this work is to investigate the effects of input uncertainties in natural convection problems. Consider a d-dimensional bounded domain $D \subset \mathbb{R}^d$ with a boundary $\partial D_d \bigcup \partial D_n$. Dirichlet boundary conditions are applied on ∂D_d, while Neumann boundary conditions are applied on ∂D_n. Natural convection sets in due to the thermal differential maintained across ∂D_d. Consider also a complete probability space $(\Omega, \mathscr{F}, \mathscr{P})$, where Ω

is the event space, $\mathscr{F} \subset 2^{\Omega}$ the σ-algebra, and $\mathscr{P} : \mathscr{F} \to [0,1]$ is the probability measure. In the problems that we consider, the input uncertainties occur due to the following random fields $f_i(\boldsymbol{x},\omega)$, $i = 1$, 2 or 3:

- Variations in the Dirichlet boundary conditions: $\theta(\boldsymbol{x}) = f_1(\boldsymbol{x},\omega)$, where $\boldsymbol{x} \in \partial D_d$ and $\omega \in \Omega$.
- Variations in the boundary topology (roughness): $\partial D_d = f_2(\boldsymbol{x},\omega)$, where $\boldsymbol{x} \in \partial D_d$ and $\omega \in \Omega$.
- Variations in material properties (viscosity, porosity): $\varrho(\boldsymbol{x}) = f_3(\boldsymbol{x},\omega)$, where $\boldsymbol{x} \in D$ and $\omega \in \Omega$.

Under these conditions the problem is to find stochastic functions that describe the velocity $\boldsymbol{u} \equiv \boldsymbol{u}(\boldsymbol{x},t,\omega) : D \times [0,T] \times \Omega \to \mathbb{R}^d$, the pressure $p \equiv p(\boldsymbol{x},t,\omega) : D \times [0,T] \times \Omega \to \mathbb{R}$ and the temperature $\theta \equiv \theta(\boldsymbol{x},t,\omega) : D \times [0,T] \times \Omega \to \mathbb{R}$, such that for $\mathscr{P}$ – almost everywhere $\omega \in \Omega$, the following equations are satisfied:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + Pr \nabla^2 \boldsymbol{u} + F(\boldsymbol{u},\theta), \tag{2}$$

$$\frac{\partial \theta}{\partial t} + \boldsymbol{u} \cdot \nabla \theta = \nabla^2 \theta, \tag{3}$$

where $F(\boldsymbol{u},\theta)$ is the forcing function in the Navier–Stokes equations and Pr is the Prandtl number of the fluid. In the problems considered later, $F(\boldsymbol{u},\theta)$ will usually be the Bousinnessq approximated buoyant force term $-RaPr\theta \boldsymbol{e}_{\mathrm{g}}$, where Ra is the thermal Rayleigh number and $\boldsymbol{e}_{\mathrm{g}}$ is the gravity vector.

The solution methodology of the above set of coupled differential equations is to first reduce the complexity of the problem by reducing the probability space into a finite-dimensional space. In some cases the random field $f(\boldsymbol{x},\omega)$ can be represented/described by a finite length random vector $[\xi^1,\ldots,\xi^N] : \Omega \to \mathbb{R}^N$. In other cases the random field can have a spatial correlation or variation. In the present work, we assume that we are given the corresponding spatial correlation for the random field. There is rich literature on techniques to extract/fit correlations for these random fields from input experimental/numerical data. This is an area of intense ongoing research [33]. Using the 'finite-dimensional noise assumption' [22], the random process is decomposed into a finite set of random variables. A commonly used decomposition is via the Karhunen–Loève expansion. In the K–L expansion, the covariance kernel of the input random process is spectrally decomposed to a finite set of uncorrelated random variables. Upon decomposition and characterization of the random inputs into N random variables, $\xi^i(\omega) i = 1,\ldots,N$, the solution to the coupled system of Eqs. (1)–(3) can be described by this set of random variables. Thus we can consider the following:

$$\boldsymbol{u}(\boldsymbol{x},t,\omega) = \boldsymbol{u}(\boldsymbol{x},t,\xi^1(\omega),\ldots,\xi^N(\omega)),$$
$$\theta(\boldsymbol{x},t,\omega) = \theta(\boldsymbol{x},t,\xi^1(\omega),\ldots,\xi^N(\omega)),$$
$$p(\boldsymbol{x},t,\omega) = p(\boldsymbol{x},t,\xi^1(\omega),\ldots,\xi^N(\omega)).$$

It is usually assumed that $\xi^i(\omega)$ are independent random variables with probability distribution functions $\rho_i : \Gamma^i \to \mathbb{R}$ with bounded ranges, Γ^i. The joint probability density of the N-tuple in the above equations $\boldsymbol{\xi} = (\xi^1(\omega),\ldots,\xi^N(\omega))$ is then given by

$$\rho(\boldsymbol{\xi}) = \prod_{i=1}^{N} \rho_i(\xi^i) \quad \forall \boldsymbol{\xi} \in \Gamma, \tag{4}$$

where the support $\Gamma = \prod_{i=1}^{N} \Gamma^i \subset \mathbb{R}^N$. The outcome of the above procedure is that the set of Eqs. (1)–(3) can now be written as a set of differential equations in $(N+d)$ dimensions, where N is the dimensionality of the truncated random space Γ and d is the dimensionality of the physical space D:

$$\nabla \cdot \boldsymbol{u}(\boldsymbol{x},t,\boldsymbol{\xi}) = 0, \quad (\boldsymbol{x},t,\boldsymbol{\xi}) \in D \times [0,T] \times \Gamma, \tag{5}$$

$$\frac{\partial \boldsymbol{u}(\boldsymbol{x},t,\boldsymbol{\xi})}{\partial t} + \boldsymbol{u}(\boldsymbol{x},t,\boldsymbol{\xi}) \cdot \nabla \boldsymbol{u}(\boldsymbol{x},t,\boldsymbol{\xi}) = -\nabla p(\boldsymbol{x},t,\boldsymbol{\xi}) + Pr \nabla^2 \boldsymbol{u}(\boldsymbol{x},t,\boldsymbol{\xi}) + F(\boldsymbol{u}(\boldsymbol{x},t,\boldsymbol{\xi}),\theta(\boldsymbol{x},t,\boldsymbol{\xi})),$$
$$(\boldsymbol{x},t,\boldsymbol{\xi}) \in D \times [0,T] \times \Gamma, \tag{6}$$

$$\frac{\partial\theta(\boldsymbol{x},t,\boldsymbol{\xi})}{\partial t}+\boldsymbol{u}(\boldsymbol{x},t,\boldsymbol{\xi})\cdot\nabla\theta(\boldsymbol{x},t,\boldsymbol{\xi})=\nabla^2\theta(\boldsymbol{x},t,\boldsymbol{\xi}),\quad(\boldsymbol{x},t,\boldsymbol{\xi})\in D\times[0,T]\times\Gamma. \tag{7}$$

For the sake of brevity, we shall denote the coupled system of Eqs. (5)–(7) as

$$\mathscr{B}(\boldsymbol{u},p,\theta:\boldsymbol{x},t,\boldsymbol{\xi})=0. \tag{8}$$

3. Generalized polynomial chaos expansions

In this approach, the dependent variables $(\boldsymbol{u},p,\theta)$ are regarded as random processes. These random fields are then expanded into a truncated series. One could use the Karhunen–Loève expansion to represent these fields, but unfortunately one does not a priori know the correlation functions for these variables. The generalized polynomial chaos expansion is used to represent the variables in terms of orthogonal polynomials in the stochastic space spanned by $\boldsymbol{\xi}\in\Gamma$. This is written as

$$\kappa(\boldsymbol{x},t,\omega)=\sum_{i=0}^{P}\kappa_i(\boldsymbol{x},t)\Phi_i(\xi^i(\omega)), \tag{9}$$

where κ is any of the dependent variables. Here κ_i are the deterministic coefficients that can be thought of as being similar to the coefficients in a Fourier expansion of a function. Note that the possible infinite summation has been truncated to only P terms. The random trial basis functions $\{\Phi_i\}$ are chosen according to the type of random variable $\{\xi^i\}$ that has been used to describe the input random field. For example, if Gaussian random variables are chosen then the Askey based orthogonal polynomials $\{\Phi_i\}$ are chosen to be Hermite polynomials, if ξ^i are chosen to be uniform random variables, then $\{\Phi_i\}$ must be Legendre polynomials (for a complete description of the GPCE scheme, the interested reader is referred to [18]). The total number of expansion terms is determined by the stochastic dimension (N) and the highest-order (p) of the orthogonal polynomials as follows:

$$P+1=\frac{(N+p)!}{N!p!}. \tag{10}$$

The GPCE is used to expand all the dependent variables in terms of the orthogonal polynomials. Substituting these expansions into the governing equation, Eq. (8), gives:

$$\mathscr{B}\left(\sum_{i=0}^{P}\boldsymbol{u}_i\Phi_i,\sum_{i=0}^{P}p_i\Phi_i,\sum_{i=0}^{P}\theta_i\Phi_i:\boldsymbol{x},t,\boldsymbol{\xi}\right)=0. \tag{11}$$

Following this, a Galerkin projection of the above equation onto each polynomial basis Φ_i is conducted to ensure that the error is orthogonal to the functional space spanned by the finite-dimensional basis Φ_i:

$$\left\langle\mathscr{B}\left(\sum_{i=0}^{P}\boldsymbol{u}_i\Phi_i,\sum_{i=0}^{P}p_i\Phi_i,\sum_{i=0}^{P}\theta_i\Phi_i:\boldsymbol{x},t,\boldsymbol{\xi}\right),\Phi_j\right\rangle=0, \tag{12}$$

where $\langle a,b\rangle$ is the inner product of the functions a and b over the ensemble ($\langle a,b\rangle=\int_\Omega ab\mathrm{d}\boldsymbol{x}$). By using the orthogonality of the polynomial basis, we can obtain a set of $(P+1)$ coupled equations for each random mode u_i,p_i,θ_i. By utilizing the polynomial chaos expansion followed by the Galerkin projection, the randomness has been transferred from the dependent variables to the basis polynomials. The resulting governing equations for the expansion coefficients are deterministic.

This methodology has been very successful in solving SPDEs [3,11–14,16,18,19,23]. We will restrict our discussion to the disadvantages of this method so as to motivate the next few sections. The coupled nature of the expansion coefficients makes the implementation of the GPCE method non-trivial. Substantial effort has to be put in to convert a validated deterministic code into a feasible stochastic one. As the number of stochastic dimensions increases, the rapidly growing number $(P+1)$ of basis functions effectively reduces the efficiency of the method. New faster and more efficient solvers and data processing methodologies are necessary for the efficient solution of higher-dimensional problems. The major constraint for the utility of this methodology is the coupled nature of the resulting equations. This resulted in the search for a method that combined the fast convergence and error estimates of the spectral Galerkin method with the decoupled nature of the more statistical methods like the Monte-Carlo methods.

4. Collocation methods

In the spectral Galerkin method described in the previous section, the spatial domain is approximated using a finite element discretization and the stochastic space is also approximated using a spectral element discretization. This representation of the stochastic space causes the coupled nature of the resulting equations. Another approach is to have a finite element approximation for the spatial domain and approximate the multi-dimensional stochastic space using interpolating functions. The interpolating functions are mutually orthogonal and the resulting equations are decoupled. This approach is called the collocation approach, where one computes the deterministic solution at various points in the stochastic space and then builds an interpolated function that best approximates the required solution. We now detail the issues involved in constructing the stochastic solution using the collocation method. We follow the spirit of the discussion in Xiu et al. [27] to give an intuitive development of the method.

4.1. Fundamentals of the collocation method

The basic idea of the collocation approach is to build an interpolation function for the dependent variables using their values at particular points in the stochastic space. The Galerkin projection is then applied to find that interpolation function which minimizes the projected error of the approximated system. Denote by $\boldsymbol{\xi}$ any point in the random space $\Gamma \subset \mathbb{R}^N$, by Π_N, the space of all N-variate polynomials and by Π_N^p, the subspace of polynomials of total degree at most p. The problem of interpolation can be stated as follows: Given a set of nodes $\Theta_N = \{\boldsymbol{\xi}_i\}_{i=1}^M$ in the N-dimensional random space Γ and the smooth function $f : \mathbb{R}^N \to \mathbb{R}$, find the polynomial $\mathscr{I}f$ such that $\mathscr{I}f(\boldsymbol{\xi}_i) = f(\boldsymbol{\xi}_i),\ \forall i = 1, \ldots, M$.

The polynomial approximation $\mathscr{I}f$ can be expressed using the Lagrange interpolation polynomials as follows:

$$\mathscr{I}f(\boldsymbol{\xi}) = \sum_{k=1}^{M} f(\boldsymbol{\xi}_k) L_k(\boldsymbol{\xi}), \tag{13}$$

where $L_i(\boldsymbol{\xi}_j) = \delta_{ij}$. Now, once the interpolating polynomial have been generated using the nodes $\{\Theta_N\}$, the value of the function at any point $\boldsymbol{\xi} \in \Gamma$ is approximately $\mathscr{I}f(\boldsymbol{\xi})$. The Lagrange interpolated values of $(\boldsymbol{u}, p, \theta)$, denoted by $(\hat{\boldsymbol{u}}, \hat{p}, \hat{\theta})$ are as follows:

$$(\hat{\boldsymbol{u}}(\boldsymbol{\xi}), \hat{p}(\boldsymbol{\xi}), \hat{\theta}(\boldsymbol{\xi})) = (\mathscr{I}\boldsymbol{u}(\boldsymbol{\xi}), \mathscr{I}p(\boldsymbol{\xi}), \mathscr{I}\theta(\boldsymbol{\xi})) = \sum_{k=1}^{M} (\boldsymbol{u}(\boldsymbol{\xi}_k), p(\boldsymbol{\xi}_k), \theta(\boldsymbol{\xi}_k)) L_k(\boldsymbol{\xi}). \tag{14}$$

Substituting this into the governing equation, Eq. (8), gives

$$\mathscr{B}\left(\sum_{i=1}^{M} \boldsymbol{u}(\boldsymbol{\xi}_i) L_i(\boldsymbol{\xi}), \sum_{i=1}^{M} p(\boldsymbol{\xi}_i) L_i(\boldsymbol{\xi}), \sum_{i=1}^{M} \theta(\boldsymbol{\xi}_i) L_i(\boldsymbol{\xi}) : x, t, \boldsymbol{\xi} \right) = 0. \tag{15}$$

The interpolation form of the solution immediately leads to M decoupled deterministic systems

$$\mathscr{B}(\boldsymbol{u}(\boldsymbol{\xi}_i), p(\boldsymbol{\xi}_i), \theta(\boldsymbol{\xi}_i) : \boldsymbol{x}, t, \boldsymbol{\xi}_i) = 0, \quad i = 1, \ldots, M. \tag{16}$$

The collocation method collapses the $(N + d)$-dimensional problem to solving M deterministic problems in d dimensions. The statistics of the random solution can then be obtained by

$$\langle z^\alpha(x) \rangle = \sum_{k=1}^{M} z^\alpha(x, \boldsymbol{\xi}_k) \int_\Gamma L_k(\boldsymbol{\xi}) \rho(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi}. \tag{17}$$

Remark 1 (*Relation to the spectral Galerkin method*). Babuska and co-workers [21,22] realized that by choosing appropriate basis functions, the so-called double orthogonal functions, the spectral stochastic Galerkin formulation can also be made decoupled. This is specific to the case when the random input is a linear function of the input random variables. They also showed that in case the input uncertainties are

multi-linear combinations of the random variables $\{\xi^i(\omega)\}$ (as would happen if one uses the K–L expansion to truncate the input random fields to a finite set of random variables), the solution obtained from the collocation procedure coincides with the stochastic Galerkin solution [26]. They argue that the collocation method can be seen as a pseudo-spectral Galerkin method with some distinct advantages.

4.2. Building the interpolating polynomial basis

In most problems involving input uncertainties, the final objective is usually to compute the moments of the dependent variables (Eq. (17)) and/or generate the probability distribution functions (PDFs) at some point in the solution phase space. This is accomplished through specific quadrature and sampling operations on the interpolation function. The analysis and construction of the interpolation function is usually introduced through a univariate interpolation formula.

Let $f : [a, b] \to \mathbb{R}$ be a function that has to be interpolated by a polynomial $\mathscr{I}_n(f)$ using a finite number of nodes $a \leqslant x_0 < x_1 < \cdots < x_n \leqslant b$. There exists a (unique) polynomial $\mathscr{I}_n(f) \in \Pi_1^n$ satisfying $\mathscr{I}_n(f)(x_i) = f(x_i)$ for $i = 0, 1 \ldots, n$. This can be written in the form:

$$\mathscr{I}_n(f)(x) = \sum_{i=0}^{n} f(x_i) L_i(x), \tag{18}$$

where the basis polynomials are given by

$$L_i(x) = \prod_{k=0, k\neq i}^{n} \frac{x - x_k}{x_i - x_k}. \tag{19}$$

As the number of points n increases, the interpolation function $\mathscr{I}_n(f)$ represents the function f better. This is irrespective of how one chooses the nodes $\{x_i\}_{i=0}^n$. However, uniform convergence ($\|f - \mathscr{I}_n(f)\|_\infty \to 0$ as $n \to \infty$) is not guaranteed for any arbitrary distribution of the nodes. Utilizing a Taylor series expansion, the interpolation error for a function $f \in \mathscr{C}^{n+1}[a, b]$ at a point x is given by [32]

$$f(x) - \mathscr{I}_n(f)(x) = \frac{f^{n+1}(\xi)}{(n+1)!} \prod_{j=0}^{n} (x - x_j). \tag{20}$$

The magnitude of the $(n + 1)$th derivative could outweigh the product of the nodes. But there are unique node distributions that exist where uniform convergence with the number of nodes can be proved if $f \in \mathscr{C}^1[a, b]$.

One way of quantifying the approximating quality of the interpolating polynomial for a given node distribution (denoted by X) is to compare it with a polynomial whose nodes have been selected in an optimal way to theoretically minimize the interpolation error. This *best approximation polynomial* $p_n^*(f) \in \Pi_1^n$, is defined as the polynomial with the following property [32]:

$$\|f - p_n^*(f)\|_\infty \leqslant \|f - \mathscr{I}(f)\|_\infty \quad \forall \mathscr{I}(f) \in \Pi_1^n. \tag{21}$$

The distance of any interpolation polynomial $\mathscr{I}(f)$ from the best approximation polynomial is measured in terms of the Lebesgue constant $\Lambda_n(X)$ as follows [32]:

$$\|f - \mathscr{I}(f)\|_\infty \leqslant \|f - p_n^*(f)\|_\infty (1 + \Lambda_n(X)), \tag{22}$$

where $\Lambda_n(X) = \max_{x\in[a,b]} \left(\sum_{i=0}^n |L_i(x)|\right)$ where X is any distribution of the nodes. It is apparent that $\Lambda_n(X)$ does not depend on the function f but in fact only depends on the distribution of the points X. Therefore it is possible to construct a priori, sets of nodes X with a small Lebesgue constant. One such type of node distribution is the interpolation based at the Chebyshev extrema [32].

Once the type of support nodes $\{X\}$ has been chosen and the Lebesgue constant for the node distribution computed, it is straightforward to arrive at error bounds for the polynomial interpolation function $\mathscr{I}^X$. For instance, the error bound for the Chebyshev node based interpolation function of the function $f \in \mathscr{C}^k$ is given by [32,34]

$$\|f - \mathscr{I}^X(f)\|_\infty \leqslant C n^{-k} \log(n). \tag{23}$$

4.3. From univariate interpolation to multivariate interpolation

When one is dealing with multiple stochastic dimensions, it is straightforward to extend the interpolation functions developed in one dimension to multiple dimensions as simple tensor products. If $u(\boldsymbol{\xi})$ is a function that has to be approximated in N-dimensional space, and $\mathbf{i} = (m_1, m_2, \ldots, m_N)$ are the number of nodes used in the interpolation in N dimensions, the full-tensor product interpolation formula is given as

$$\mathscr{I}^N u(\boldsymbol{\xi}) = (\mathscr{I}^{i_1} \otimes \cdots \otimes \mathscr{I}^{i_N})(u)(\boldsymbol{\xi}) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_N=1}^{m_N} u(\xi_{j_1}^{i_1}, \ldots, \xi_{j_N}^{i_N}).(L_{j_1}^{i_1} \otimes \cdots \otimes L_{j_N}^{i_N}), \tag{24}$$

where $\mathscr{I}^{i_k}$ are the interpolation functions in the i_k direction and $\xi_{j_m}^{i_k}$ is the j_mth point in the kth coordinate. Clearly, the above formula needs $m_1 \times \cdots \times m_N$ function evaluations, at points sampled on a regular grid. In the simplest case of using only two points in each dimension, the total number of points required for a full-tensor product interpolation is $M = 2^N$. This number grows very quickly as the number of dimensions is increased. Thus one has to look at intelligent ways of sampling points in the regular grid described by the full-tensor product formula so as to reduce the number of function evaluations required. This immediately leads to sparse collocation methods that are briefly reviewed next.

5. Sparse grid collocation methods

For the univariate case, Gauss points and Chebechev points have least interpolation error (for polynomial approximations). In the case of multivariate interpolation, one feasible methodology that has been used is to construct interpolants and nodal points by tensor products of one-dimensional interpolants and nodal points. An obvious disadvantage of this strategy is that the number of points required increases combinatorially as the number of stochastic dimensions is increased.

The Smolyak algorithm provides a way to construct interpolation functions based on a minimal number of points in multi-dimensional space. Using Smolyak's method, univariate interpolation formulae are extended to the multivariate case by using tensor products in a special way. This provides an interpolation strategy with potentially orders of magnitude reduction in the number of support nodes required. The algorithm provides a linear combination of tensor products chosen in such a way that the interpolation property is conserved for higher dimensions.

5.1. Smolyak's construction of sparse sets

Smolyak's algorithm provides a means of reducing the number of support nodes from the full-tensor product formula while maintaining the approximation quality of the interpolation formula up to a logarithmic factor. Consider the one-dimensional interpolation formula

$$\mathscr{U}^m(f) = \sum_{j=1}^{m} f(x_j) L_j. \tag{25}$$

Let us denote the set of points used to interpolate the one-dimensional function by $\Theta^{(k)}$. For instance, $\Theta^{(3)}$ could represent the Chebyshev points that interpolate a third-order polynomial. The Smolyak algorithm constructs the sparse interpolant $A_{q,N}$ (where N is the number of stochastic dimensions and q-N is the order of interpolation) using products of one-dimensional functions. $A_{q,N}$ is given as [31,32]

$$A_{q,N}(f) = \sum_{q-N+1 \leqslant |\mathbf{i}| \leqslant q} (-1)^{q-|\mathbf{i}|} \cdot \binom{N-1}{q-\mathbf{i}} \cdot (\mathscr{U}^{i_1} \otimes \cdots \otimes \mathscr{U}^{i_N}), \tag{26}$$

with $A_{N-1,N} = 0$ and where $\mathbf{i} = (i_1, \ldots, i_N)$ and $|\mathbf{i}| = i_1 + \cdots + i_N$. Here i_k can be thought of as the level of interpolation along the kth direction. The Smolyak algorithm builds the interpolation function by adding

a combination of one-dimensional functions of order i_k with the constraint that the sum total ($|\mathbf{i}| = i_1 + \cdots + i_N$) across all dimensions is between $q - N + 1$ and q. The structure of the algorithm becomes clearer when one considers the incremental interpolant, Δ^i given by [28,30–32]

$$\mathscr{U}^0 = 0, \quad \Delta^i = \mathscr{U}^i - \mathscr{U}^{i-1}. \tag{27}$$

The Smolyak interpolation $\mathscr{A}_{q,N}$ is then given by

$$\mathscr{A}_{q,N}(f) = \sum_{|\mathbf{i}| \leqslant q} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N})(f) = \mathscr{A}_{q-1,N}(f) + \sum_{|\mathbf{i}|=q} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N})(f). \tag{28}$$

To compute the interpolant $\mathscr{A}_{q,N}(f)$ from scratch, one needs to compute the function at the nodes covered by the sparse grid $\mathscr{H}_{q,N}$

$$\mathscr{H}_{q,N} = \bigcup_{q-N+1 \leqslant |\mathbf{i}| \leqslant q} (\Theta_1^{(i_1)} \times \cdots \times \Theta_1^{(i_N)}). \tag{29}$$

The construction of the algorithm allows one to utilize all the previous results generated to improve the interpolation (this is immediately obvious from Eq. (28)). By choosing appropriate points (like the Chebyshev and Gauss–Lobatto points) for interpolating the one-dimensional function, one can ensure that the sets of points $\Theta^{(i)}$ are nested ($\Theta^{(i)} \subset \Theta^{(i+1)}$). To extend the interpolation from level i to $i+1$, one only has to evaluate the function at the grid points that are unique to $\Theta^{(i+1)}$, that is, at $\Theta^i_\Delta = \Theta^i \setminus \Theta^{i-1}$. Thus, to go from an order $q-1$ interpolation to an order q interpolation in N dimensions, one only needs to evaluate the function at the differential nodes $\Delta\mathscr{H}_{q,N}$ given by

$$\Delta\mathscr{H}_{q,N} = \bigcup_{|\mathbf{i}|=q} (\Theta_\Delta^{i_1} \otimes \cdots \otimes \Theta_\Delta^{i_N}). \tag{30}$$

5.2. Interpolation error

As a matter of notation, the interpolation function used will be denoted $A_{N+k,N}$, where k is called the level of the Smolyak construction. The interpolation error using the Smolyak algorithm to construct multi-dimensional functions (using the piecewise multi-linear basis) is [31,32]

$$\|f - A_{q,N}(f)\| = \mathscr{O}(M^{-2}|\log_2 M|^{3(N-1)}), \tag{31}$$

where $M = \dim(\mathscr{H}(q,N))$ is the number of interpolation points. On the other hand, the construction of multi-dimensional functions using polynomial basis functions gives an interpolation error of [31,32]

$$\|f - A_{q,N}(f)\| = \mathscr{O}(M^{-k}|\log_2 M|^{(k+2)(N+1)+1}), \tag{32}$$

assuming that the function $f \in F_d^k$, that is, it has continuous derivatives up to order k.

5.3. Solution strategy

The final solution strategy is as follows: A stochastic collocation method in $\Gamma \subset \mathbb{R}^N$ along with a finite element discretization in the physical space $D \subset \mathbb{R}^d$ is used. Given a particular level of interpolation of the Smolyak algorithm in N-dimensional random space, we define the set of collocation nodes $\{\boldsymbol{\xi}_k\}_{k=1}^M$ on which the interpolation function is constructed. Given a piecewise FEM mesh X_d^h, find, for $k = 1, \ldots, M$, solutions

$$(\boldsymbol{u}_k^h(x), p_k^h(x), \theta_k^h(x)) = (\boldsymbol{u}^h(x, \boldsymbol{\xi}_k), p^h(x, \boldsymbol{\xi}_k), \theta^h(x, \boldsymbol{\xi}_k)), \tag{33}$$

such that

$$\mathscr{B}(\boldsymbol{u}^h(\boldsymbol{\xi}_i), p^h(\boldsymbol{\xi}_i), \theta^h(\boldsymbol{\xi}_i) : \boldsymbol{\xi}_i, x, t) = 0, \quad i = 1, \ldots, M. \tag{34}$$

The final numerical solution takes the form

$$z(x, \boldsymbol{\xi}) = \sum_{k=1}^{M} z_k(x) \hat{L}_k(\boldsymbol{\xi}), \tag{35}$$

where z is one of $\boldsymbol{u}^h$, p^h, θ^h and $\widehat{L}_k$ are the multi-dimensional interpolation functions constructed using the Smolyak algorithm.

Recently, rigorous error estimates for the collocation based stochastic methodology have been developed in [27 and 35]. The error is basically split into a superposition of errors: the error due to the discretization of the deterministic solution, ε_{D}, and the error due to the interpolation of the solution in stochastic space, ε_{I}. This error can be written as $\varepsilon \leqslant (C_1\varepsilon_{\mathrm{D}}^2 + C_2\varepsilon_I^2)^{1/2}$, where C_1 and C_2 are constants independent of the discretizations in physical space and stochastic space.

5.4. Numerical illustrations

In this subsection, we illustrate the sparse grid interpolation method with a few examples which will also serve as a motivation for the development of the adaptive sparse grid methodology.

5.4.1. Grid sizes

Fig. 1 shows the two-dimensional interpolation nodes for the sparse grid along with the two-dimensional tensor product grid based on the same one-dimensional interpolation nodes. As the number of dimensions or the level of interpolation increases, the number of points to construct interpolants with similar errors using the sparse grid collocation method reduces appreciably as compared to the full-tensor product collocation method.

5.4.2. Interpolating smooth functions

The sparse grid collocation method requires function evaluations at a finite number of discrete points θ. These function evaluations are usually finite element simulations of the deterministic problem at each stochastic point Θ_i. To illustrate the methodology, we will instead assume that certain analytical functions are the inputs at specific nodal points. Consider the function $f = \mathrm{e}^{-x^2-\varrho y^2}$, where ϱ is a positive number. This function is analytically smooth. For $\varrho = 1$, the function is isotropically smooth. The computational domain is $[-1,1]\times[-1,1]$. Fig. 2 compares the actual function and the interpolated function. The absolute error $\max|f - \mathscr{A}_{q,2}(f)|$ is 3×10^{-5} when considering 3329 points. This corresponds to building the sparse interpolant using one-dimensional interpolating functions up to order 9. To achieve a similar interpolation error using the full-tensor product grid would require 263,169 points.

Now we gradually change the value of ϱ to make the function more anisotropic. The interpolation error versus the number of nodes in the sparse grid is plotted in Fig. 3 for ϱ varying from 1 to 10,000. Notice that as the anisotropy increases, the variation of the function along one direction becomes progressively steeper

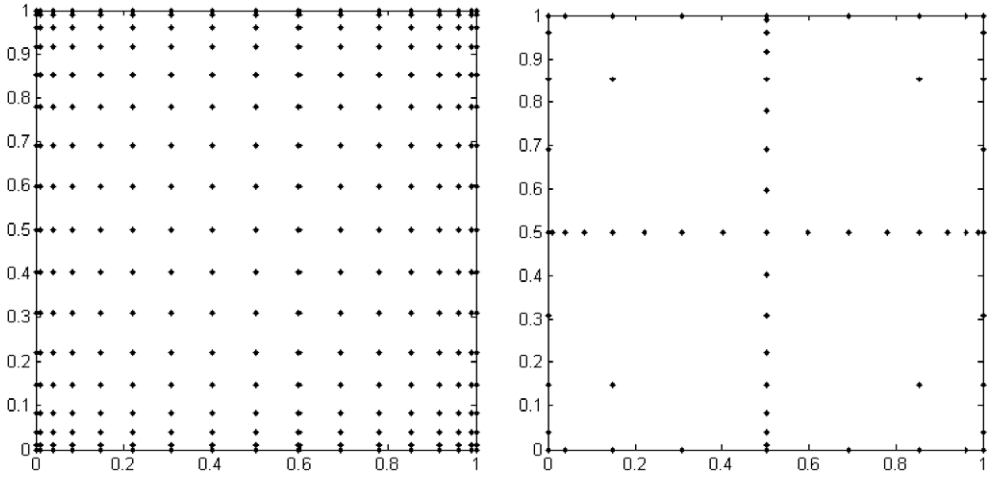


Fig. 1. Collocation nodes for a two-dimensional problem in a level 4 full (left) and sparse grid (right). Both the grids are constructed from the same level of one-dimensional interpolation functions. The sparse grid offers marginally worse interpolation properties with huge improvements in the number of points sampled.

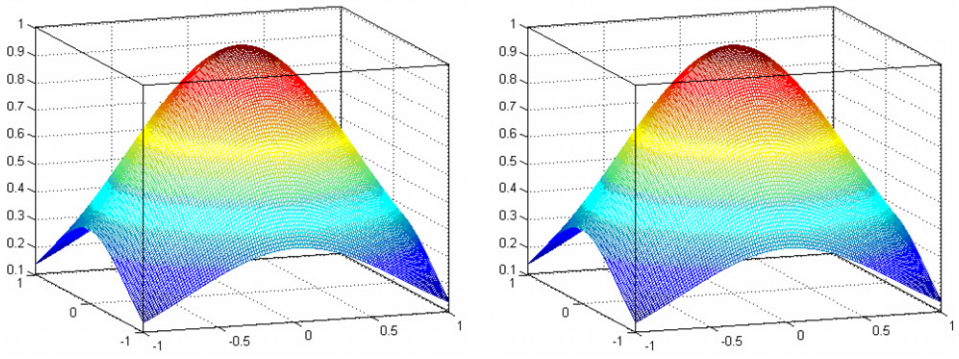


Fig. 2. Comparison of the exact (left) and interpolant (right) functions.

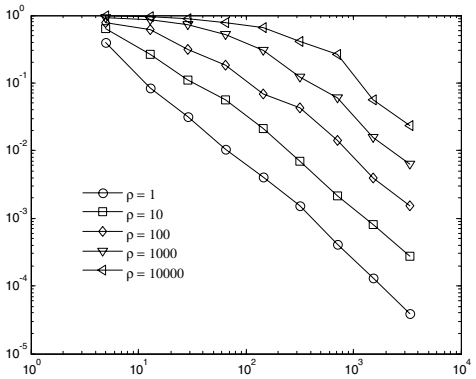


Fig. 3. Interpolation error for anisotropic functions: as ϱ increases, the function becomes steeper (Fig. 4). More points are required to accurately interpolate the function.

(see Fig. 4). Thus, more support nodes are required to accurately interpolate the function. To reach an accuracy of 2×10^{-2} the number of points required are 65, 145, 321, 705, 1537 and 3329, for $\rho = 1, 10, 100, 1000$ and 10,000, respectively.

The critical point to note here is that even though only one dimension of the function is very steep (as ϱ increases), the sparse grid increases the node allocation in all directions. This suggests that one can further reduce the number of function evaluations if we can detect the directions in which the function varies more smoothly than the others.

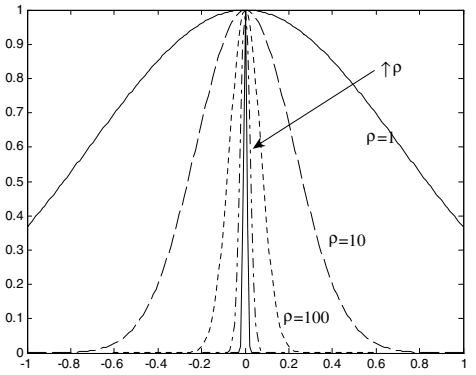


Fig. 4. Slice of the function in the x plane. The arrow shows the direction of increasing ϱ.

5.4.3. Interpolating discontinuous functions

In many physical systems governed by SPDEs, one is primarily interested in analyzing mode jumps and equilibrium shifts that occur due to small variations in the random variables. One extreme case is when there is a sharp (finite) discontinuity in the solution across a particular random variable. We are interested in the performance of the sparse grid collocation method in capturing this discontinuity.

Consider a function $f = e^{-x^2+2\text{sign}(y)}$ in the domain $[-1,1]\times[-1,1]$. Here sign is the signum function: 1, when y is positive and -1, otherwise. The function f has a sharp discontinuity along the $y=0$ plane. On the other hand, the function is very smooth in the x direction. That is, the function $f(x,c)$, for a constant c is an exponentially decaying function. The sparse grid method is used to build the interpolation for this function. Fig. 5 plots the error as a function of the number of support nodes used by the algorithm. Notice that the error reduction is sub-linear during the first few interpolation levels. This is because of the error caused by the jump across the x axis.

Fig. 6 plots the 'evolution' of the interpolation function as the number of support nodes increases. The first row shows a plot of the interpolated function while the second row shows the corresponding support nodes required to build that particular interpolation function. To construct a polynomial interpolation for the function f with an error in the range of 2×10^{-3} required 32,769 support nodes. But notice that the sparse grid adds points equally in both dimensions, even when the function varies very smoothly in one dimension.

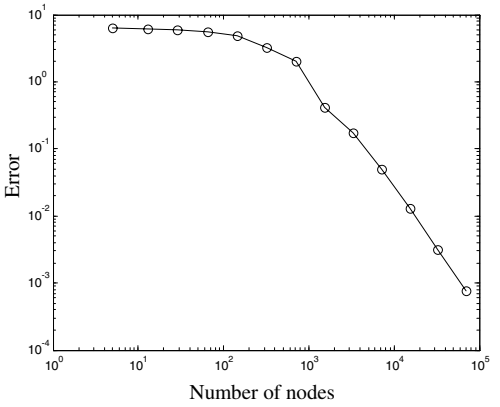


Fig. 5. Interpolation error for a discontinuous function.

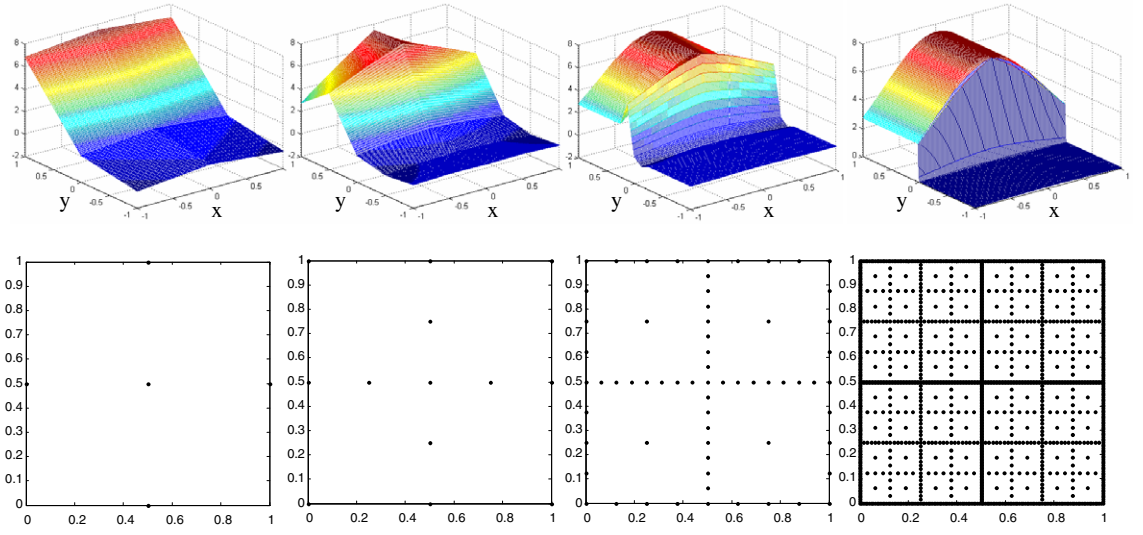


Fig. 6. Top row: Interpolated function at different levels of interpolation, Bottom row: The corresponding sparse grids used.

The two examples discussed above illustrate the possibility of further reducing the number of needed function evaluations for cases in which (i) the functional dependence on the stochastic variables is additive or nearly additive, (ii) the function is anisotropic in the stochastic dimensions. and (iii) when discontinuities exist in the function in certain dimensions.

The choice of support nodes now depends implicitly on the function that one is interpolating. Therefore, one must decide which nodes to sample on-the-fly for minimizing the number of function evaluations. This naturally leads to the idea of adaptive sparse grid collocation, which is discussed in the next section.

6. Adaptive sparse grid collocation methods

As stated earlier, in the standard sparse grid approach, all the stochastic dimensions are treated equally. By treated equally, we mean that the number of grid points in each direction is equal. In most physical problems that one deals with, there usually exists some structure (additive, nearly-additive, anisotropic, discontinuous) that can be taken advantage of, to reduce the number of function evaluations. However, the specific kind of structure that the particular solution exhibits is not known a priori. Thus one must construct an approach that automatically detects which dimensions require more nodal points.

The basic proposition of the adaptive sparse grid collocation method is to assess the stochastic dimensions differently, according to the error of interpolation in that dimension. We closely follow the dimension adaptive quadrature approach developed by Gerstner and Griebel [36] and further developed by Klimke [32]. The reason all dimensions are sampled equally in the standard sparse grid method is apparent by revisiting Eq. (28). The multi-index $|\mathbf{i}| = i_1 + \cdots + i_N$ that determines the number of sampling points in each dimension i_k is sampled from the simplex $|\mathbf{i}| \leqslant q$. This ensures that if $(n_1, n_2, \ldots, n_N)$ is a valid index, then $(n_2, n_1, \ldots, n_N)$ is also a valid index and so on. One simple approach to extend the conventional sparse grid to sample differently important dimensions is to consider a weighted simplex $\mathbf{a} \cdot \mathbf{i} \leqslant q$ (this simple extension is in fact not feasible [36]). Another approach is to allow any general index set satisfying certain conditions. The problem essentially boils down to choosing index sets from the set of conventional sparse grid points. One can heuristically improve the performance of the interpolant in directions where the function is varying rapidly.

6.1. Generalized sparse grid construction

The conventional Smolyak algorithm imposes a strict admissibility criterion on the index sets ($|\mathbf{i}| \leqslant q$). The incorporation of heuristically choosing the most important dimensions/directions/regions can be achieved by relaxing this strict admissibility criterion. The so-called generalized sparse grids [36] are obtained by relaxing this admissibility criteria.

An index set S is called admissible if for all indices $\mathbf{k} \in S$ [36]

$$\mathbf{k} - \mathbf{e}_j \in S \quad \text{for } 1 \leqslant j \leqslant N, \;\; k_j > 1, \tag{36}$$

where $\mathbf{e}_j$ is the jth unit vector. This means that an admissible set contains for every index $\mathbf{k}$ all indices which have smaller entries than $\mathbf{k}$. The generalized sparse grid includes both the conventional sparse grids as well as the classical product formulae within it. This criterion still ensures the telescopic property of the differential interpolant $\Delta^i = \mathscr{U}^i - \mathscr{U}^{i-1}$. This means that one can construct a better interpolant starting from a coarse interpolant by just sampling on the unique nodes of the finer interpolant, see Eq. (28). Denote the space of all admissible indices by $\mathscr{F}$. The generalized sparse grid construction is then [36]

$$\mathscr{A}_S(f) = \sum_{\mathbf{i} \in S} (\Delta^{i_1} \otimes \ldots \otimes \Delta^{i_1})(f) \tag{37}$$

for an admissible set $S \in \mathscr{F}$. By carefully constructing the index sets, it is possible to construct polynomials at least as good as the case of the conventional sparse grids.

6.2. Interpolation procedure

To construct interpolation polynomials of increasing accuracy by increasing the number of indices, one uses a nested sequence of index sets (as in the case of the conventional sparse grid method). The construction starts

with the one element index set $\mathbf{1} = (1, \ldots, 1)$. Indices are added successively such that two criteria are satisfied: the resulting index sets all remain admissible and a large reduction in the interpolation error is achieved. To achieve the second criterion, an error estimator has to be defined that quantifies the error reduction. Once a suitable error indicator is determined, the corresponding error indicators are computed in all the stochastic dimensions.

One starts the interpolation construction with the smallest index $\mathbf{1}$ and adds the indices $(2, \ldots, 1), \ldots, (1, \ldots, 2)$. One then obtains an initial set of error indicators. The index set is partitioned into two sets, the active indices $\mathscr{A}$ and the old indices $\mathscr{O}$ [36]. Newly added indices are denoted as active indices. The error indicators of the forward neighbors of active indices have not yet been computed. All other indices are denoted as old indices.

At each step, the forward neighborhood of the index set with the largest error indicator is considered, and the corresponding error indicators computed. The forward neighborhood of an index $\mathbf{i}$ are the N indices $\{\mathbf{i} + \mathbf{e}_j | j = 1, \ldots, N\}$. These new sets are added to the active index sets. Then the index set with the largest error indicator is considered and so on. The error indicator is chosen to be the deviation of the computed value of a particular point from its expected point (from the interpolation polynomial built at this level). This is just the differential interpolation operator $\sum \Delta^{i-1}(f)$ (see Eq. (28)) i.e. it is the difference between the computed value of the function at an active index and the interpolated value of the function using all the old indices.

In this way, the algorithm adaptively samples the regions of the conventional sparse grid support region where the error is maximized. The following subsection illustrates the effectiveness of this methodology with some sample examples.

6.3. Numerical illustrations

6.3.1. Interpolating anisotropic functions

We revisit the first problem in the sparse grid collocation section. Briefly, consider the function $f = \mathrm{e}^{-x^2-\varrho y^2}$, where ϱ is a positive number. This function is analytically smooth. For $\varrho = 1$, the function is isotropically smooth. The computational domain is $[-1, 1] \times [-1, 1]$. We look at the problem for increasing ϱ. As the anisotropy increases the adaptive procedure must sample more points in that direction where the error is maximum (the y direction in this particular case). Fig. 7 plots the adaptive grids used for the two particular cases of ϱ. Fig. 8 plots the error reduction with the number of support nodes. Compare this with the case of using the conventional sparse grids (Fig. 6). The number of nodes required to achieve the same interpolation error of 5×10^{-2} for the case of $\rho = 1000$ is 577 using the adaptive sparse grid method as compared to 1537 using the conventional sparse grid method. As the anisotropy of the problem increases, the adaptive sparse grid method selectively improves the function in the y direction, until the same level of accuracy is reached in both directions.

6.3.2. Interpolating discontinuous functions

We revisit the second problem discussed in the previous section. Briefly, consider a function $f = \mathrm{e}^{-x^2+2\mathrm{sign}(y)}$ in the domain $[-1, 1] \times [-1, 1]$, where sign is the signum function. The function f has a sharp discontinuity

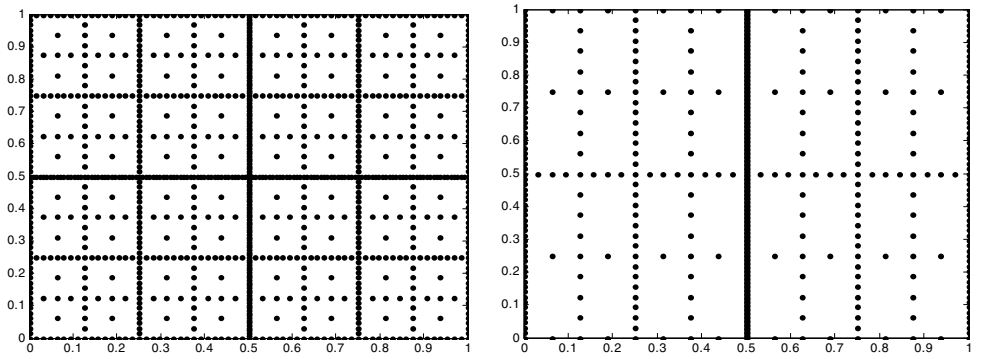


Fig. 7. The adaptive grid used for $\rho = 1$ (left) and $\rho = 1000$ (right). Notice the increased node points in the y direction for the second case.

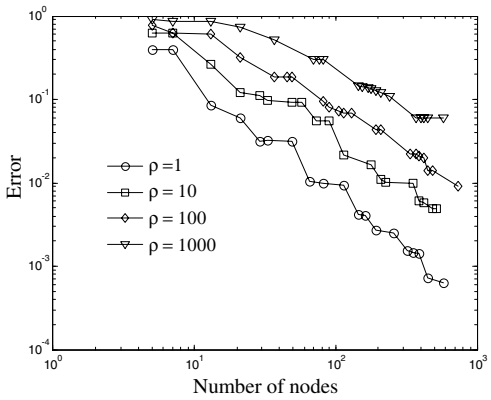


Fig. 8. Error versus the number of nodes using the adaptive sparse grid method for interpolating anisotropic functions. Compare with the error plot using the conventional sparse grid method shown in Fig. 3.

along the $y = 0$ plane. Fig. 9 shows the adaptive sparse grid used to compute the interpolation function for this discontinuous function. Notice the large bias towards the y direction, which is to be expected from an adaptive algorithm. Fig. 9 also plots the interpolated function. The error after using just 553 nodal points is 1×10^{-1}. To achieve the same level of error using the conventional sparse grid method (see Fig. 5) 3300 points have to be sampled.

The adaptive sparse grid collocation strategy – at worst – samples as many points as the conventional sparse grid collocation strategy. This is for problems where every stochastic dimension is equally important. In most physical problems encountered, the solutions exhibit some level of dissimilarity along various directions or have some structure that can be easily exploited using the adaptive sparse grid collocation method. Furthermore, there is very little overhead in computing the error indicator function (since the error indicators are just the incremental interpolation function at the new indices (Eq. (27)) [32]).

7. Ingredients for an adaptive sparse grid collocation implementation

In this section, we outline an approach to perform stochastic analysis of partial differential equations by non-intrusively using an existing deterministic code. The solution procedure can be broadly divided into three distinct operations:

- A subroutine for computing deterministic solutions.
- A subroutine for building the interpolation functions.
- A subroutine for computing moments and other post-processing operations.

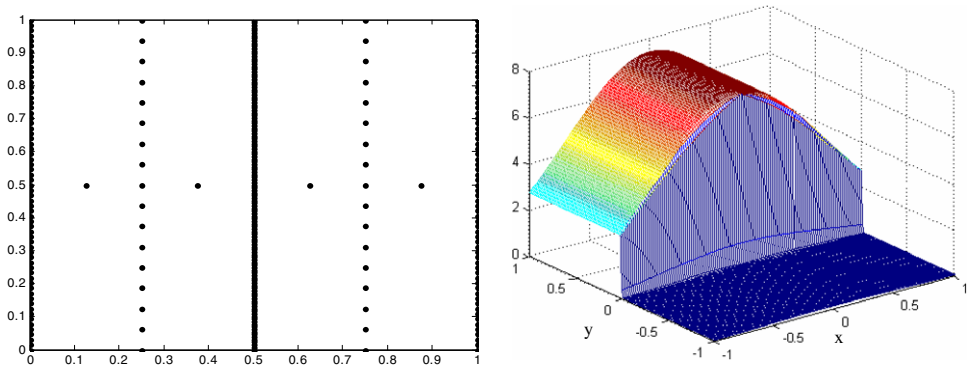


Fig. 9. Left: Adaptive mesh for the discontinuous problem, Right: Interpolated discontinuous functions.

7.1. Deterministic code

The concept of the collocation method lies in effectively decoupling the physical domain computation from the computations in the stochastic dimensions. The deterministic codes used in the present work took in as input, the sparse grid coordinates. For example, consider the case of two stochastic dimensions (ξ_1, ξ_2), that determine the input random process ω. The range of (ξ_1, ξ_2) is assumed to be $[0,1] \times [0,1]$ without loss of generality. The deterministic executable must be able to take in the different sampled two-tuples (ξ_1^i, ξ_2^i) and output logically named result files R_i.out.

7.2. Interpolation functions

In the present work, we use a simple MATLAB wrapper program that first initializes the stochastic dimensions and constructs the sparse grid coordinates. In the case of the conventional sparse grid collocation method, all the sampled nodes are known a priori. Hence in this case, the deterministic code is run and the appropriately named result files are available. The wrapper program then reads the input data and constructs the interpolation function. For the rapid construction of the interpolation functions, we utilize the sparse grid interpolation toolbox developed by Klimle [37,38]. In the case of the adaptive sparse grid collocation method, the wrapper program builds the input files for the deterministic code, calls the executable of the deterministic program and reads in the corresponding result at each step of the process.

7.3. Post-processing

Post-processing usually involves computing the moments of the stochastic solution across the whole domain. This is essentially a cubature of the interpolating function across the stochastic space. In the case of conventional sparse grid collocation, the integration is very straightforward because the weights corresponding to the known nodal positions are computed a priori. A data set containing the weights for each nodal position, for different dimensions and different interpolation depths was first created. Then the cubature over the stochastic space for simultaneously computing moments of various fields becomes simple scalar–matrix products. In the case of the adaptive sparse grid collocation method, the number of collocation points is not known a priori and hence the weights for each point are computed on the fly. That is at each level (or if a particular error tolerance has been reached), the interpolation functions are numerically integrated to get the corresponding weights associated with each nodal point. This results in significantly larger post-processing time for higher-dimensional integrations.

8. Numerical examples

In this section, we showcase the collocation based strategy to solving stochastic natural convection problems. In all the examples discussed in this work, we construct the interpolation functions based on Clenshaw–Curtis points [32]. The Clenshaw–Curtis type grid has been shown to outperform other types of grids in numerical tests [31,32]. In all the numerical examples, the global error computed is the maximum of the absolute variation of the interpolated value from the actual value at the newly computed sparse grid points $(\sup_i(\Delta^i))$. That is, at each level, the function is evaluated at some points. The error is defined as the difference between the interpolated value (using the interpolation function at the previous level) and the actual function value at these points. The global error reported is the maximum of these errors.

8.1. Natural convection with random boundary conditions

In the following example, a natural convection problem in a square domain $[-0.5, 0.5] \times [-0.5, 0.5]$ is considered. The schematic of the problem is shown in Fig. 10. The left wall is maintained at a higher temperature of 0.5. The right wall is maintained at a lower mean temperature of -0.5. The temperature at different points on the right hand side boundary is correlated. This is physically feasible through, say, a resistance heater, where the mean temperature remains constant, but material variations cause local fluctuations from this mean

Fig. 10. Schematic of a natural convection problem with random boundary temperature conditions.

temperature. We assume this correlation to be a simple exponential correlation function, $C(y_1, y_2) = \exp(-c \cdot |y_1 - y_2|)$. Here, c is the inverse of the correlation length that is taken as 1. For ease of analysis, we consider a non-dimensionalized form of the governing equations. The Prandtl number is set to 1.0 and the thermal Rayleigh number is set to 5000. The upper and lower boundaries are thermally insulated. No slip boundary conditions are enforced on all four walls.

Following the solution procedure in [3], the Karhunen–Loève expansion of the exponential correlation for the input uncertainty is performed. Truncating the infinite-dimensional representation of the boundary temperature to a finite-dimensional approximation gives

$$T(y, \boldsymbol{\xi}) = -0.5 + \sum_{i=0}^{M} \xi_i \sqrt{\lambda_i} f_i(y), \tag{38}$$

where ξ_i are normally distributed ($N(0,1)$). The eigenvalues of the correlation function decay very rapidly. The two largest eigenvalues contribute to 95% of the energy of the spectrum. To showcase the sparse grid collocation method, three cases, where the Karhunen–Loève expansion is truncated at 2, 4 and 8 terms, are considered.

The physical domain is discretized into 50×50 uniform quadrilateral elements. Each deterministic simulation is performed until steady-state is reached. This was around 600 time steps with a time interval of $\Delta t = 1 \times 10^{-3}$.

8.1.1. Convergence with increasing number of nodes

Fig. 11 plots the variation in the error with increasing number of support points for both the conventional isotropic sparse grid and the adaptive sparse grid methods. In the case of the isotropic sparse grid method, the level of interpolation is increased linearly for the three problems of dimensions 2, 4 and 8, respectively. For the two-dimensional case, the adaptive sparse grid collocation (ASGC) is marginally better than the conventional sparse grid collocation (CSGC) technique. The same level of accuracy is achieved in 137 function evaluations as compared to the 321 required in the CSGC method. As the number of stochastic dimensions is increased, the ASGC method provides orders of magnitude reduction in the number of functional evaluations. For the same magnitude of error ($\sim 4 \times 10^{-2}$), in four dimensions the ASGC required 751 function evaluations compared to the 7537 that the CSGC required (see Fig. 11). This is illustrated clearly in the case of eight stochastic dimensions where a 65 times reduction in the number of function evaluation is seen going from the CSGC to ASCG methods (15,713 to 241). What has to be noted is that this dramatic reduction in the number of functional evaluations is mainly due to the structure of the solution. While the adaptive strategy neglects the last four dimensions and quickly converges to an accurate solution, the isotropic sparse grid method takes many more functional evaluations to do the same because it treats all dimensions equally. As the number of dimensions (i.e. more terms in the KL expansion are taken into account) increases, the isotropic method treats these dimensions equally, even though the higher dimension have a negligible effect on the stochastic solution. On the other hand, the adaptive method takes this structure into account and preferentially samples the dimensions with the larger eigenvalues.

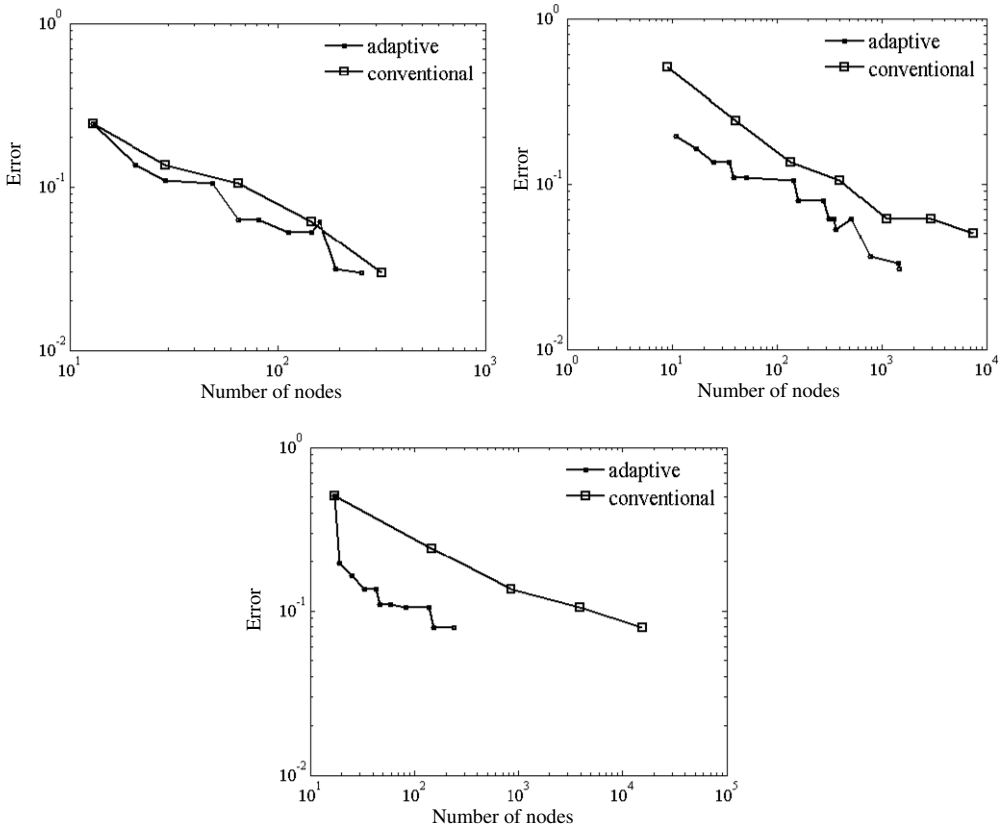


Fig. 11. Error reduction with increased number of support nodes for 2, 4 (top figures) and 8 (bottom figure) dimensions.

8.1.2. Comparison with Monte-Carlo simulations and GPCE

In the case of the conventional sparse grid collocation method for the problem truncated to two-dimensions, a level six interpolation scheme was used in the sparse grid interpolation method. This resulted in solving the deterministic problem at 321 different realizations. In the case of the adaptive sparse grid collocation method, 137 realizations were required. The sparse grid points correspond to the Clenshaw–Curtis cubature points in two-dimensions. The solution procedure is embarrassingly parallel. The complete simulation was performed independently on 16 nodes (32 processors) of the *V*3 cluster at the Cornell Theory Center. The total simulation time was about 8 min. To compare the results with other available methods, we performed Monte-Carlo simulations of the same process. 65,000 samples from the two-dimensional phase space were generated. A second-order polynomial chaos expansion of the above problem was also performed. The mean and standard deviation values of the three methodologies match very closely as seen in Figs. 12 and 13, respectively. The total computational time for the GPCE problem was 235 min. Table 1 shows representative times for the natural convection problem solved using GPCE and collocation based methods. The dimension 2 and 4 problems were solved using a second-order polynomial chaos representation, while the problem in eight dimensions was solved using a first-order polynomial chaos representation. All problems were solved on 16 nodes (32 processors) of the *V*3 cluster at the Cornell Theory Center. Notice that as the number of dimensions increases, the performance of the collocation method improves.

8.1.3. Higher-order moments and PDFs

Once the interpolant has been constructed, it is relatively straightforward to extract higher-order statistics from the stochastic solution. The effect of the uncertain boundary conditions is to change the convection patterns in the domain. At the extreme ranges, the direction of the convective roll actually reverses. It is informative to look at the probability distribution of the velocity at the points of high standard deviation. This

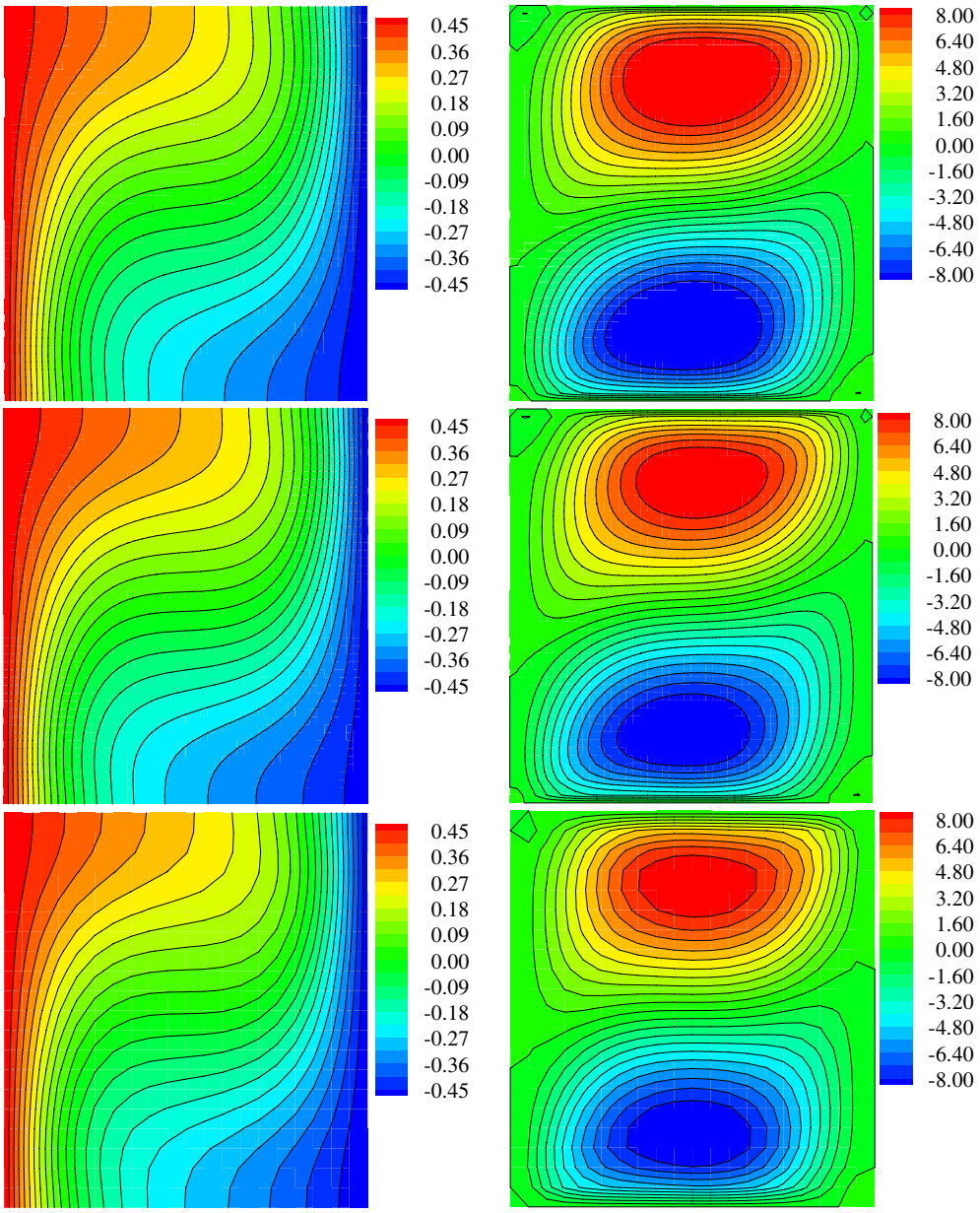


Fig. 12. Mean Temperature and u velocity contours from different solution strategies. Top row: A level 6 collocation method, Middle row: Monte-Carlo sampling over 65,000 points, Bottom row: Second-order GPCE expansion.

provides a clue towards the existence of equilibrium jumps and discontinuities in the field. The point $(0.34, 0.0)$ is one such point of high-deviation (see Fig. 13). The velocity at that point was sampled randomly 500,000 times from the two-dimensional stochastic space and the corresponding PDF from its histogram distribution was computed. The probability distribution functions for the temperature, pressure and velocities at this spatial point are plotted in Fig. 14. The applied random temperature boundary conditions results in some realizations having temperatures (along the right wall) both higher and lower than the left wall. This leads to instances where there is a complete reversal in the direction of circulation. Furthermore, as seen from Fig. 12, most of the flow in the horizontal direction is concentrated in two regions. Consequently, the u velocity in these regions experiences a large variation due to the imposed boundary conditions. This results in the

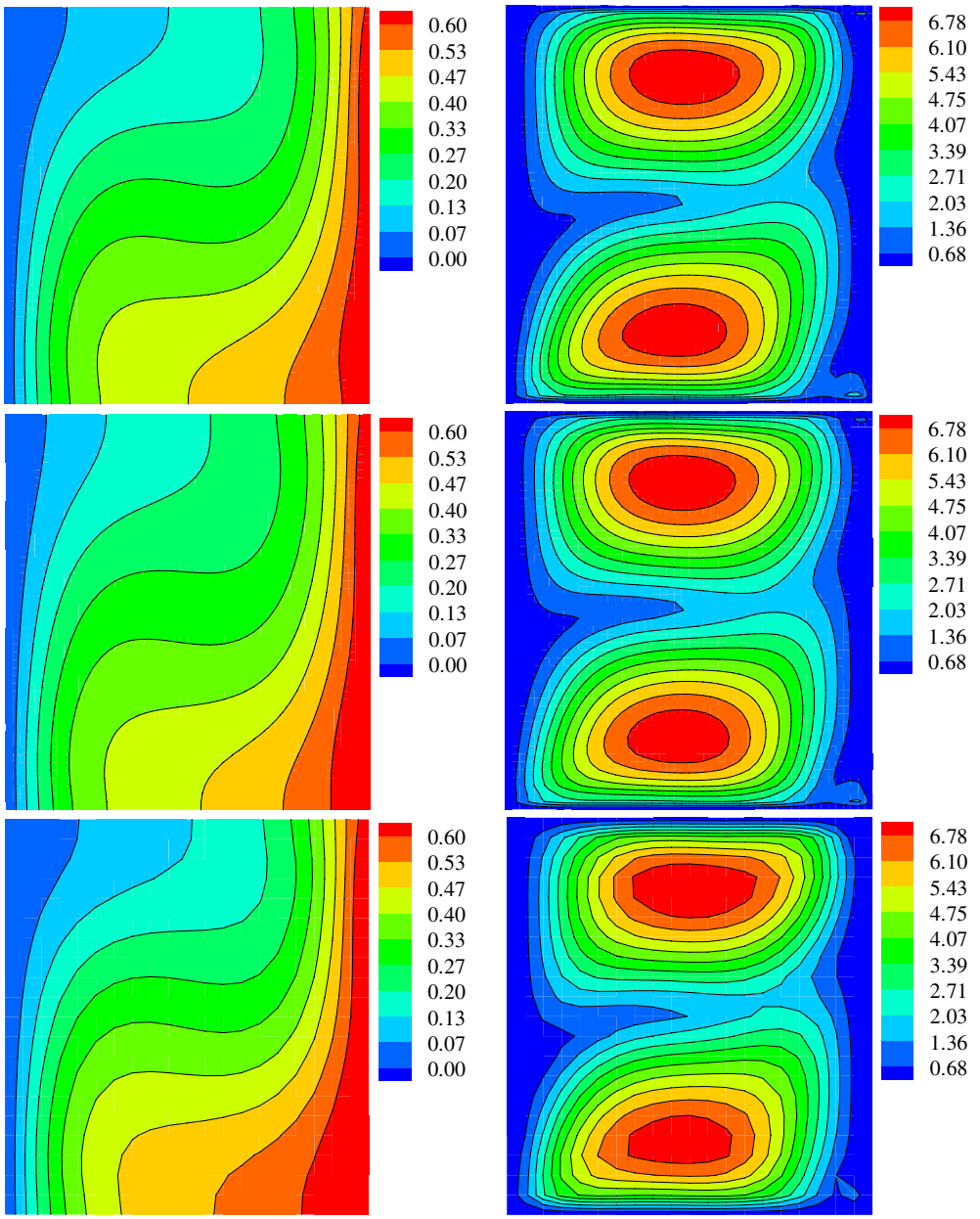


Fig. 13. Standard deviation of temperature and u velocity contours from different solution strategies. Top row: A level 6 collocation method, Middle row: Monte-Carlo sampling over 65,000 points, Bottom row: Second-order GPCE expansion.

tailing of the u velocity distribution seen in PDF for u velocity in Fig. 14. Since pressure can be considered to enter the physics as a means of imposing continuity, the PDF for the pressure also exhibits a similar tailing.

8.2. Natural convection with random boundary topology

In the previous problem, we modelled the uncertainty due to the randomness in the boundary conditions. In the present problem, we investigate the effects of surface roughness on the thermal evolution of the fluid heated from below. The surface of any material is usually characterized broadly into two aspects: waviness and roughness. Waviness refers to the large scale fluctuations in the surface. Roughness is the small scale perturbations of the surface. Roughness is described by two components: the probability of the point being z

Table 1
Solution times (min)

N (stochastic dims)	GPCE	Conventional SC	Adaptive SC
2	235	8	3.4
4	5340	187	19
8	1585	391	8

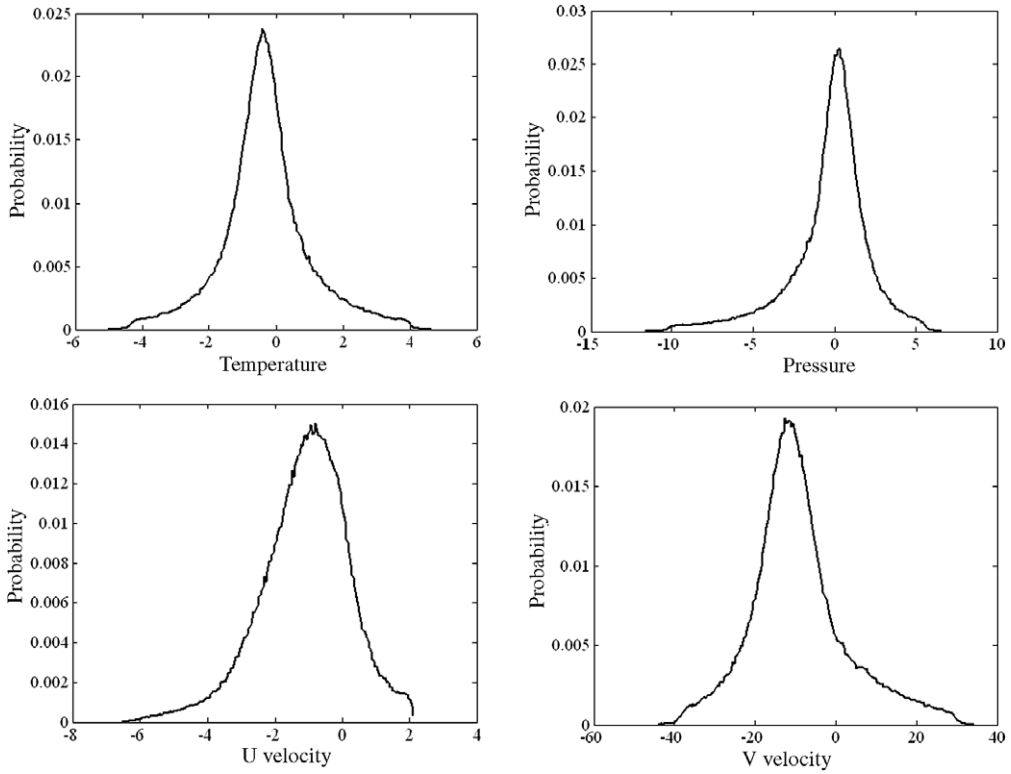


Fig. 14. Probability distribution functions for the dependent variables at point (0.34, 0). Top row: Temperature and pressure, Bottom row: u and v velocity components.

above/below the datum (PDF), and the correlation between two points on the surface (ACF). The type of surface treatment that resulted in the surface can be directly concluded from the auto-correlation function. For example, if the surface has been shot-peened, its ACF would have a larger correlation distance that if the surface had been milled. Usually the PDF of the position of a particular point is well represented by a Gaussian like distribution. To a degree, the ACF can be approximated by an exponential correlation kernel with a specified correlation length. Surface roughness has been shown to play an important role in many systems, particularly those involving turbulent flow [39]. A recent work of Tartakovsky and Xiu [40] deals with this problem in a spectral stochastic framework.

A natural convection problem in a rectangular domain $[-1.0, 1.0] \times [-0.5, 0.5]$ is considered. The schematic of the problem is shown in Fig. 15. The upper wall is smooth while the lower wall is rough. The ACF of the surface roughness of the lower wall is approximated to be an exponential function. The correlation length is set to 0.10. The mean roughness (absolute average deviation from a smooth surface) is taken to be $\frac{1}{100}$th of the characteristic length ($L = 1.0$). The first eight eigenvalues are considered to completely represent the surface randomness. The problem is solved in eight-dimensional stochastic space. The bottom wall is maintained at a higher temperature of 0.5. The top wall is maintained at a lower temperature of -0.5. The side walls are thermally insulated. The Prandtl number is set to 6.4 (corresponding to water) and the thermal Rayleigh number is set to 5000. No slip boundary conditions are enforced on all four walls. The physical domain is discretized into

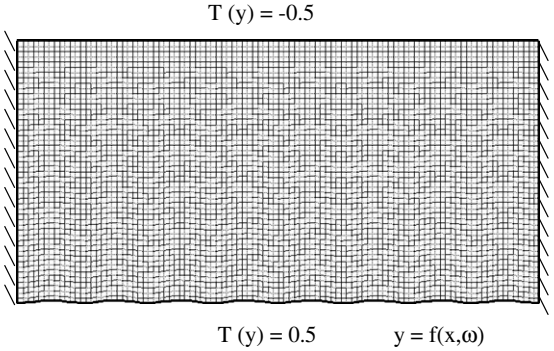


Fig. 15. Schematic of natural convection with random boundary topology (surface roughness).

100×50 uniform quadrilateral elements. Each deterministic simulation is performed until steady-state is reached. This was around 400 time steps with a time interval of $\Delta t = 1 \times 10^{-2}$.

Each deterministic simulation represents one realization of the random (rough) lower wall sampled at different $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_8)$. The lower wall is characterized by the truncated KL expansion, $y(x) = \sum_{i=1}^{8} \sqrt{\lambda_i}\xi_i f_i(x)$, where λ_i and $f_i(x)$ are the eigenvalues and eigenvectors of the ACF. A reference domain ($[-1.0, 1.0] \times [-0.5, 0.5]$) and grid ($100 \times 50$ uniform quadrilateral elements) are considered and the nodes on the bottom boundary are transformed according to the above formula to get each realization of the random boundary. The Jacobian of the transformation from the reference grid to the new grid can be easily computed (it is just the ratio of the elemental areas before and after the transformation). The deterministic solutions at each collocation point in stochastic space is computed on the reference grid. This allows the subsequent computation of the ensemble averages naturally.

In the case of the conventional sparse grid collocation method, a level 4 sparse interpolation grid was utilized for discretizing the stochastic space. 3937 collocation points were used to discretize the eight-dimensional stochastic space using the conventional sparse grid collocation method (CSCG) while the adaptive sparse grid collocation method (ASGC) method required 820 realizations to construct the interpolation function. Eight nodes on the Velocity-3 cluster at the Cornell Theory Center took 500 min for simulating the problem at all the collocation points. Fig. 16 plots the reduction in the error with the number of nodal supports used.

Fig. 17 shows some sample realizations of the temperature and velocity contours. The first row of figures is the case when the roughness is minimal. The temperature and velocity contours closely resemble those of Rayleigh–Bénard convection with smooth boundaries. The next four rows of figures show extremal values of the surface roughness. The thermal plume changes position according to the local surface geometry. Also note that the flow field has been substantially enhanced due to roughness. Experiments have shown that surface

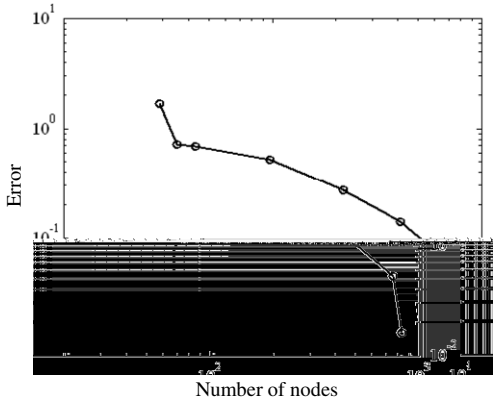
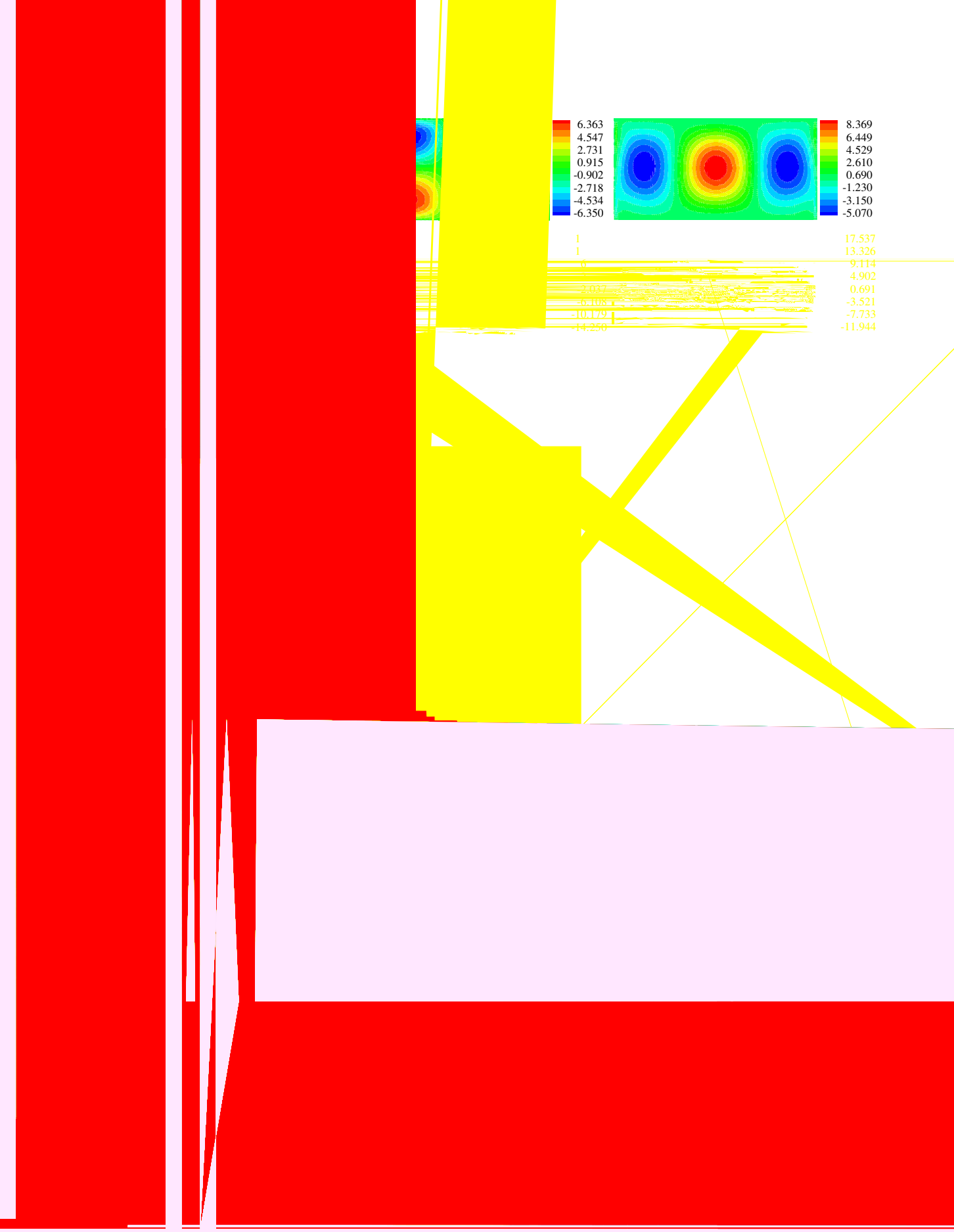


Fig. 16. Error reduction using the adaptive sparse grid methodology.


6.363
4.547
2.731
0.915
-0.902
-2.718
-4.534
-6.350
8.369
6.449
4.529
2.610
0.690
-1.230
-3.150
-5.070
1
1
6
2.077
-6.108
-10.179
-14.250
17.537
13.326
9.114
4.902
0.691
-3.521
-7.733
-11.944

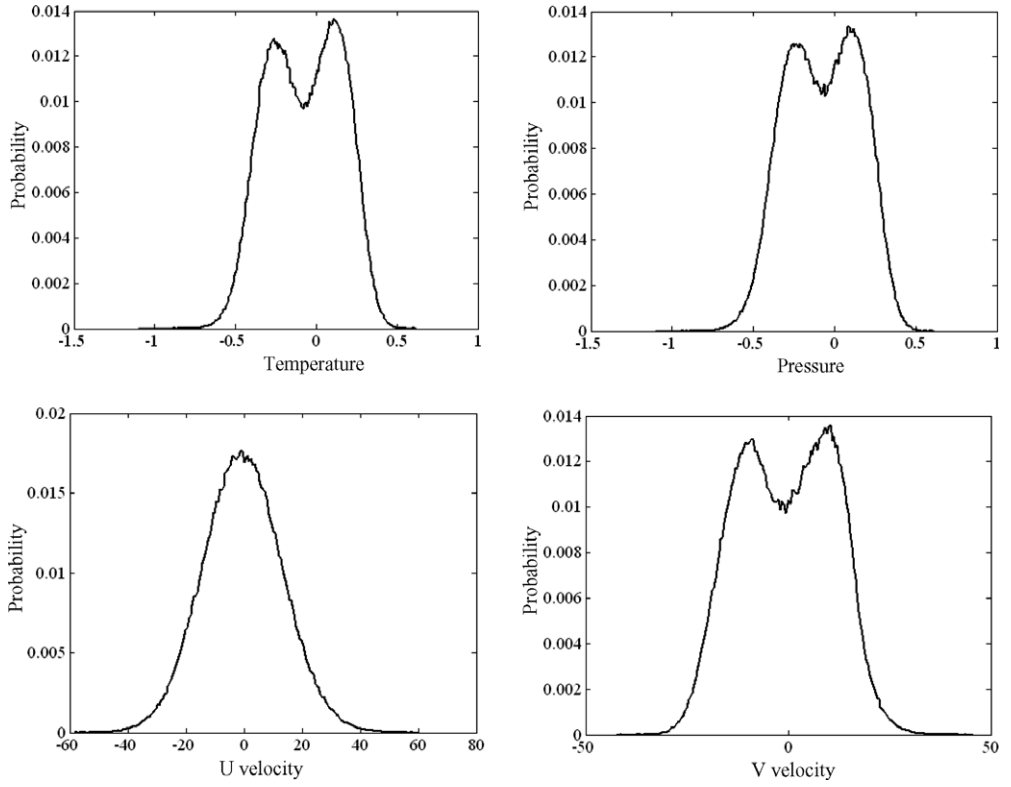


Fig. 21. Probability distribution functions for the dependent variables at point (0.25, 0). Top row: Temperature and pressure, Bottom row: u and v velocity components.

the non-linearities introduced by the surface roughness. To verify if this is indeed the case, Fig. 22 shows the temperature and v velocity variations at a specific point ($[0.0, 0.25]$, chosen because of its large temperature and v velocity second moment) plotted over the first two random variables. Notice the abrupt jump in the temperature and v velocity as the random variables cross 0. Fig. 23 shows the projection of the variation onto a single dimension. The mode shift is apparent in the abrupt change of the dependent variables as the random variable crosses over zero.

8.2.3. Below the critical Rayleigh number

As an extension to the previous problem, we investigate the possible mode shifts from a conduction based state to a convection based (Rayleigh–Bénard flow) thermal pattern caused by the effect of surface roughness.

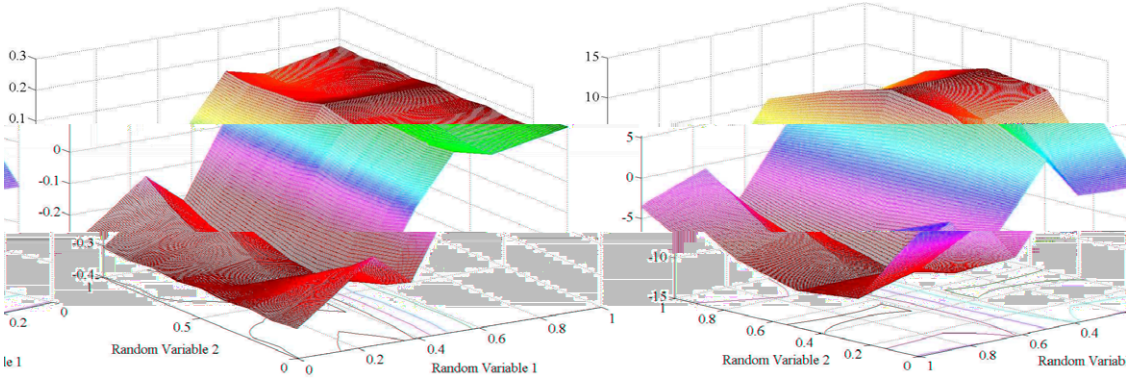
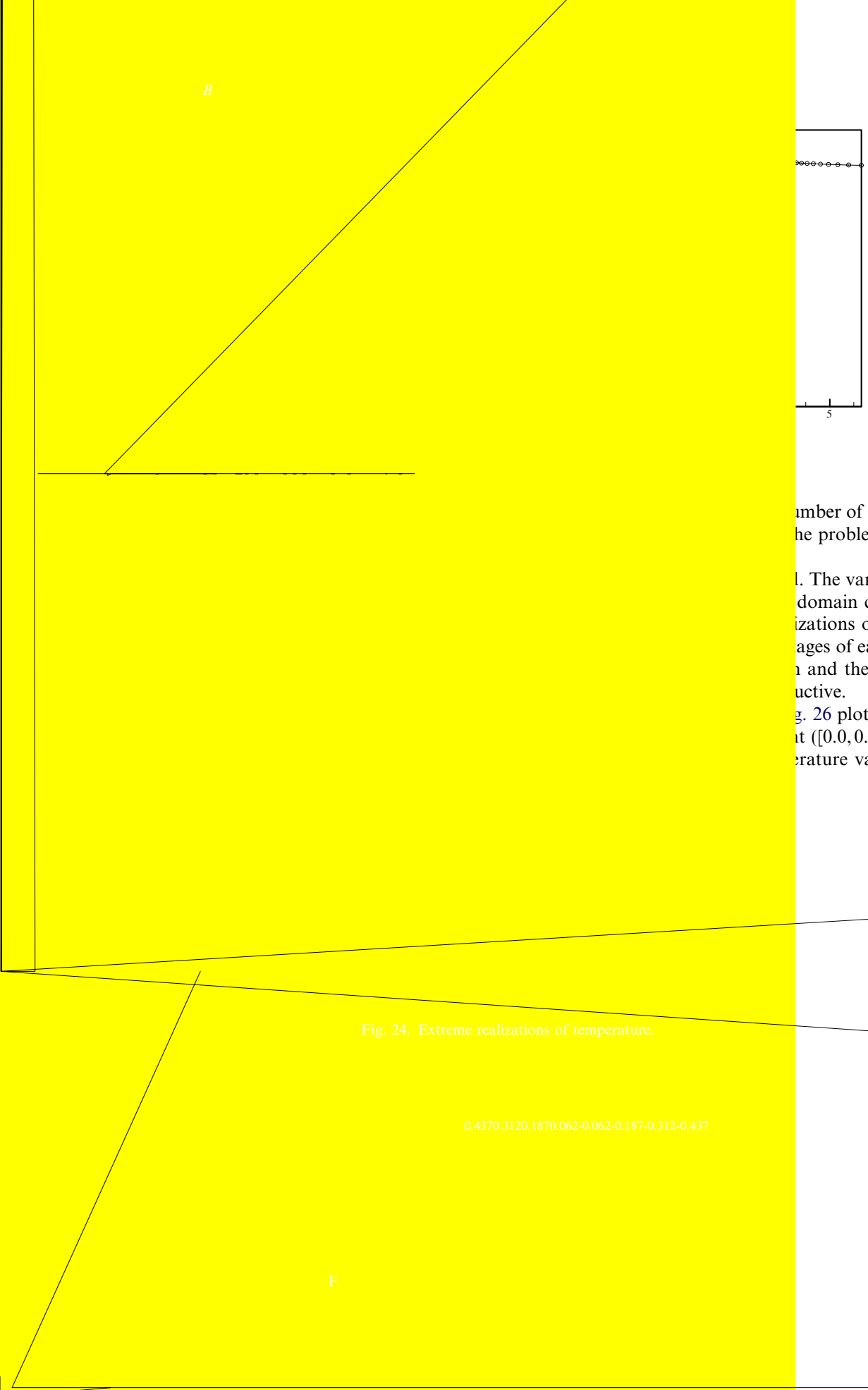


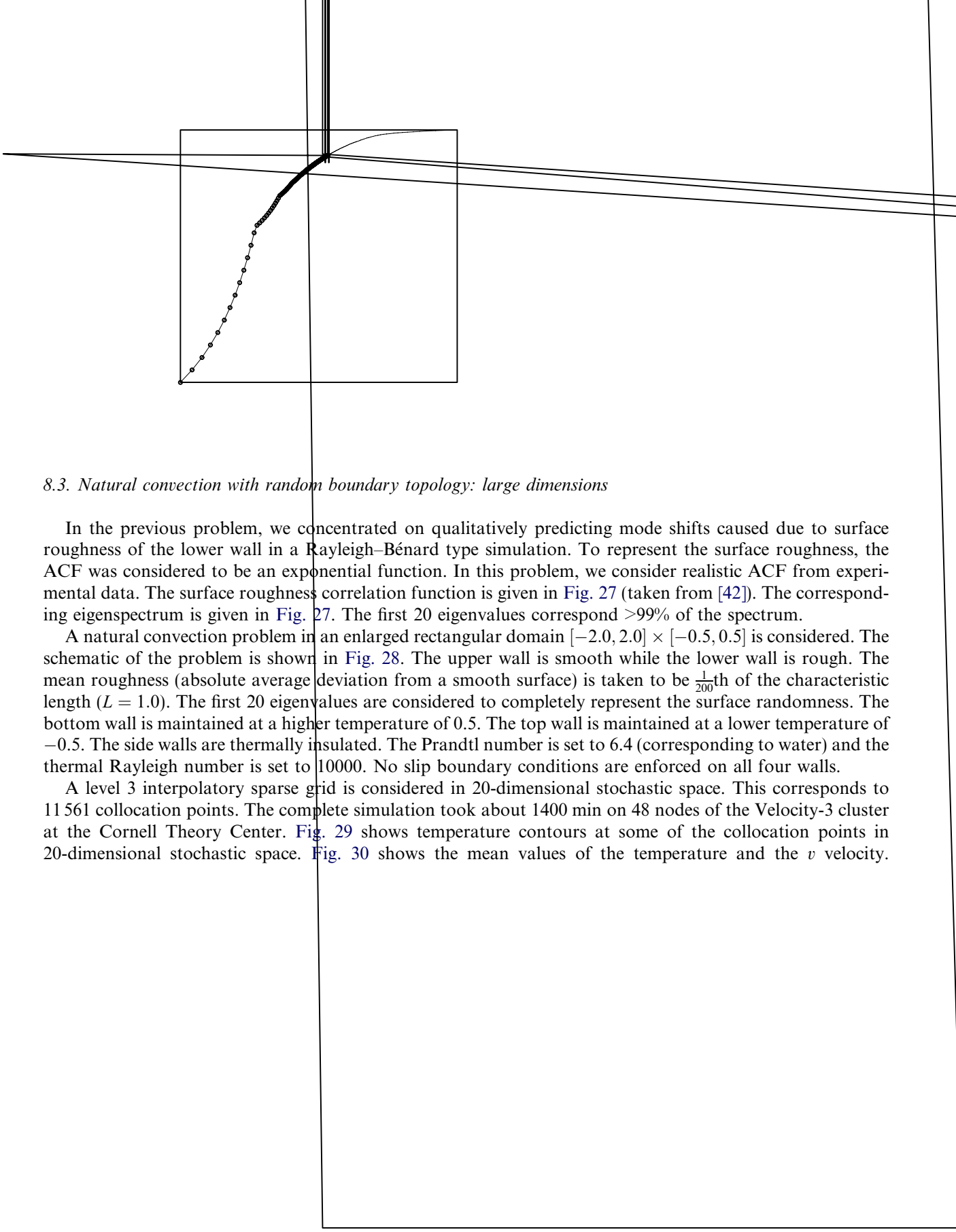
Fig. 22. Mode shift: Left: Temperature, Right: v velocity component.



umber of 1708. The
he problem in four

l. The variations in
domain causes the
izations of temper-
ages of each other.
n and the standard
uctive.
g. 26 plots the pro-
nt ([0.0, 0.0]). The v
erature variation is

Fig. 24. Extreme realizations of temperature.



8.3. Natural convection with random boundary topology: large dimensions

In the previous problem, we concentrated on qualitatively predicting mode shifts caused due to surface roughness of the lower wall in a Rayleigh–Bénard type simulation. To represent the surface roughness, the ACF was considered to be an exponential function. In this problem, we consider realistic ACF from experimental data. The surface roughness correlation function is given in Fig. 27 (taken from [42]). The corresponding eigenspectrum is given in Fig. 27. The first 20 eigenvalues correspond >99% of the spectrum.

A natural convection problem in an enlarged rectangular domain $[-2.0, 2.0] \times [-0.5, 0.5]$ is considered. The schematic of the problem is shown in Fig. 28. The upper wall is smooth while the lower wall is rough. The mean roughness (absolute average deviation from a smooth surface) is taken to be $\frac{1}{200}$th of the characteristic length ($L = 1.0$). The first 20 eigenvalues are considered to completely represent the surface randomness. The bottom wall is maintained at a higher temperature of 0.5. The top wall is maintained at a lower temperature of -0.5. The side walls are thermally insulated. The Prandtl number is set to 6.4 (corresponding to water) and the thermal Rayleigh number is set to 10000. No slip boundary conditions are enforced on all four walls.

A level 3 interpolatory sparse grid is considered in 20-dimensional stochastic space. This corresponds to 11 561 collocation points. The complete simulation took about 1400 min on 48 nodes of the Velocity-3 cluster at the Cornell Theory Center. Fig. 29 shows temperature contours at some of the collocation points in 20-dimensional stochastic space. Fig. 30 shows the mean values of the temperature and the v velocity.

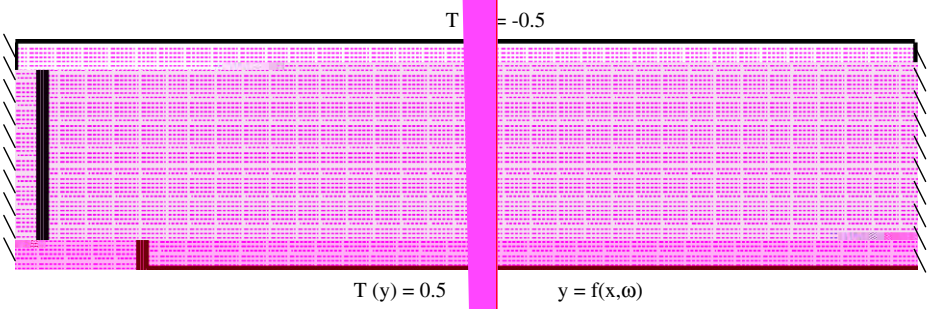


Fig. 28. Schematic of natural convection with wa urface roughness computed from experiments.

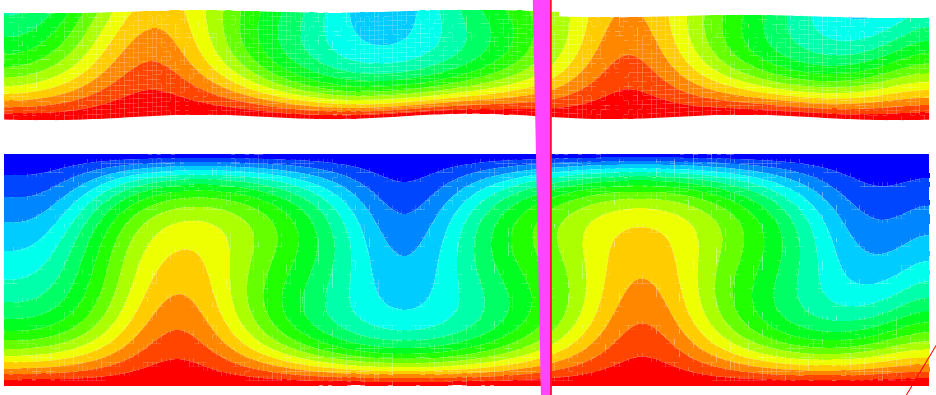
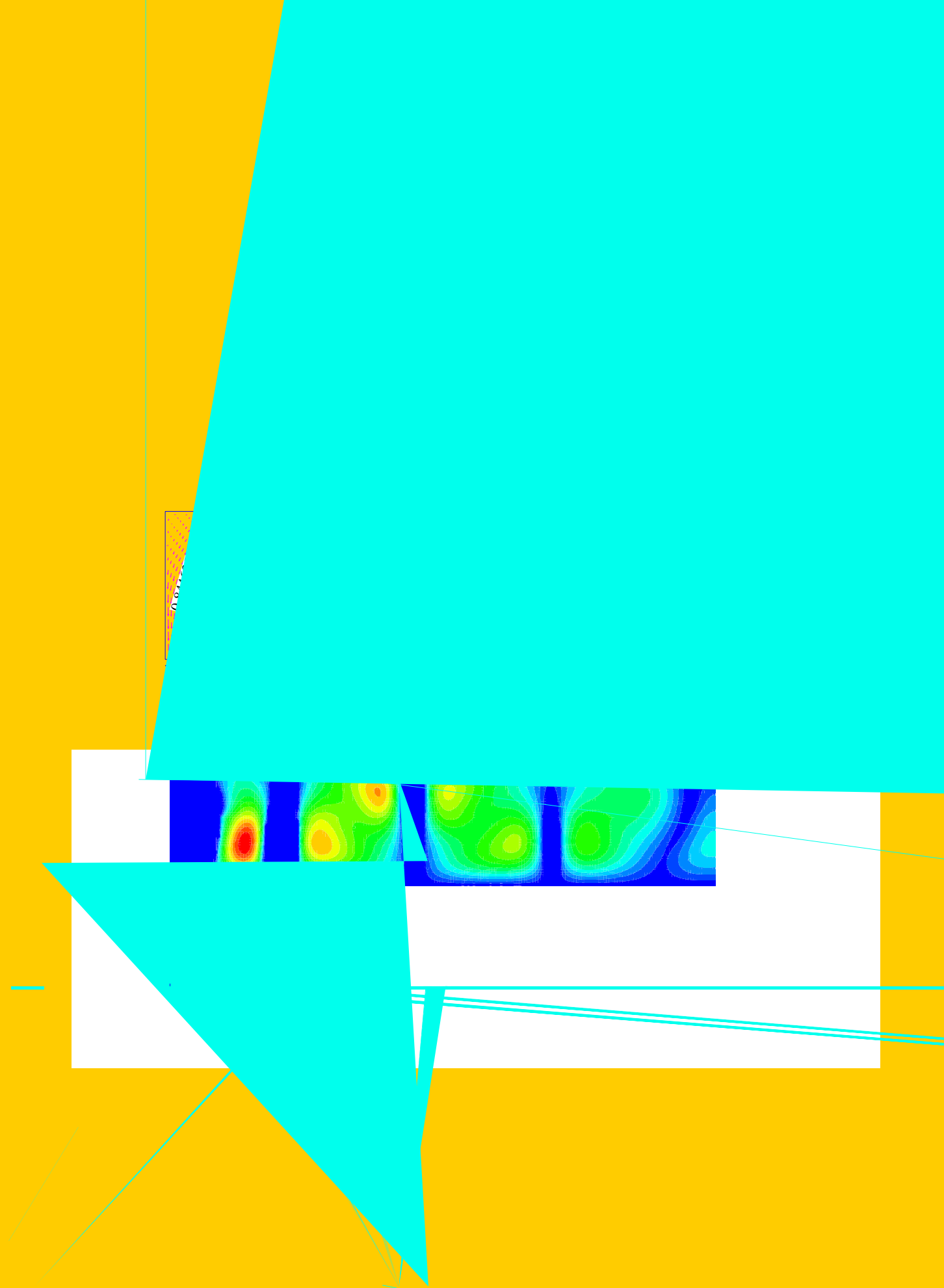


Fig. 31 plots the mean velocity vectors and some streaml nes. The standard deviations of the temperature and the v velocity are shown in Fig. 32. Notice that the surface roughness causes the mal plumes to form, which are seen in the variations in the temperature and the v velocity. These plumes cau by the non-uniformities in the lower surface result in improved heat transfer across the domain. This con can be further extended to the design of the ACF of the surface roughness to achieve enhanced heat tra characteristics. This will be showcased in one of our forthcoming publications.

8.4. Convection in heterogeneous porous media

Fluid flow through porous media is an ubiquitous process occurri ious sca scale: geothermal energy systems, oil recovery, to much smaller sc f liqui growth in alloy solidification and flow through fluidized beds. T flow



A schematic of the problem is shown in Fig. 33. This problem has been investigated by Nithiarasu et al. [43]. A square domain of dimensions $[-0.5, 0.5] \times [-0.5, 0.5]$ is considered. The inner half of the square is considered to be free fluid. The rest of the domain is filled with a porous material. The porous material is assumed to be Fontainebleau sandstone, whose experimental correlation function is taken from [44]. The correlation function and a cross-sectional image of the sandstone is shown in Fig. 34.

The governing equation for the fluid flow in the variable porosity medium is given by (see [43,45]):

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla\left(\frac{\boldsymbol{v}}{\epsilon}\right) = -\frac{Pr}{Da}\frac{(1-\epsilon)^2}{\epsilon^2}\boldsymbol{v} - \frac{1.75\|\boldsymbol{v}\|(1-\epsilon)}{(150Da)^{1/2}\epsilon^2}\boldsymbol{v} + Pr\nabla^2\boldsymbol{v} - \epsilon\nabla p - \epsilon PrRaT\mathbf{e}_g, \tag{39}$$

where ϵ is the volume fraction of the fluid and Da is the Darcy number. Natural convection in the system is considered. The left wall is maintained at a dimensionless temperature 1.0, while the right wall is maintained at 0.0. The other walls are insulated. No slip conditions are enforced on all four walls. The Rayleigh number is 10,000 and the wall Darcy number is 1.185×10^{-7}. The wall porosity is $\epsilon = 0.8$ and the porosity of the free fluid in the interior of the domain is 1.0.

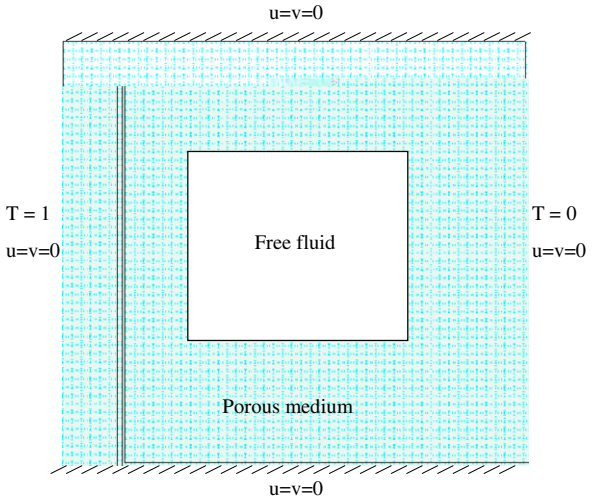


Fig. 33. Schematic of convection in a medium with random porosity.

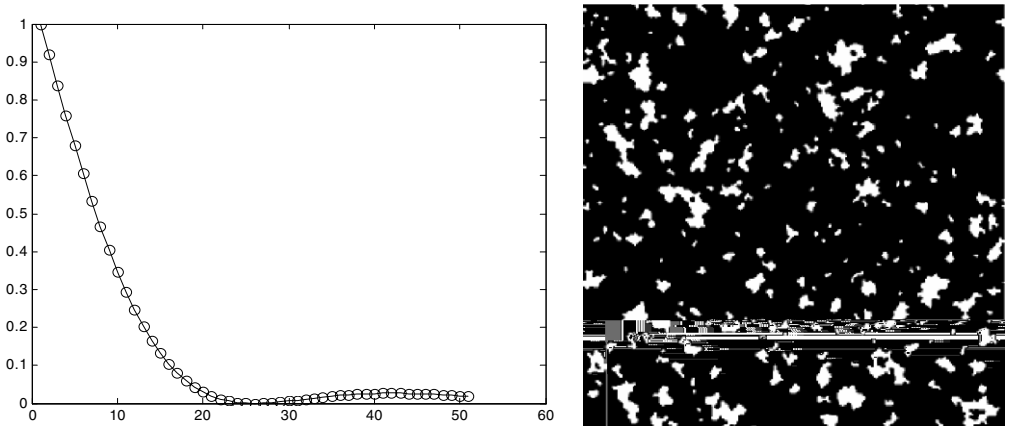


Fig. 34. Fontainebleau sandstone: correlation function and cross-sectional image.

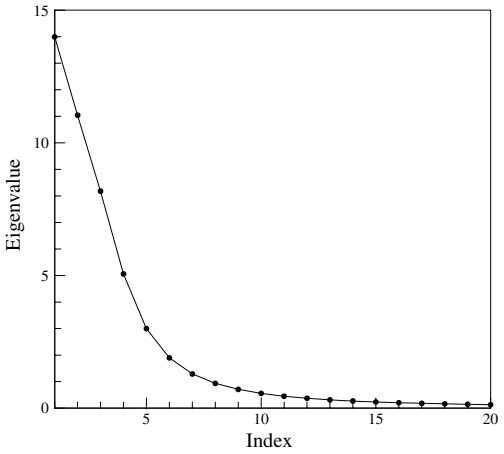


Fig. 35. Eigenspectrum of the correlation kernel.

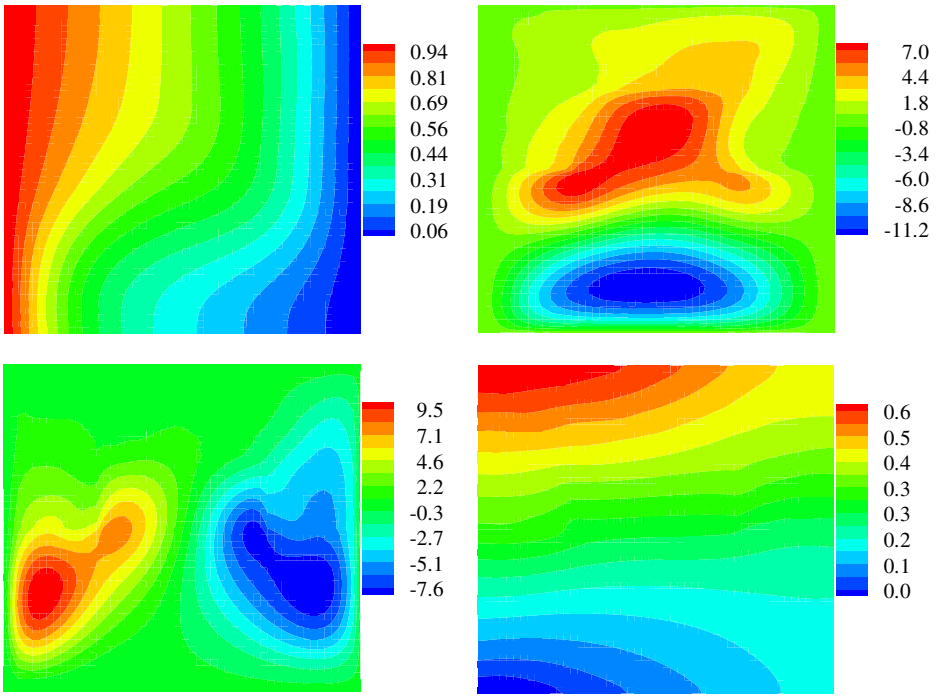


Fig. 36. Mean values of the variables: Top left: Temperature, Top right: u velocity component, Bottom left: v velocity component, Bottom right: pressure.

The numerically computed eigenvalues of the correlation are shown in Fig. 35. The first eight eigenvalues correspond to 94% of the spectrum. We consider the porosity to described in a eight-dimensional stochastic space. The mean porosity is 0.8. A level 4 interpolatory sparse grid with 3937 collocation points was utilized for the stochastic simulation. A fractional time step method based on the work of Nithiarasu et al. [43] is used in the simulation of the deterministic problem. Fig. 36 plots the mean temperature, velocity and pressure contours. Fig. 37 shows the mean velocity vectors and some streamlines. The standard deviation of the dependent variables is shown in Fig. 38.

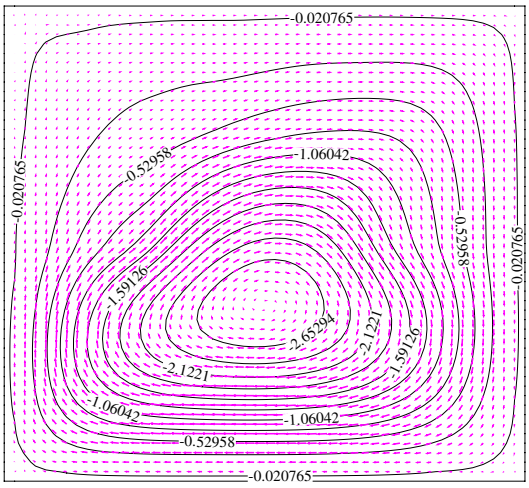
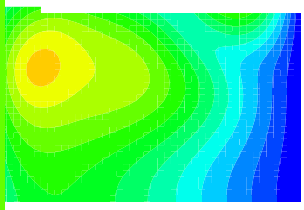


Fig. 37. Mean velocity vectors and streamlines.



9. Conclu

The g polynomial chaos approach and its derivatives have been extensively utilized to solve thermal and problems involving random inputs and boundary conditions. The GPCE is restrictive in the sense tha es computationally infeasible to solve problems involving large number of expansion terms due to th nature of the governing equations. In this context collocation based approaches are considered. ttempted to provide a natural development of collocation based approaches starting from the draw he spectral stochastic methodology. We show that this leads naturally to the tensor product

collocation method [26] and the sparse grid collocation method [27]. An extension of the sparse grid collocation method to adaptively sample regions/dimensions of high function variations is presented. The methodology can also efficiently detect additive and nearly additive structures in the solution. The adaptive sparse grid collocation methodology provides at least an order of magnitude improvement in the number of function evaluations required to construct comparable interpolation functions as compared to the conventional sparse grid method. The collocation methodology results in the solution of completely decoupled deterministic problems. The embarrassingly parallel nature of these problems is exploited to construct fast and efficient solution strategies to problems in high stochastic dimensions. Using the adaptive sparse grid collocation method, it is relatively straightforward to convert any deterministic code into a code that solves the corresponding stochastic problem. We provide a detailed road-map, description of the necessary changes in the deterministic code and the required subroutines for pre- and post-processing to easily accomplish this change.

We have attempted to showcase the utility of the sparse grid collocation method to solve large stochastic dimensional heat transfer problems. We have solved realistic natural convection problems in rough surfaces with stochastic dimensions up to $d = 20$ as well as natural convection problems through heterogeneous porous media. We further looked at the shifts in equilibrium that occur due to slight variations in the boundary conditions and successfully captured these equilibrium jumps.

We are currently investigating the solution of stochastic inverse design problems utilizing the decoupled nature of the resulting deterministic partial differential equations.

Acknowledgments

This research was supported by the Computational Mathematics program of AFOSR (Grant F49620-00-1-0373). The computing was conducted using the resources of the Cornell Theory Center, which receives funding from Cornell University, New York State, federal agencies and corporate partners.

References

[1] F.F. Abraham, J.Q. Broughton, N. Bernstein, E. Kaxiras, Spanning the length scales in dynamic simulation, Comput. Phys. 12 (1998) 538–546.
[2] D. Frenkel, B. Smit, Understanding Molecular Simulations: From Algorithms to Applications, Academic Press, 2002.
[3] R.G. Ghanem, P.D. Spanos, Stochastic Finite Elements: A Spectral Approach, Dover publications, 1991.
[4] C.L. Winter, D.M. Tartakovsky, Mean flow in composite porous media, Geophys. Res. Lett. 27 (2000) 1759–1762.
[5] C.L. Winter, D.M. Tartakovsky, Groundwater flow in heterogeneous composite aquifers, Water Resour. Res. 38 (2002) 23.1–23.11.
[6] C.L. Winter, D.M. Tartakovsky, A. Guadagnini, Numerical solutions of moment equations for flow in heterogeneous composite aquifers, Water Resour. Res. 38 (2002) 13.1–13.8.
[7] C.L. Winter, D.M. Tartakovsky, A. Guadagnini, Moment equations for flow in highly heterogeneous porous media, Surv. Geophys. 24 (2003) 81–106.
[8] R. Ghanem, Probabilistic characterization of transport in heterogeneous porous media, Comput. Methods Appl. Mech. Eng. 158 (1998) 199–220.
[9] R. Ghanem, A. Sarkar, Mid-frequency structural dynamics with parameter uncertainty, Comput. Methods Appl. Mech. Eng. 191 (2002) 5499–5513.
[10] R. Ghanem, Higher order sensitivity of heat conduction problems to random data using the spectral stochastic finite element method, ASME J. Heat Transfer 121 (1999) 290–299.
[11] D. Xiu, G.E. Karniadakis, Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos, Comput. Methods Appl. Mech. Eng. 191 (2002) 4927–4948.
[12] D. Xiu, G.E. Karniadakis, Modeling uncertainty in flow simulations via generalized polynomial chaos, J. Comp. Phys. 187 (2003) 137–167.
[13] D. Xiu, G.E. Karniadakis, A new stochastic approach to transient heat conduction modeling with uncertainty, Int. J. Heat Mass Transfer 46 (2003) 4681–4693.
[14] M. Jardak, C-H. Su, G.E. Karniadakis, Spectral polynomial chaos solutions of the stochastic advection equation, J. Sci. Comput. 17 (2002) 319–338.
[15] D. Lucor, C-H. Su, G.E. Karniadakis, Karhunen–Loève representation of periodic second-order autoregressive processes, Int. Conf. Comput. Sci. (2004) 827–834.
[16] X. Wan, G.E. Karniadakis, An adaptive multi-element generalized polynomial chaos method for stochastic differential equations, J. Comp. Phys. 209 (2005) 617–642.
[17] X. Wan, D. Xiu, G.E. Karniadakis, Stochastic solutions for the two-dimensional advection–diffusion equation, SIAM J. Sci. Comput. 26 (2005) 578–590.

[18] D. Xiu, G.E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations, SIAM J. Sci. Comput. 24 (2002) 619–644.
[19] B. Velamur Asokan, N. Zabaras, Variational multiscale stabilized FEM formulations for transport equations: stochastic advection–diffusion and incompressible stochastic Navier–Stokes equations, J. Comp. Phys. 202 (2005) 94–133.
[20] M.K. Deb, I.K. Babuska, J.T. Oden, Solution of stochastic partial differential equations using the Galerkin finite element techniques, Comput. Methods Appl. Mech. Eng. 190 (2001) 6359–6372.
[21] I. Babuska, R. Tempone, G.E. Zouraris, Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation, Comput. Methods Appl. Mech. Eng. 194 (2005) 1251–1294.
[22] I. Babuska, R. Tempone, G.E. Zouraris, Galerkin finite elements approximation of stochastic finite elements, SIAM J. Numer. Anal. 42 (2004) 800–825.
[23] D. Xiu, D. Lucor, C.-H. Su, G.E. Karniadakis, Performance evaluation of generalized polynomial chaos, international conference on computational science, Lecture Notes Comput. Sci. 2660 (2003) 346–354.
[24] R. Tempone, Numerical Complexity Analysis of Weak Approximation of Stochastic Differential Equations, Ph.D. Thesis, 2002.
[25] B. Velamur Asokan, N. Zabaras, Using stochastic analysis to capture unstable equilibrium in natural convection, J. Comp. Phys. 208 (2005) 134–153.
[26] I. Babuska, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data, ICES Report 05-47, 2005.
[27] D. Xiu, J.S. Hesthaven, High order collocation methods for the differential equation with random inputs, SIAM J. Sci. Comput. 27 (2005) 1118–1139.
[28] E. Novak, K. Ritter, The curse of dimension and a universal method for numerical integration, in: G. Nurnberger, J.W. Schmidt, G. Walz (Eds.), Multivariate Approximation and Splines, 1997, pp. 177–188.
[29] T. Gerstner, M. Griebel, Numerical integration using sparse grids, Numer. Algor. 18 (1998) 209–232.
[30] E. Novak, K. Ritter, R. Schmitt, A. Steinbauer, On an interpolatory method for high dimensional integration, J. Comp. Appl. Math. 112 (1999) 215–228.
[31] V. Barthelmann, E. Novak, K. Ritter, High dimensional polynomial interpolation on sparse grids, Adv. Comput. Math. 12 (2000) 273–288.
[32] A. Klimke. Uncertainty Modeling using Fuzzy Arithmetic and Sparse Grids, Ph.D. Thesis, Universität Stuttgart, Shaker Verlag, Aachen, 2006.
[33] C. Desceliers, R. Ghanem, C. Soize, Maximum likelihood representation of stochastic chaos representations from experimental data, Int. J. Numer. Met. Eng. 66 (2006) 978–1001.
[34] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods: Fundamentals in Single Domains Series, Springer, 2006.
[35] D. Xiu, Efficient collocational approach for parametric uncertainty analysis, Commun. Comput. Phys. 2 (2007) 293–309.
[36] T. Grestner, M. Griebel, Dimension adaptive tensor product quadrature, Computing 71 (2003) 65–87.
[37] A. Klimke, B. Wohlmuth, Algorithm 847: *spinterp*: piecewise multilinear hierarchical sparse grid interpolation in *MATLAB*, ACM Trans. Math. Software 31 (2005).
[38] A. Klimke, Sparse Grid Interpolation Toolbox – User's Guide, IANS report 2006/001, University of Stuttgart, 2006.
[39] Y. Shen, P. Tong, K.-Q. Xia, Turbulent convection over rough surfaces, Phys. Rev. Lett. 76 (1996) 908–911.
[40] D.M. Tartakovsky, D. Xiu, Stochastic analysis of transport in tubes with rough walls, J. Comput. Phys. 217 (2006) 248–259.
[41] E. Bodenschatz, W. Pesch, G. Ahlers, Recent developments in Rayleigh–Bénard convection, Annu. Rev. Fluid Mech. 32 (2000) 709–778.
[42] H. Li, K.E. Torrance, An experimental study of the correlation between surface roughness and light scattering for rough metallic surfaces, in: Advanced Characterization Techniques for Optics, Semiconductors, and Nanotechnologies II, in: Angela Duparré, Bhanwar Singh, Zu-Han Gu (Eds.), Proceedings of SPIE, vol. 5878, SPIE, Bellingham, WA, 2005.
[43] P. Nithiarasu, K.N. Seetharamu, T. Sundararajan, Natural convective heat transfer in a fluid saturated variable porosity medium, Int. J. Heat Mass Transfer 40 (1997) 3955–3967.
[44] C. Manwart, R. Hilfer, Reconstruction of random media using Monte Carlo methods, Phys. Rev. E 59 (1999) 5597–5600.
[45] D. Samanta, N. Zabaras, Modeling melt convection in solidification processes with stabilized finite element techniques, Int. J. Numer. Methods Eng. 64 (2005) 1769–1799.