SANDIA REPORT

2013-5517
Unlimited Release
Printed July, 2013

Spatially Varying Embedded Stochastic Galerkin Methods for Steady-State PDEs

Eric C. Cyr

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

2013-5517
Unlimited Release
Printed July, 2013

Spatially Varying Embedded Stochastic Galerkin Methods for Steady-State PDEs

Eric C. Cyr

Abstract

Existing discretizations for stochastic PDEs, based on a tensor product between the deterministic basis and the stochastic basis, treat the required resolution of uncertainty as uniform across the physical domain. However, solutions to many PDEs of interest exhibit spatially localized features that may result in uncertainty being severely over or under-resolved by existing discretizations. In this report, we explore the mechanics and accuracy of using a spatially varying stochastic expansion. This is achieved through an adaptive refinement algorithm where simple error estimates are used to independently drive refinement of the stochastic basis at each point in the physical domain. Results are presented comparing the accuracy of the adaptive techinque to the accuracy achieved using uniform refinement.

Acknowledgment

I am grateful to the the Sandia National Laboratories early career LDRD program for sponsoring this effort. I was aided by the guidance of my ECLDRD mentor Tim Trucano. I also received encouragement and advice from Brian Adams, Mike Eldred, Rob Hoekstra, John Jakeman, Eric Phipps, John Shadid, Laura Swiler, and Tim Wildey.

Contents

List of Figures

Chapter 1

Introduction

Uncertainty quantification (UQ) has been recognized as an essential computational technique in the validation of simulation software. A key aspect of UQ is forward propagation of uncertainty through a model to output Quantities-of-Interest (QoI). Abstractly, for a model M with *input parameters* p and *output responses* u (succinctly stated as $u = M(p)$) the forward propagation problem is to characterize the uncertainty in u given that the parameters p are drawn from the probability space $(\Gamma, \mathcal{F}, \rho)$, where Γ is the sample space, $\mathcal{F}$ is the σ-algebra and ρ is the probability density function (p.d.f.). In this report the model M is assumed to be deterministic, that is it does not introduce additionally uncertainty beyond what is prescribed by the input parameters. The nature of the characterization of the uncertainty in u is somewhat up to the application and propagation methodology. This report attempts to develop a smooth representation of the response u as a function of p, while maintaing accuracy where the p.d.f. ρ is significant.

For this report the model of interests are all steady-state partial differential equations (PDE) of the form

$$\mathcal{L}(x, p; u(x, p)) = f(x, p) \quad \text{for } (x, p) \in \Omega \times \Gamma \tag{1.1}$$

where $\mathcal{L}$ is a nonlinear differential operator, Ω is the physical domain, and u is the solution [9, 17]. Notice that the operator and source function are parameterized by p from the sample space Γ. Additionally, the solution u is implicitly a function of this parameter. In this case the input parameters are $p \in \Gamma$, the output response is the PDE solution u, and the model is implicitly defined by the PDE. If the sample space is endowed with a probability measure then moments of the output response can be computed, for example:

$$\mathbb{E}\left[u(x, \cdot)\right] = \int_\Gamma u(x, p)\rho(p)\, dp \quad \text{and} \quad \mathbb{V}ar\left[u(x, \cdot)\right] = \int_\Gamma \left(u(x, p) - \mathbb{E}\left[u(x, \cdot)\right]\right)^2 \rho(p)\, dp. \tag{1.2}$$

These can be used in subsequent analysis, inversion or optimization algorithms.

Existing forward propagation of uncertainty techniques for PDEs largely fall into one of two approaches: black-box and embedded. Black-box techniques, like Monte-Carlo [8] and stochastic collocation [13], are built around a deterministic PDE code and externally "sample" that code at carefully chosen points in parameter space. The end goal is to compute moments (as in Monte-Carlo) or build an interpolated surface (as in stochastic collocation).

Both black-box approaches have well known deficiencies. Monte-Carlo suffers from slow convergence and stochastic collocation suffers from the "curse of dimensionality" (several approaches such as anisotropic sparse grids [12] have been proposed to circumvent this issue). Embedded techniques, like the stochastic Galerkin method [9], require a direct modification of the code. The stochastic Galerkin method finds a projection of the PDE across stochastic and physical space simultaneously, thus inducing an artificial coupling between parameters. This method suffers from the same curse of dimensionality as stochastic collocation. Recent advances in preconditioners has demonstrated that this technique can be made competitive for linear problems [7], while effective solution methods for nonlinear PDEs remains an outstanding issue.

Both the black-box and embedded methods share a core theme. They assume that the uncertainty in the solution and the sensitivity of the response to that uncertainty is uniformly distributed acrossed the physical domain. This is implicit in the black-box techniques since they have no control over localized resolution of the solution in space. For the stochastic Galerkin method the uniformity assumption is apparent from the discrete space used. For instance, given a set of basis functions $\{\psi_i\}_{i=0}^{M}$, u is approximated as

$$u(x,p) \approx \sum_{i=0}^{M} u_i(x)\psi_i(p). \tag{1.3}$$

Here the coefficient of the expansion u_i is a function of only Ω. In a fully discrete setting, u_i is approximated using a deterministic finite element basis resulting in

$$u(x,p) \approx \sum_{i=0}^{M}\sum_{j=0}^{N} u_{i,j}\phi_j(x)\psi_i(p). \tag{1.4}$$

where ϕ_j is the spatial finite element basis. From this expansion it is clear that the approximation for u is a tensor product between the deterministic and stochastic basis functions.

To understand the consequence of this, consider a convection dominated flow where uncertainty is injected at a point in the center of the spatial domain (see Figure 1.1). Only the solution downstream of the injection point will be affected by the uncertainty. The upstream solution will be fully defined by the deterministic PDE. Existing uncertainty propagation methods would approximate both the upstream and downstream solutions with the same resolution. But this wastes computational resources on the upstream solution and may under resolve the solution downstream.

This report discusses an approach to break the tensor product structure in Eq. (1.4) and allow the stochastic expansion to vary as a function of the spatial location. We also present an algorithm for adaptively refining the approximation so that an improved distribution of stochastic degrees of freedom is achieved.

The outline for this report is as follows. To make the process of forward propagation of uncertainty explicit Chapter 2 presents a simplified nuclear-reactor simulation under uncer-

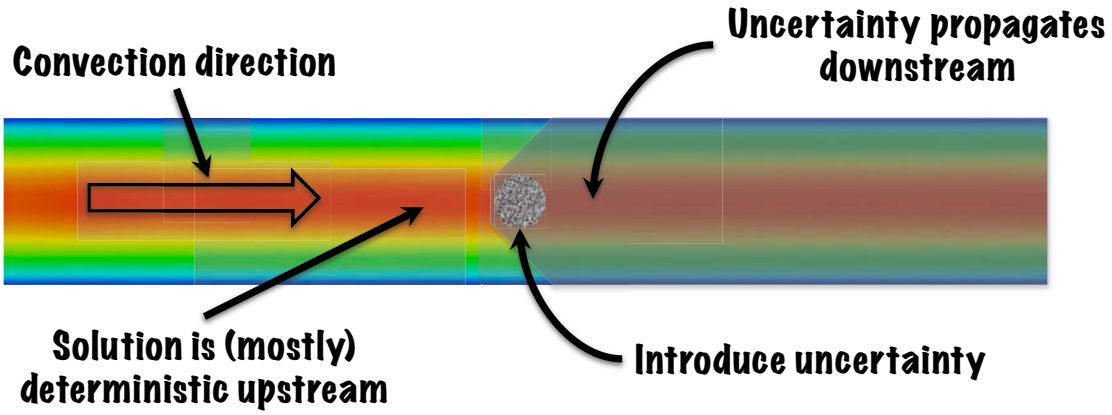

Figure 1.1. Example advection-diffusion problem with uncertainty introduced in the center of the domain. Notice that the uncertainty is propagated only downstream of where it was introduced.

tainty (this Chapter can be skipped for those with UQ experience). In Chapter 3, several uncertainty propagation methods are summarized. Including the stochastic Galerkin method that is reviewed in some detail. Critically, this includes the process of assembling the linear system. Chapter 4 presents the spatially varying adaptive assembly. In this section we discuss the adaptive assembly process, in addition to presenting several adaptive algorithms. In Chapter 5 we present numerical results comparing the uniform stochastic Galerkin methods to the spatially varying method developed here. Two steady-state convection diffusion problems are considered. Finally, in Chapter 6 we conclude and provide suggestions for future directions.

Chapter 2

An Example Problem

To demonstrate one possible use case for forward uncertainty propagation we considered a problem of potential interest to the nuclear reactor simulation community and, in particular, the Consortium for Advanced Simulation of Lightwater reactors (CASL) project[1]. Figure 2.1 shows the simulation geometry. The figure shows a simplified fuel-rod fluid geometry modeling the heat transfer of the rod to the fluid. Heat energy is produced in the *Fuel* sub-domain, it is passed through the *Cladding* and finally it is advected away (towards the *Outflow* boundary) in the *Fluid* subdomain. The geometry of the figure and simulation is understood to be only part of the physical domain, with the far left boundary of the *Fluid* containing a symmetry boundary condition being the centerline of the fluid. To be precise, the steady-state principal equations are

$$\vec{u} \cdot \nabla \vec{u} - \nu \nabla^2 \vec{u} + \nabla p = 0 \quad x \in \Omega_{Fluid} \tag{2.1}$$

$$\nabla \cdot \vec{u} = 0 \quad x \in \Omega_{Fluid} \tag{2.2}$$

$$\vec{u} \cdot \nabla T - \nu \nabla^2 T + S(\xi_0) = 0 \quad x \in \Omega_{Fluid} \cup \Omega_{Cladding} \cup \Omega_{Fuel} \tag{2.3}$$

and the fluid inflow condition is defined to be a plug flow with the magnitude defined by the random variable ξ_1. The remaining boundary conditions are shown in Figure 2.1. The source term is defined in terms of the random variable ξ_0

$$S(\xi_0) = \begin{cases} \xi_0 & x \in \Omega_{Fuel} \\ 0 & x \in \Omega_{Fluid} \cup \Omega_{Cladding}. \end{cases} \tag{2.4}$$

The heat production source random variable ξ_0 is normally distrubuted with mean 3.0 and variance 0.0625, the magnitude of the inflow velocity, ξ_1, is normally distributed with mean 1.0 and variance 0.01. Therefore the stochastic space is normally distributed with dimension two.

This problem was simulated using the **Drekar** [14] finite element code developed with support from ASCR, LDRD and CASL. The physical domain is discretized by the finite element methods using a quadrilateral mesh and equal order stabilized discretizations. The

[1]See `http://www.casl.gov`. The example referenced here was included in the image gallery (current as of June 25, 2013) see `http://www.casl.gov/media/gallery_images/CASL_24.jpg`.

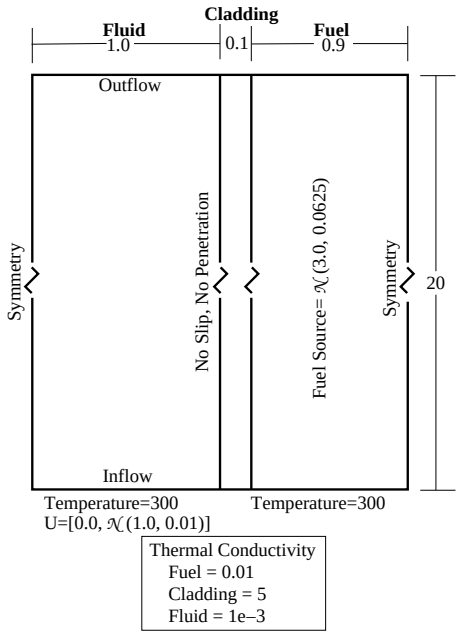

Figure 2.1. Two dimensional geometry of a simplified single pin reactor model. This problem has three subdomains, and the boundary conditions are specified.

temperature flux boundary condition across the *Fluid-Cladding* and *Cladding-Fuel* interface is enforced naturally. The stochastic space is discretized using a stochastic Galerkin method using Hermite polynomial basis functions (for a discussion of the stochastic Galerkin method see [9, 17] and Chapter 3).

Figure 2.2 shows the mean temperature and standard deviation in the fluid, cladding and fuel. Here we see that the mean has a large variation inside the rod, but only a small boundary layer is growing within the fluid. This suggests that the fluid, as described does not transfer much energy away from the rod. Additionally the standard deviation plot shows the temperature varying by around 5% as a result of the uncertainty in both the source and the inflow condition.

Figure 2.3 shows three plots showing the mean of the temperature and velocity components at three different heights of the rod. In addition, error bars on the plot denote one standard deviation of the solution. The three different heights the solution is sampled at are $y = 5$ (black line), $y = 10$ (blue line) and $y = 15$ (red line). For reference, the inflow condition is at $y = 0$ and the total height of the rod (as simulated) is 20. The first plot shows the temperature field. Notice the temperature reaches a maximum at the centerline of the fuel rod and increases with the height of the rod (this was already observed in Figure 2.2). The

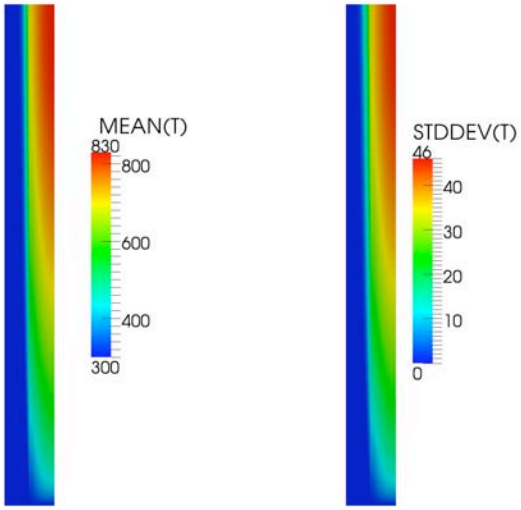

Figure 2.2. Mean and standard deviation of the temperature in a simplified reactor model. Here the left edge is the center line of the fluid, and the right edge is the centerline of the fuel rod.

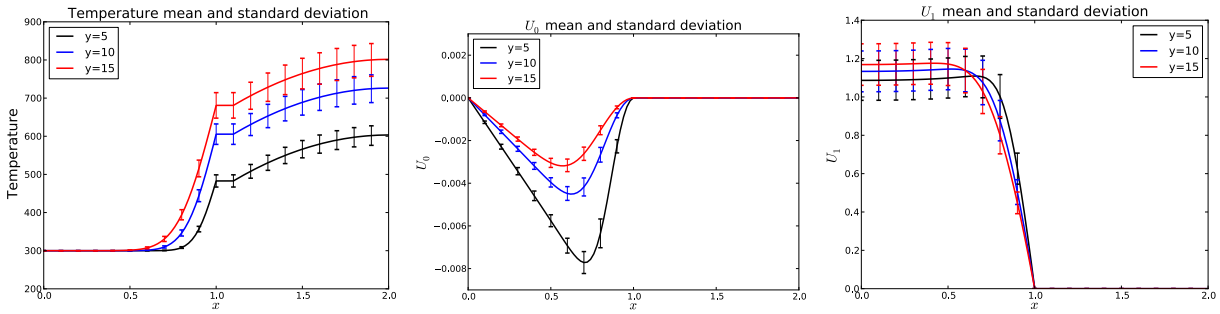

Figure 2.3. Plots of the mean with standard deviation error bars. This shows how the mean and standard deviation of the different fields vary at three different heights on the rod ($y = 0$ is the inflow).

temperature decays all the way to the cladding, where the temperature remains relatively flat due to the thermal-diffusivity constant. Finally, in the fluid the formation of a boundary layer is clear as the temperature decays to the inflow temperature of 300. The sensitivity of the temperature is shown by the error bars, with a steady increase in the size of the error bars with increasing height. The two velocity plots show the velocity evolving to a parabolic flow (flow in a channel) with increasing height. This suggests that the flow is not yet fully developed. Additionally, unlike the temperature, the velocity does not demonstrate an increase in standard deviation with increasing height. This suggests it is relatively insensitive to the uncertainty in the heat source and is primarily dominated by the uncertainty in the inflow condition (this makes sense since the velocity is not dependent on the temperature field the temperature is a dependent function of velocity). Adding a Bousinesq source term to the momentum equation would cause a coupling from temperature to velocity, and would certainly change this result.

Chapter 3

The Stochastic Galerkin Method

In the previous chapters we motivated why solving a PDE with stochastic parameters is a useful goal. However, practical numerical methods for solving the problems have not as of yet been discussed. In this chapter, we review the stochastic Galerkin method that is critical to the development of the spatially varying method developed in this report. Additional methods for black-box UQ such as stochastic collocation [13] and Monte-Carlo [8] are not discussed further.

3.1 Notation

We employ the standard definitions for the deterministic finite element space, see for example [1, 2, 4]. Let $L^2_\omega(\Omega)$ be the space of square integrable functions on the domain:

$$L^2_\omega(\Omega) = \left\{ f : \int_\Omega f(x)^2 \omega(x)\, dx < \infty \right\}. \tag{3.1}$$

If the measure ω on the domain is omitted it is assumed to be 1, or it is clear from context. Similarly the space $H^k(\Omega)$ is the space of k differentiable functions:

$$H^k(\Omega) = \left\{ f \in L^2(\Omega) : f^{(i)} \in L^2(\Omega) \quad \text{for } i = 1 \dots k \right\}. \tag{3.2}$$

If homogeneous Dirichlet conditions are implied on atleast a subset of the boundary then the standard notation $H^k_0(\Omega)$ is used. The space H^k for $k = 0 \dots$ is a Hilbert space equiped with norm

$$\|f\|^2_{H^k} = \sum_{i=0}^{k} \int_\Omega (f^{(i)}(x))^2\, dx \quad \forall f \in H^k(\Omega) \tag{3.3}$$

and inner product

$$(f, g)_{H^k} = \sum_{i=0}^{k} \int_\Omega f^{(i)}(x) g^{(i)}(x)\, dx \quad \forall f, g \in H^k(\Omega). \tag{3.4}$$

We will drop the subscript notation on both the norm and inner product where it is clear from context. The discrete approximation of a Hilbert space is denoted by V^h where h is a

parameter indicating the resolution of the finite element mesh. Note that V^h is understood to be in the range of piecewise linear finite element basis functions defined on the mesh.

In the stochastic space we are interested in functions who live in $L^2_\rho(\Gamma)$. If the solution to the deterministic Hilbert space is V^h then the PDE solved with uncertain parameters is in the Bochner space $L^2_\rho(\Gamma, V^h)$. Note for the purposes here this is simply the completion of the product space $V \times L^2_\rho(\Gamma)$. Discrete approximations of the stochastic space will be denoted by $U^k(\Gamma) \subset L^2_\rho(\Gamma)$ where k is the polynomial order of the approximation, that is

$$U^k(\Gamma) = \left\{ f \in L^2_\rho(\Gamma) : f = \sum_{i=0}^{k} \alpha_k p^k, \quad p \in \Gamma \right\} \tag{3.5}$$

The notation for the Bochner space will be extended to any discrete space. For instance if $U^k(\Gamma) \subset L^2_\rho(\Gamma)$ is a discrete subset then $U^k(\Gamma, V^h)$ is the Bochner space equivalent (simply the completion of $V^h \times U^k(\Gamma)$).

3.2 Stochastic Galerkin Weak Form

Let $u(\cdot,\cdot) \in L^2(\Gamma, H^1_0(\Omega))$ be the solution to the PDE with uncertain parameters drawn from the probability space $(\Gamma, \mathcal{F}, \rho)$

$$\begin{aligned} \mathcal{L}(u(x,p);x,p) &= f(x,p) \\ u(\partial\Omega_d, p) &= 0. \end{aligned} \tag{3.6}$$

Further assume that Eq. 3.6 has an equivalent weak form statement for fixed $p \in \Gamma$

$$\text{Find } u(p) \in H^1_0(\Omega) \text{ such that} \quad a(u(p); v\,|p) = L(v\,|p) \quad \forall v \in H^1_0(\Omega) \tag{3.7}$$

where $u(p)$ is the deterministic solution at $p \in \Gamma$. The ';' is used to indicate that $a(\cdot;\cdot)$ is possibly nonlinear in the first position.

To find $u \in L^2(\Gamma, H^1_0(\Omega))$ the stochastic Galerkin method uses a projection over the stochastic space giving

$$\int_\Gamma \int_\Omega \left(f(x,p) - \mathcal{L}(x,p;u(x,p))\right) v(x,p)\rho(p)\,dxdp = 0 \quad \forall v \in L^2(\Gamma, H^1_0(\Omega)) \tag{3.8}$$

$$\iff \int_\Gamma L\left(v(\cdot,p)\,|p\right) - a\left(u(\cdot,p), v(\cdot,p)\,|p\right)\rho(p)\,dp = 0 \quad \forall v \in L^2(\Gamma, H^1_0(\Omega)). \tag{3.9}$$

This is the weak form residual for the approximation of the stochastic solution. This section is concerned with discretization of Eq. (3.9) and how to assemble Jacobian operators and residual right hand side vectors for this problem. In particular, finite element assembly in the context of a full nonlinear iterative solve will be considered.

3.2.1 Approximation and Assembly

The discrete statement of the stochastic Galerkin equation is to find $u^{h,k} \in U^k(\Gamma, V^h)$ such that

$$\int_\Gamma a\left(u^{h,k}, v\right) \rho(p)\, dp = \int_\Gamma L\left(v\right) \rho(p)\, dp \quad \forall v \in U^k(\Gamma, V^h) \tag{3.10}$$

where the explicit dependence on the stochastic parameter p has been dropped to simplify presentation. To be concrete any function in $f \in U^k(\Gamma, V^h)$ can be written as

$$f(x,p) = \sum_{i=0}^{M} \sum_{j=0}^{N} f_{i,j} \phi_j(x) \psi_i(p) \tag{3.11}$$

where ϕ_i is a finite element basis function [1, 2, 4] and ψ_i is a spectral basis function over stochastic space [9, 10, 17]. Through out this report we use low-order finite element basis functions and orthogonal polynomials for the stochastic space. For simplicity we instead write

$$f(x,p) = \sum_{i=0}^{M} f_i(x) \psi_i(p) \quad \text{where} \quad f_i(x) = \sum_{j=0}^{N} f_{i,j} \phi_i(x) \tag{3.12}$$

in this section.

To solve Eq. 3.10 we will use Newton's method, the corresponding residual is

$$B_i(u^{h,k}) = \frac{1}{\|\psi_i\|_{L^2}^2} \int_\Gamma \left(L(v_i) - a(u^{h,k}, v_i)\right) \psi_i(p) \rho(p)\, dp \quad i \in [0, M]. \tag{3.13}$$

Notice that B_i is actually a vector of length $N+1$ if you fully expand the basis as in Eq. 3.11. Further because L is linear and a is linear in the second slot the ψ_i factors out. The Jacobian from Newton's method is defined as

$$A_{ij} = \frac{\partial B_i}{\partial u_j} = \frac{1}{\|\psi_i\|_{L^2}^2} \int_\Gamma -\frac{\partial a}{\partial u}(u^{h,k}, v_i) \psi_i(p) \psi_j(p) \rho(p)\, dp. \tag{3.14}$$

Notice that constructing this with quadrature would amount to computing $O(P^2)$ entries (where each is a deterministic matrix of size $(N+1) \times (N+1)$) over a set of quadrature points. This is prohibitively expensive for high dimensional stochastic spaces. Instead the Jacobian is approximated with the stochastic expansion

$$-\frac{\partial a}{\partial u}(u^{h,k}; v_i) \approx \sum_{l=0}^{M} J^l \psi_l(p) \tag{3.15}$$

where the matrix J^k is defined using a Galerkin projection

$$J^l = \frac{1}{\|\psi_i\|_{L^2}^2} \int_\Gamma -\frac{\partial a}{\partial u}(u^{h,k}; v_l) \psi_l(p) \rho(p)\, dp. \tag{3.16}$$

Now the Jacobian is approximated by substituting Eq. (3.15) into Eq. (3.14), giving

$$\bar{A}_{ij} = \sum_{l=0}^{M} J^l C_{lji} \tag{3.17}$$

where the sparse tensor C is

$$C_{ijl} = \frac{\langle \psi_i \psi_j \psi_l \rangle}{\langle \psi_l^2 \rangle}. \tag{3.18}$$

Note that C_{ijl} is symmetric in i and j ($C_{ijl} = C_{jil}$).

3.3 Example Convection Diffusion Problem

To demonstrate and motivate the spatially varying stochastic expansions we consider the solution of the stochastic partial differential equation

$$-\nu(\xi)\frac{\partial^2 u(x,\xi\,)}{\partial x^2} + \frac{\partial u(x,\xi\,)}{\partial x} = 0 \quad x \in (0,1), \xi \in \mathcal{N}(0,1), \tag{3.19}$$

$$u(0) = 1,\ u(1) = 6. \tag{3.20}$$

Uncertainty is introduced by the normally distributed random variable ξ with the diffusion coefficient taking the form

$$\nu(\xi) = \nu_0 + \alpha \mathbf{e}^{\sigma\xi} \tag{3.21}$$

where $\alpha = 0.25$. The expectation and variance of ν are

$$\mathbb{E}[\nu] = \nu_0 + \alpha e^{\sigma^2/2} \quad \text{and} \quad \mathbb{V}ar[\nu] = \alpha^2 e^{\sigma^2}(e^{\sigma^2} - 1). \tag{3.22}$$

Clearly, depending on the parameters ν_0, σ and the realization of the random variable ξ the balance between convection and diffusion in Eq. (3.19) can vary widely. The solution to this problem as a function of ξ and x is

$$u(x,\xi\,) = 1 + \frac{5}{e^{\nu(\xi)^{-1}} - 1}\left(e^{\nu(\xi)^{-1}x} - 1\right). \tag{3.23}$$

Using normalized Hermite basis functions in the stochastic domain, Fig. 3.1 shows the coefficients of each stochastic basis solving Eq. (3.19) as function of space. Note that because the basis functions are normalized, the "influence" of coefficients carries the same weight on the solution value $u(x,\cdot)$ allowing a direct comparison between the coefficients (the smaller the coefficient the less it effects the solution value $u(x,\cdot)$). This allows us to determine the level of resolution required for the expansion of the stochastic solution at a fixed point in space. We see that for problems with viscosity near zero (including stochastic perturbation) there is a high dependence on the required resolution of the stochastic solution on the spatial

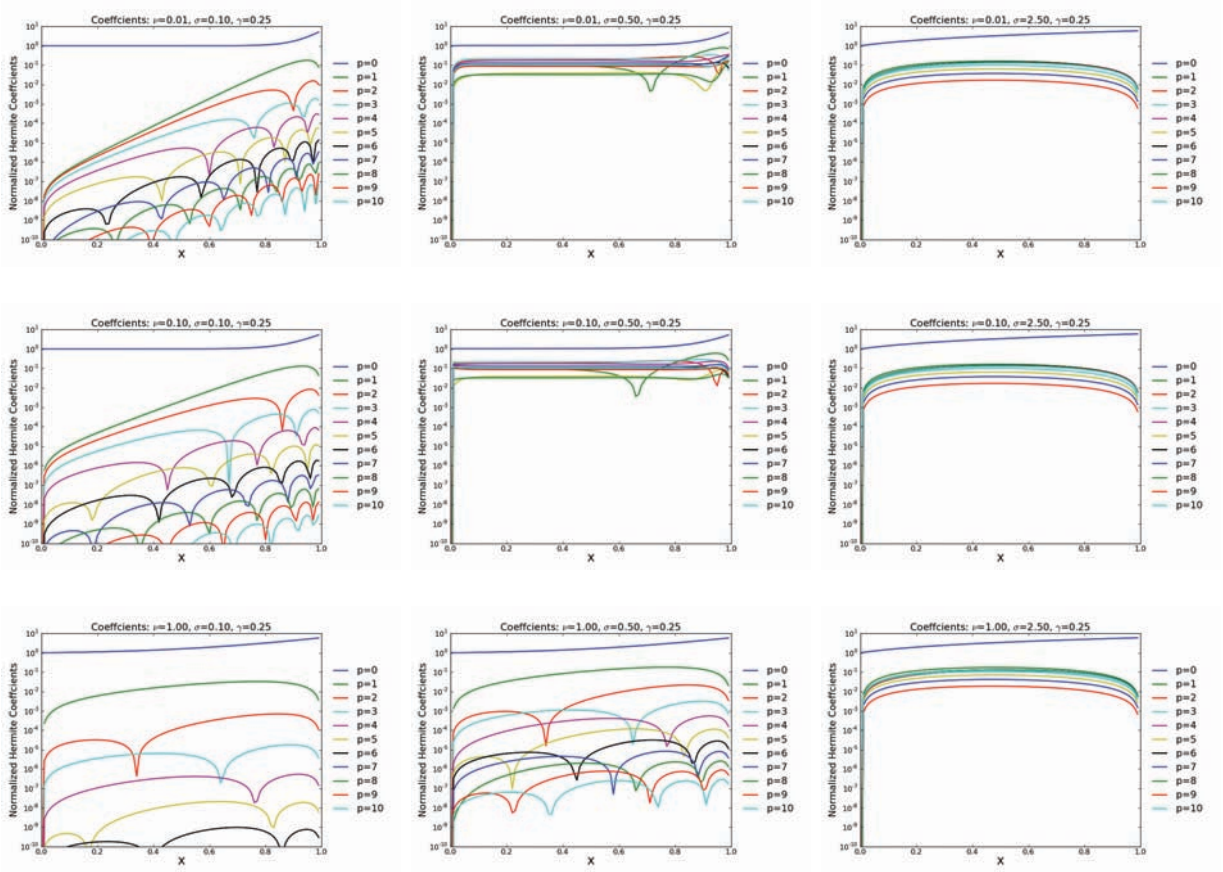

Figure 3.1. The convection-diffusion system using a log-normal distribution of the viscosity and a tensor product stochastic discretization.

domain. However as the viscosity is allowed to vary more widely (and hence gets larger) the required resolution of the stochastic solution becomes independent of space. Note that the low order solution is required regardless of expansion. This is a function of the chosen dirichlet boundary conditions, otherwise its influence can be made arbitrarily large.

Chapter 4

Spatially Varying Discretization

4.1 Adaptive Assembly

4.1.1 Abstract Basis

The assembly outlined in Sec. 3.2.1 is by now a fairly standard approach for embedded uncertainty quantification. In this section we take a step back to the general case where the basis is not assumed to be formed by a tensor product, but is still *separable*. In this case we will define our test and trial spaces to be

$$\mathcal{V}^h := \mathbf{Span}\left(\{\phi_i(x)\eta_i(\theta)\}_{i=1}^N\right) \subset V \times \Xi. \tag{4.1}$$

This gives us the N nonlinear equations

$$\int_\Gamma \left[L(\phi_i|\theta) - a(u;\phi_i|\theta)\right]\eta_i(\theta)\, dP(\theta) = 0. \tag{4.2}$$

Choosing our discrete trial space to be $\mathcal{V}^h$ and writing $u \approx \hat{u} = \sum_{i=1}^N u_i\phi_i\eta_i$ gives N equations for N unknowns

$$B_i(\hat{u}) = \frac{1}{\langle\eta_i^2\rangle}\int_\Gamma \left(L(\phi_i|\theta) - a(\hat{u};\phi_i|\theta)\right)\eta_i(\theta)\, dP(\theta) = 0 \quad \forall i \in [1\ldots N]. \tag{4.3}$$

Differentiating with respect to u_j gives

$$A_{ij} = \frac{\partial B_i}{\partial u_j} = \frac{1}{\langle\eta_i^2\rangle}\int_\Gamma -\frac{\partial a}{\partial u}(\hat{u};\phi_j,\phi_i|\theta)\eta_j(\theta)\eta_i(\theta)\; dP(\theta). \tag{4.4}$$

Comparing to Eq. (3.14), we see that each i,j pair corresponds to a single entry in the matrix A, before it was denoting a block matrix.

To clarify things we will write the expansion of $\partial a/\partial u$ in terms of a different basis over the stochastic space. Thus defining

$$\Xi^K := \mathbf{Span}\left(\{\xi_i\}_{i=1}^K\right) \subset \Xi. \tag{4.5}$$

and writing

$$-\frac{\partial a}{\partial u}(\hat{u};\phi_j,\phi_i|\theta) \approx \sum_{k=1}^{K} J_{ij}^k \xi_k(\theta). \tag{4.6}$$

Using a Galerkin projection we have

$$J_{ij}^k = \frac{1}{\langle \xi_k^2 \rangle} \int_\Gamma -\frac{\partial a}{\partial u}(\hat{u};\phi_j,\phi_i|\theta)\xi_k(\theta)\, dP(\theta). \tag{4.7}$$

(Notice that for $\xi_k = \psi$ we have that J^k from Eq. (3.16) can be written as $[J^k]_{ij} = J_{ij}^k$) Substituting back into the exact expression for A_{ij} yields the new, *approximate*, expression

$$\hat{A}_{ij} = \sum_{k=1}^{K} J_{ij}^k \frac{\langle \xi_k \eta_j \eta_i \rangle}{\langle \eta_i^2 \rangle}. \tag{4.8}$$

The point of using the spaceΞ^K should now be clear. The question of how to approximate $\partial a/\partial u$ is largely independent of how the test and trial spaces are chosen, as long as one is willing to adjust the tensor used in the approximation.

4.1.2 Locally Defined Stochastic Basis

Equations (4.3) and (4.8) give a very general view of what an adaptive strategy for stochastic Galerkin methods might look like. Here we go through a more practical view that explicitly handles spatially varying discrete stochastic spaces. First some additional notation is necessary.

- Let $T \subset \{1 \dots N\} \times \{1 \dots P\}$, where N is the number of deterministic degrees of freedom and P is the maximum number of stochastic degrees of freedom, such that

  $$T = \{(i,j) : i \in \{1 \dots N\} \text{ and } j \in \{1 \dots P_i\}\} \quad \text{where } 0 < P_i \leq P.$$

  Here P_i is the number of stochastic basis functions associated with a particular deterministic degree of freedom i. The cardinality $|T|$ is equal to the number of unknowns in the system. Finally let γ be an invertible map between T and $\{1 \dots |T|\}$ (thus $\gamma : T \to \{1 \dots |T|\}$ and $\gamma^{-1} : \{1 \dots |T|\} \to T$).

- Let the deterministic basis be defined by $\phi_i(x)$ for $i = 1 \dots N$. The full set of stochastic basis functions is defined to be $\psi_i(\theta)$ for $i = 1 \dots P$. For any particular determinsitic basis, the set of associated stochastic basis is denoted as $\psi(\theta)_j^i$ where $j \in \{1 \dots P_i\}$ and $\psi_j^i(\theta) = \psi_j(\theta)$. Using the definition of T from above we have the full discrete space

  $$S^h = \mathbf{Span}\left(\phi_i(x)\psi_j^i(\theta)\ \forall (i,j) \in T\right) \tag{4.9}$$

Using these definitions of T, γ and S^h from above we can outline the assembly of a spatially varying adaptive basis.

First write the approximation to u as

$$u(x,\theta) \approx \hat{u}(x,\theta) = \sum_{i=1}^{N} \sum_{j=1}^{P_i} u_{ij} \phi_i(x) \psi_j^i(\theta). \tag{4.10}$$

The using the approximation $\hat{u}$ the residual vector B is defined as

$$B_{\gamma(i,m)}(\hat{u}) = \frac{1}{\langle (\psi_m^i)^2 \rangle} \int_\Gamma \left[L(\phi_i|\theta) - a(\hat{u};\phi_i|\theta) \right] \psi_m^i(\theta)\, dP(\theta) \tag{4.11}$$

This equation corresponds directly to Eq. (4.3). Now to construct the Jacobian operator we differentiate the residual with respect to the coefficients

$$A_{\gamma(i,m)\gamma(j,n)} = \frac{\partial B_{\gamma(i,m)}}{\partial u_{jn}} = \frac{1}{\langle (\psi_m^i)^2 \rangle} \int_\Gamma -\frac{\partial a}{\partial u}(\hat{u};\phi_j,\phi_i|\theta) \psi_m^i(\theta) \psi_n^j(\theta)\, dP(\theta) \tag{4.12}$$

which corresponds to Eq. (4.4). Here again we approximate the integral over $\partial a/\partial u$ using a truncated expansion in the stochastic space. This gives

$$-\frac{\partial a}{\partial u}(\hat{u};\phi_j,\phi_i|\theta) = \sum_{k=1}^{P} J_{ij}^k \psi_k(\theta) \tag{4.13}$$

where again using a Galerkin projection

$$J_{ij}^k = \frac{1}{\langle \psi_k^2 \rangle} \int_\Gamma -\frac{\partial a}{\partial u}(\hat{u};\phi_j,\phi_i|\theta) \psi_k(\theta)\, dP(\theta). \tag{4.14}$$

Notice that here we have used the generic *master* basis over the stochastic space because this expansion is not associated with any deterministic basis. Again plugging this into the expression for the Jacobian gives the new (approximate) Jacobian operator

$$\hat{A}_{\gamma(i,m)\gamma(j,n)} \approx \sum_{k=1}^{P} J_{i,j}^k \frac{\langle \psi_k \psi_n^j \psi_m^i \rangle}{\langle (\psi_m^i)^2 \rangle}. \tag{4.15}$$

4.2 Example Convection Diffusion Problem

Figure 4.1 again shows the coefficients of a set of normalized Hermite basis functions varying as a function of space. In this case however, only polynomials of order $0-5$ (inclusive) cover the whole domain. Polynomials of order $6-10$ are only active on the interval $[0.5, 1.0]$. From these images it is clear that for some values of ν,σ and γ the very highest order polynomials

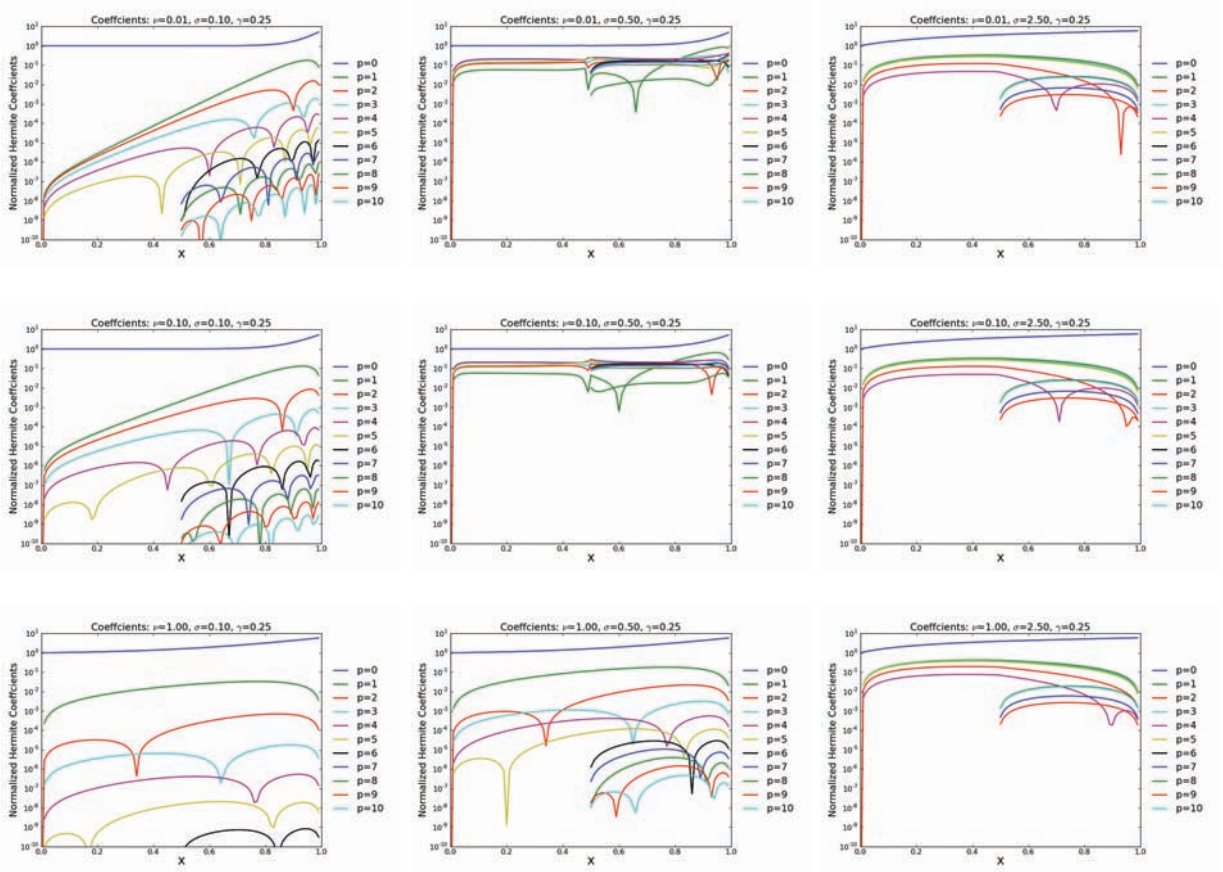

Figure 4.1. The convection-diffusion system using a log-normal distribution of the viscosity and a spatially varying stochastic expansion.

are not needed on the interval from $[0, 0.5]$. For instance the upper left hand corner image shows that to achieve point wise accuracy up to approximately 10^{-7} only the first six Hermite polynomials are needed on the interval from $[0, 0.5]$. However on the interval from $[0.5, 1.0]$ almost all 10 polynomials are needed. This shows that the requirement for accuracy varies across the physical domain and that adaptively improving the approximation space may lead to a more accurate solution with fewer unknowns.

4.3 Adaptive Algorithm

The adaptive algorithm we have developed follows the rough outline presented in [6, 3] for adaptive finite element methods. The process proceeds in four stages:

$$\boxed{\text{SOLVE}} \rightarrow \boxed{\text{ESTIMATE}} \rightarrow \boxed{\text{MARK}} \rightarrow \boxed{\text{REFINE}}. \tag{4.16}$$

In the SOLVE stage the PDE is solved, this step will not be discussed further here. In the ESTIMATE step error indicators are used to compute an approximation of the local contribution to the error. These error indicators are used by the MARK step to choose a subset of the discretization to enrich using refinement. Finally REFINE is the step where the discretization is improved. For instance in finite element h-refinement the mesh elements are subdivided increasing the fidelity of the discretization. We discuss our approach to ESTIMATE, MARK and REFINE below. Ideally, the goal of this algorithm is to achieve an equidistribution of error across the mesh. If this is successful then a uniform refinement scheme would be the optimal approach. In practice, this equidistribution is not achieved but it serves as a motivating benchmark for the algorithm.

4.3.1 ESTIMATE

In this section we present two error estimators. The first is an element based estimator that uses a higher-order approximation to compute error estimates. The second uses a nodal error estimator based on asymptotic convergence assumptions for polynomial approximations.

Element indicator

The first adaptive estimator is costly and suboptimal from a practical perspective; however, it will give a proof of principle demonstration of the adaptive spatially varying stochastic discretization. The general approach has been succinctly outlined in the monograph [15]. Let L_h^{-1} represent a discrete solution operator mapping the right hand side of a PDE $f \in L^2(\Omega)$ to a discrete representation V_h. Further let $\hat{L}_h^{-1}$ map the same right hand side $f \in L^2(\Omega)$ to a more accurate solution space $\hat{V}_h$ where $V_h \subset \hat{V}_h$. For instance in deterministic finite element methods V_h may be the space of piecewise linear polynomials defined on a mesh, and $\hat{V}_h$ might be these space of piecewise quadratics.

Now let $u = L^{-1}(f)$ be the continuous "exact" solution, $u_h = L_h^{-1}(f) \in V_h$ and $\hat{u}_h = \hat{L}_h^{-1}(f) \in \hat{V}_h$. The high order solution represents a more accurate version of the original solution and will be used to estimate the error in the L^2 norm:

$$\|u - u_h\|^2 \leq \|u - \hat{u}_h\|^2 + \|\hat{u}_h - u_h\|^2. \tag{4.17}$$

Note that because $\hat{u}_h$ is higher-order than u_h, then the term $\|u - \hat{u}_h\|^2$ is relatively small compared to $\|\hat{u}_h - u_h\|^2$. Thus a reasonable bound on the L^2 error in the low order solution is

$$\|u - u_h\|^2 \lesssim \|\hat{u}_h - u_h\|^2. \tag{4.18}$$

While a bound on the error is useful for error estimation, to adaptively refine the solution a localization of that error must be constructed. A readily computed approximation is

$$\|\hat{u}_h - u_h\|^2 = \sum_K \eta_K \tag{4.19}$$

where $\eta_K = \|\hat{u}_h - u_h\|^2_{L^2(K)}$ is the element wise indicator for the L^2 error.

Nodal indicator

The second type of indicator is based on an assumption of being in an asymptotic regime with respect to the stochastic solution space. Here we briefly review the idea, we point the interested reader to the further discussion and analysis found in [11, 16]. Assume the stochastic function $g(\xi)$ is approximated by

$$g(\xi) \approx \hat{g}(\xi) = \sum_{j=1}^{M} g_j \psi_j(\xi) \tag{4.20}$$

where ψ_j are orthonormal basis functions with respect to the relevant probablility density function. Using orthonormality the variance of the approximation is

$$\mathrm{Var}[\hat{g}] = \sum_{j=2}^{M} g_j^2. \tag{4.21}$$

The simple error indicator we will use is

$$\eta_g = \frac{g_M^2}{\mathrm{Var}[\hat{g}]}. \tag{4.22}$$

Assuming that the expansion of M terms is sufficient to put the approximation in the asymptotic regime, then the indicator determines how much of the variance is provided by the last coefficient. If that ratio is high, then adding an additional term is required until asymptotic convergence is achieved.

4.3.2 MARK

The goal of the MARK stage of the adaptive refinement algorithm is to produce a list (denoted $\mathcal{M}$ below) of either elements or deterministic degrees of freedom to refine. Note

that because this is a list multiple entries can be repeated. The input to the mark step is the error estimates for each element or deterministic degree of freedom. There are several possible approaches to marking, here we list two such approaches we have considered (see for instance [6] for other approaches used in adaptive mesh refinement).

Algorithm 1: MARK_PERCENT

input $\mathcal{E}$: Vector of index/error indicators pairs (nodal or elemental)
input τ : Tolerance for this marking strategy
output $\mathcal{M}$: List of marked indices (either nodal or elemental)

$\mathcal{M} \leftarrow \emptyset$
$\eta_{max} \leftarrow \text{Max}\{\eta : (i,\eta) \in \mathcal{E}\}$
for $(i,\eta) \in \mathcal{E}$ **do**
  `// i is the index and η is the indicator`
  if $\tau \cdot \eta_{max} < \eta$ **then**
    $\mathcal{M} \leftarrow \mathcal{M} :: i$ `// :: is a concatenation operator`
  end
end

Algorithm 1 is a simple strategy appropriate for marking either nodes or elements. To simplify the discussion we will always refer to the type of entry to be marked as an "element". However, it is understood that error indicators based on nodes can also be used. The algorithm takes as input a list of error indicators (this implicitly includes the index associated with the element under consideration for marking) and a user specified error tolerance. This alogrithm marks all elements whose error indicator is within a percentage of the largest error indicator. The percentage used is specified by the user tolerance. At the end of the algorithm, $\mathcal{M}$ contains a list of indices that need to be marked.

Algorithm 2: MARK_MEAN

input $\mathcal{E}$: Vector of index/error indicators pairs (nodal or elemental)
input τ : Tolerance for this marking strategy
output $\mathcal{M}$: List of marked indices (either nodal or elemental)

$\mathcal{M} \leftarrow \emptyset$
$\eta_{mean} \leftarrow \text{Mean}\{\eta : (i,\eta) \in \mathcal{E}\}$
for $(i,\eta) \in \mathcal{E}$ **do**
  `// i is the index and η is the indicator`
  if $\tau \cdot \eta_{mean} \leq \eta$ **then**
    $\mathcal{M} \leftarrow \mathcal{M} :: i$ `// :: is a concatenation operator`
  end
end

A second marking strategy seen in Algorithm 2, has the same inputs and outputs as the previous algorithm. The goal of this approach is to mark all indicators significantly larger then the mean value of the error indicator. This ensures that the very largest elements are refined multiple times.

Each of the above MARKing strategies can be used in isolation. However, it maybe effective to use them in combination or with varying error tolerances. This will cause elements to be marked multiple times. To be precise, we introduce here an additional notation to specify the exact sequence of MARKing schemes and tolerances. For instance one possibility would be to MARK using Alg. 1 followed by Alg. 2. This would be written

$$\mathcal{M} = \text{MARK\_PERCENT}(\mathcal{E},\, \tau_p) :: \text{MARK\_MEAN}(\mathcal{E},\tau_m).$$

Note that we have specified arbitrary tolerances τ_p and τ_m with an assumed set of error indicators $\mathcal{E}$. Again the "::" operator is used to denote concatenation between two lists.

4.3.3 REFINE

In the ESTIMATE and MARK sections we presented an error indicator and marking strategy based on "nodal" coefficients (or more precisely the error in the coefficient of a particular basis function). We also presented an element-wise error indicator which localizes the error to an element of the mesh and a compatible marking strategy. Here we discuss the refinement algorithms used in both these contexts.

Element Error Indicators

Let $\mathcal{M}$ be the list of marked elements. For $K \in \mathcal{M}$, let $\mathcal{I}_K$ be the indices for the set of deterministic basis functions whose support overlaps with element K. Then the refinement algorithm is

$$P_i \leftarrow P_i + 1 \quad \forall i \in \mathcal{I}_K \text{ s.t. } K \in \mathcal{M}. \tag{4.23}$$

Stated succinctly this simply says the stochastic order for each basis intersecting with K is increased by one (note that this includes multiple fields). If an element is marked multiple times then the polynomial order can be incremented multiple times for each iteration of the adaptive refinement algorithm.

Nodal Error Indicators

The nodal based indicators are computed for each degree of freedom in the deterministic problem. If the marking strategy returns a list of deterministic degrees of freedom $\mathcal{M}$ to be

marked, then the refinement algorithm is

$$P_i \leftarrow P_i + 1 \quad \forall i \in \mathcal{M}. \tag{4.24}$$

Again, if a deterministic index appears multiple times in $\mathcal{M}$ then its stochastic polynomial order is increased multiple times.

Chapter 5

Results

In this chapter we present results for two convection-diffusion problems that demonstrate the performance of the adaptive refinement algorithm. In general, finding steady-state problems that the adaptive methodology performs well on can be difficult. Problems are required to have a degree of spatially locality in the solution which unfortunately eliminates many steady-state diffusion problems. However, convection-diffusion problems appear to have the right locality properties so that the proposed approach can be effective. Further study of highly nonlinear and transient problems is necessary to go beyond the proof-of-principle problems discussed here.

5.1 Convection Diffusion with an Internal Layer

The first model problem used to test the spatially varying adaptive UQ technology is a high Peclet number convection-diffusion problem with an internal layer. In this case

$$\begin{aligned} -\nabla\cdot\ \nu\nabla T + w\begin{bmatrix}\sin\theta\\ \cos\theta\end{bmatrix}\cdot\nabla T = 0 x\in\Omega \\ T\left(\partial\Omega_l\right) = 1 \\ T\left(\partial\Omega_b\right) = 0 \end{aligned} \tag{5.1}$$

The angle θ is distributed according to $\mathcal{N}(40,2)$, the viscosity $\nu = 10$ and the magnitude of the velocity $w = 10^8$ (giving a Peclet number of 10^7). For stabilization we employ a streamlined upwind Petrov-Galerkin method [5]. This method does a reasonable job of stabilizing the internal layer, with relatively small overshoots and undershoots observed for the 60×60 mesh used in these studies. Figure 5.1 is a schematic image of the domain, boundary conditions and convection direction.

An "exact" solution to this problem is constructed using a 20^{th} order stochastic Galerkin projection using a Hermite basis. Furthermore, to provide a reference to compare the adaptive scheme to, a sequence of 10 uniformly refined stochastic Galerkin problems (starting with order $p = 1$) using a Hermite basis are solved. In Figure 5.2 the blue line shows the error measured in the L^2 norm over physical and stochastic space as a function of the number

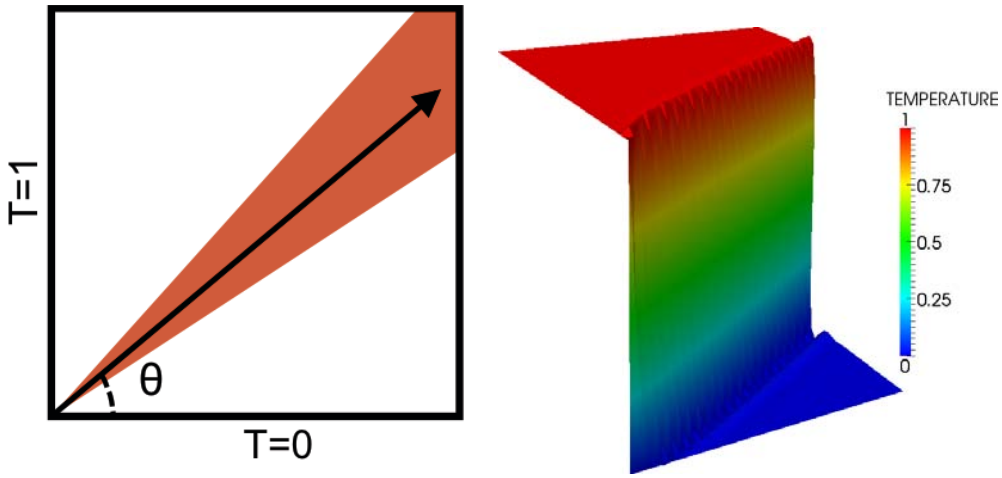

Figure 5.1. The left image shows a schematic diagram of a high Peclet number convection-diffusion problem with an internal layer (see Eq. (5.1)). The solution using SUPG is shown in the image on the right for $\theta = 40$.

of unknowns. Note that this is a semi-log plot and the log of the error is converging linearly in the total number of unknowns. Thus if N is the number of unknowns and Err is the error, then

$$\log \text{Err} = -\alpha N + \beta \Rightarrow \text{Err} = e^{\beta} e^{-\alpha N} \propto e^{-\alpha N}. \tag{5.2}$$

This implies the convergence rate is exponential as expected for polynomial refinement. Below we discuss results using both the element based and nodal mesh refinement schemes.

5.1.1 Element Based Refinement

To test the element based error indicators we used the MARKing strategy

$$\text{MARK\_PERCENT}(\mathcal{E},\, 0.1) :: \text{MARK\_MEAN}(\mathcal{E}, 10.0)$$

with an initial polynomial order of 0 over the physical domain. The high-order discretization used to compute the error indicator increases the polynomial order by one throughout the physical domain (thus it is still allowed to vary spatially, but will be one polynomial order higher). An iterative GMRES solver is converged to a relative tolerance of 10^{-6} using an ILU(2) preconditioner.

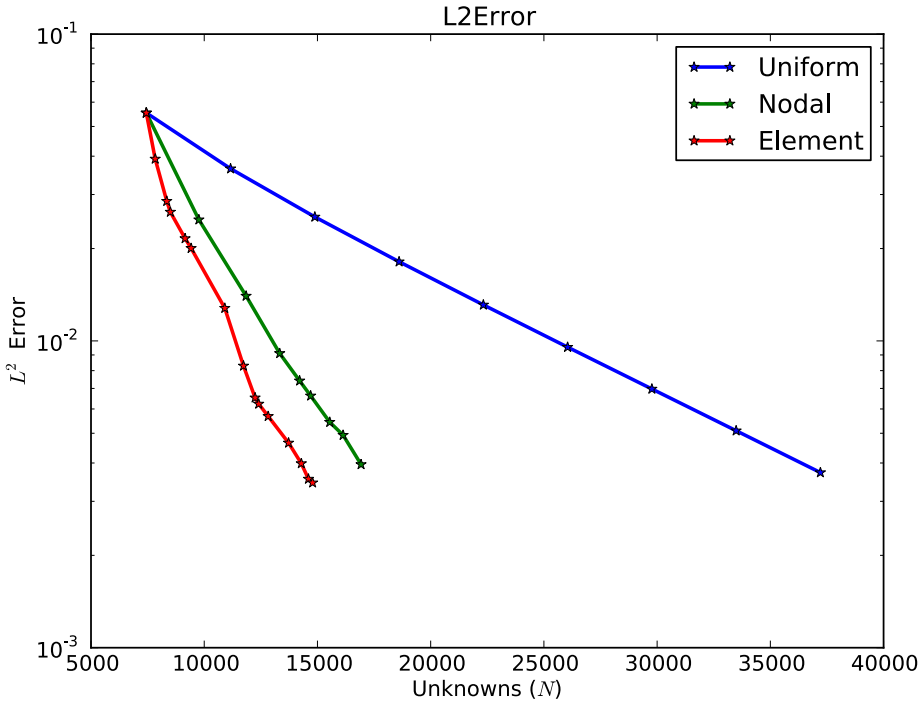

Figure 5.2. The plot shows the error as a function of total number of unknowns for refinement of both uniform and spatially varying stochastic expansions for the convection-diffusion problem defined in Eq. (5.1).

Figure 5.2 shows convergence using both uniform refinement (blue line) and the spatially varying adaptive method using element-wise indicators (red line) in the L^2 norm as a function of the number of unknowns. The error plotted is for the high-order solution thus the error between the first order uniformly refined solution is the same. Note that the adaptive method requires far fewer unknowns than uniform refinement of the stochastic Galerkin discretization to achieve the same level of accuracy. The reason for this improvement over the uniform case can be seen in Fig. 5.3. Each image is a color plot of the polynomial order at each point in the physical domain (red corresponds to high order and blue corresponds to 0^{th} order) as a function of levels of refinement (the first image is after one level of refinement, the second image is after two, and so on). Note how the refinement is strongly localized over the range of values for the angle θ. With the strongest refinement taking place where the solution is most sensitive to the angle.

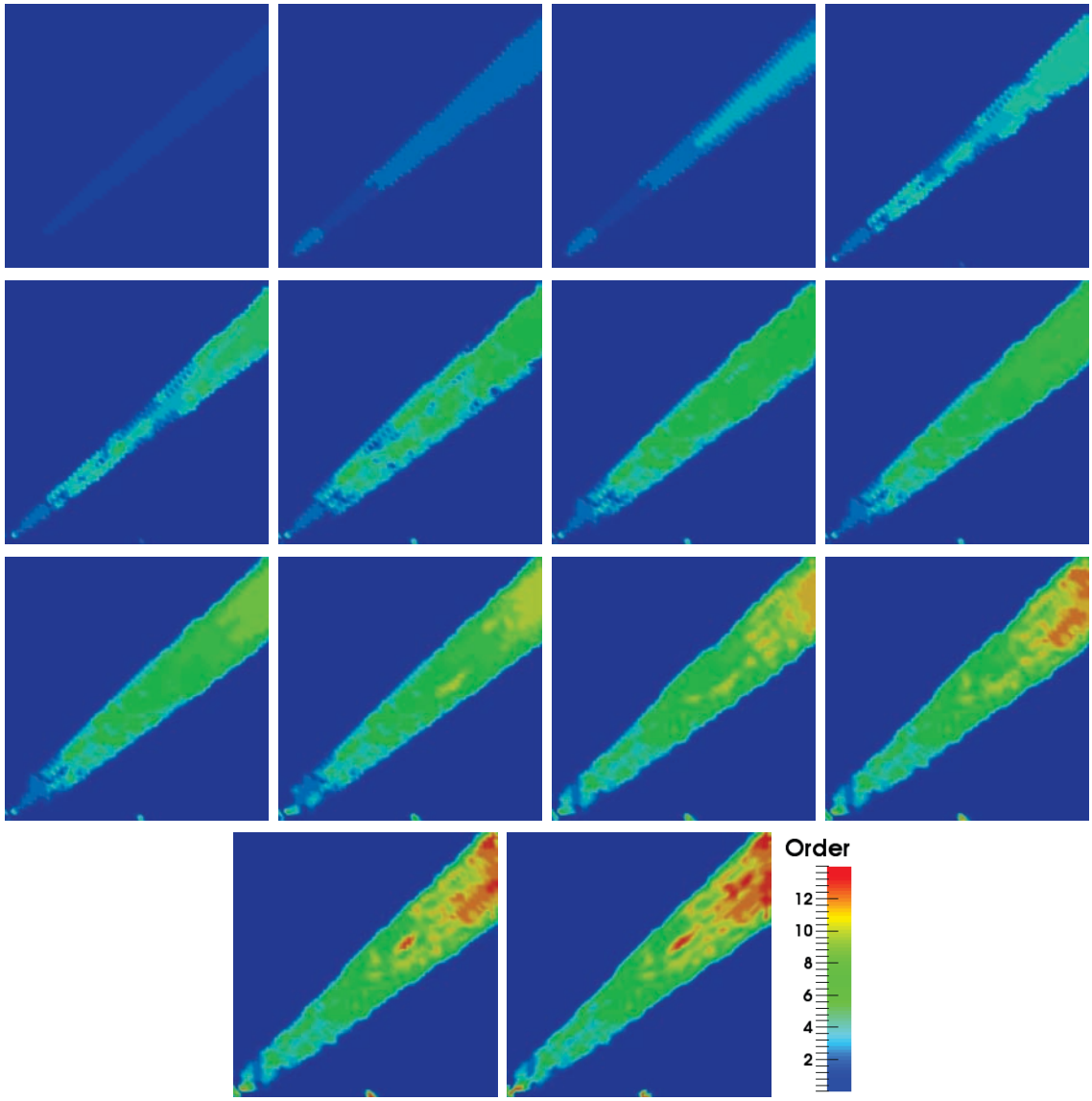

Figure 5.3. This shows the polynomial order of the stochastic Galerkin expansion over fourteen levels of refinement for the convection-diffusion problem using the element based indicator and marking strategy. Plot of the L^2 convergence for this sequence can be seen in Fig. 5.1 labeled "Element".

5.1.2 Nodal Based Refinement

To test the nodal based error indicators we used the MARKing strategy

$$\text{MARK\_PERCENT}(\mathcal{E},\, 0.1) :: \text{MARK\_PERCENT}(\mathcal{E},0.025)$$

with an initial polynomial order of one across the physical domain. An initial polynomial order of one is required to compute the nodal indicator in Eq. 4.22. The green line in Fig. 5.2 shows the L^2 convergence measured as a function of the number of unknowns. Note that the convergence is slightly slower than using the element based indicators. However, it remains faster than uniform refinement. After eight levels of refinement the uniform scheme achieves an error around 3×10^{-3} requiring greater than $35,000$ unknowns. The adaptive scheme after eight levels of refinement requires only around $17,000$ unknowns, about half as many as the uniform refinement, yet achieves the same level of accuracy. Of course this must be balanced with the increased cost of solving the adaptive system multiple times. Figure 5.4 shows the color plots of the nodally refined polynomial order. Again the images show that the refinement occurs primarily near the region with greatest uncertainty focusing on the location of the jump in the solution. These images are comparable, but different than the element based refinement scheme in Fig. 5.3.

5.2 Convection Diffusion Around a Cylinder

The second problem considered models steady-state heat transfer from a solid body into a fluid with variable but spatially constant thermal conductivity. Mathematically this is represented as a problem with two domains, a small interior cylinder represented as a circle in two dimensions, surrounded by a fluid whose flow is modeled as a constant left to right convection over the cylinder. Figure 5.5 shows a plot of the mesh used containing 3240 cells and 3375 nodes, and a color plot showing the fluid domain (blue) and the solid cylinder (red). To be precise, the steady-state principal equations are

$$-\nu\nabla^2 T + \begin{bmatrix}1\\0\end{bmatrix} \cdot \nabla T = 0 \quad x \in \Omega_{Fluid}, \tag{5.3}$$

$$-\nabla^2 T = 1 \quad x \in \Omega_{Solid}. \tag{5.4}$$

Note that the flow is from left to right across the fluid domain (there is no flow or advection in the solid cylinder). The boundary conditions are natural on the inflow and outflow boundary (left and right) and the temperature is set to zero on the upper and lower boundaries. Heat is generated in the cylinder using a constant source term of one, the heat is taken away from the cylinder using the advection and diffusion in the fluid domain. The strength of the diffusion in the fluid is defined by the stochastic parameter ν which is spatially constant but allowed to vary uniformly from $[0.02, 2.0]$.

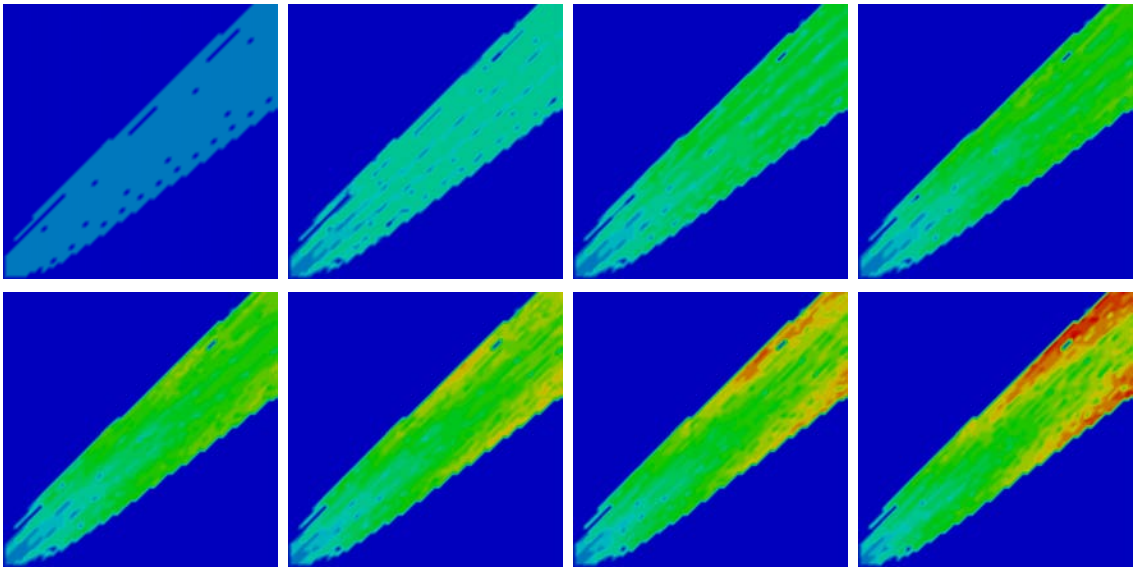

Figure 5.4. This shows the polynomial order of the stochastic Galerkin expansion over eight levels of refinement for the convection-diffusion problem using the nodal indicator and marking strategy. Plot of the L^2 convergence for this sequence can be seen in Fig. 5.1 labeled "Nodal" (note the scale on the color maps is the same as in Fig. 5.3).

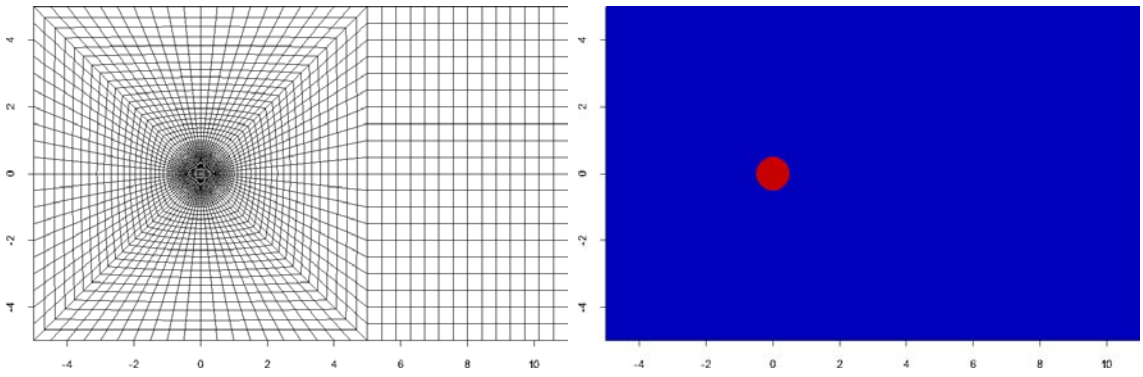

Figure 5.5. Plot of the mesh (left) and subdomains (right) for the cylinder in cross flow problem. The subdomain plot shows the "fluid" (blue) and the solid cylinder (red).

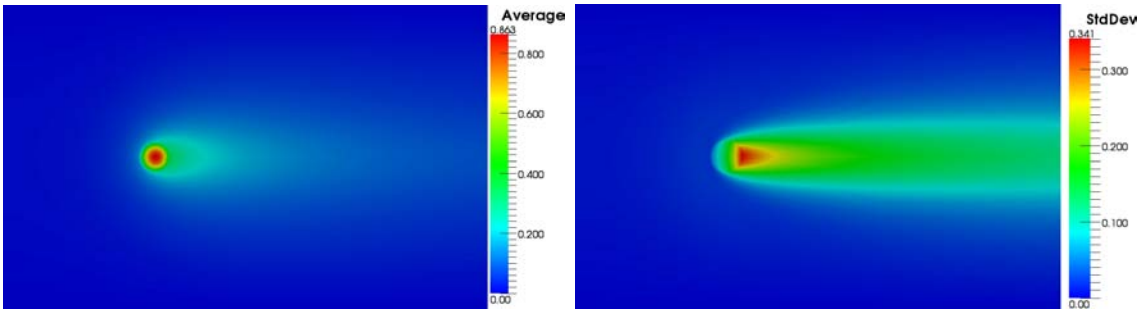

Figure 5.6. Color map of the average (left) and standard deviation (right) for the cylinder in cross flow convection diffusion problem.

The reference stochastic Galerkin solution uses a 15^{th} order Legendre polynomial expansion. The mean and standard deviation can be seen in Fig. 5.6. The blue line in Fig. 5.7 shows the convergence as teh stochastic basis is uniformly refined. Again this demonstrates the expected exponential convergence.

5.2.1 Element Based Refinement

The refinement using the element based indicators used the MARKing strategy

$$\text{MARK\_PERCENT}(\mathcal{E},\ 0.01)\ ::\ \text{MARK\_AVERAGE}(\mathcal{E},5.0)$$

with an initial polynomial order of one for the low order solution. Unlike the previous case, the REFINE step incremented the stochastic polynomial order by two (instead of one) if an element was marked for refinement. Six levels of refinement were performed, Fig. 5.8 shows a sequence of color maps of the polynomial order across the physical domain for each level of refinement. The red line in Fig. 5.7 demonstrates that this adaptive scheme is more accurate then uniform refinement. Ultimately it achieves a greater level of accuracy then uniform refinement requiring about 20% fewer unknowns. The color plots show how higher-order expansions are required down-stream of the cylinder with relatively little refinement required upstream. It is also interesting to note that in a small region immediately upstream of the cylinder little to no refinement is needed.

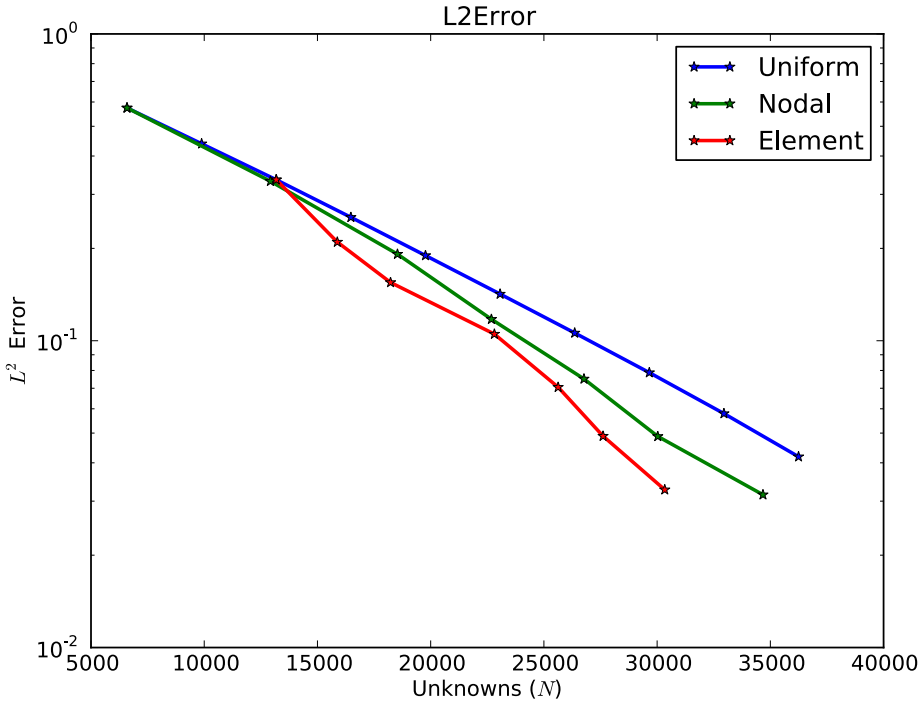

Figure 5.7. The plot shows the error as a function of total number of unknowns for refinement of both uniform and spatially varying stochastic expansions for the cylinder in cross flow problem.

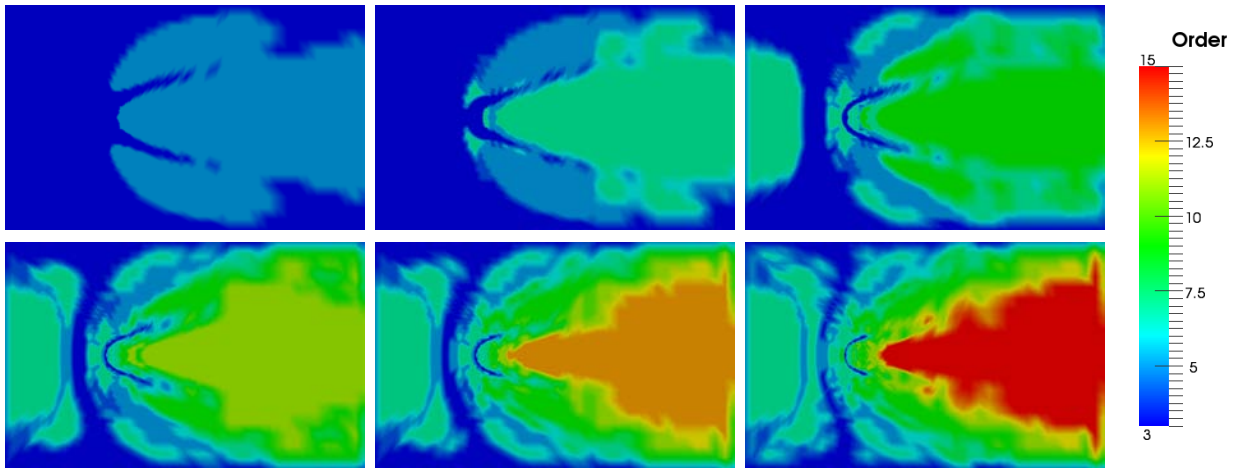

Figure 5.8. This shows the polynomial order of the stochastic Galerkin expansion for the cylinder in cross flow problem over six levels of refinement using the element indicator and marking strategy. Plot of the L^2 convergence for this sequence can be seen in Fig. 5.7 labeled "Element".

5.2.2 Nodal Based Refinement

The refinement using the nodal based indicators used the MARKing strategy

$$\text{MARK\_PERCENT}(\mathcal{E},\, 0.1) :: \text{MARK\_PERCENT}(\mathcal{E},0.025)$$

with an initial polynomial order of one. The REFINE step incremented the stochastic polynomial order by one (unlike for the element based approach above) if an element was marked for refinement. Six levels of refinement were performed, Fig. 5.9 shows a sequence of color maps of the polynomial order across the physical domain for each level of refinement. The green line in Fig. 5.7 demonstrates that this adaptive scheme is more accurate then uniform refinement, but less accurate then the element based scheme.

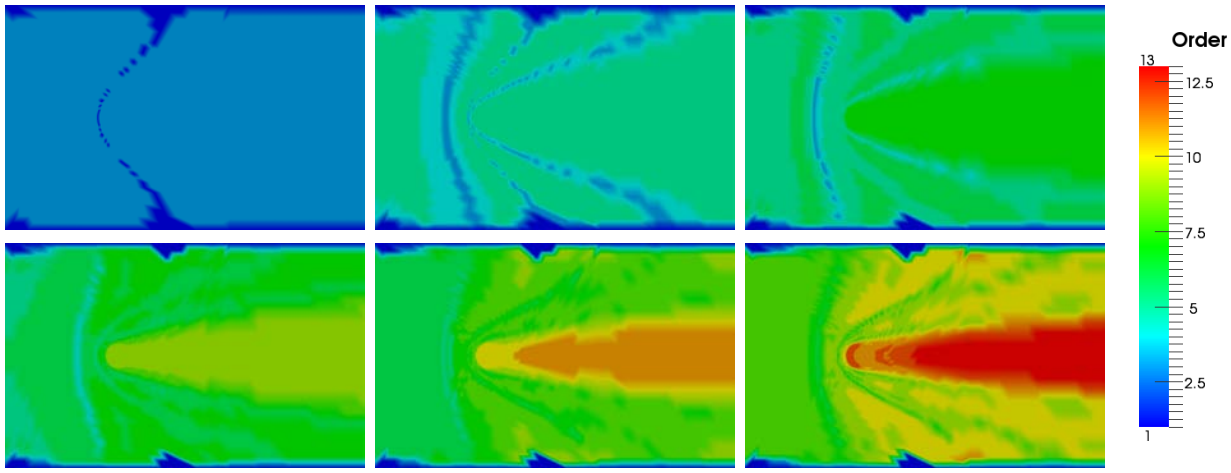

Figure 5.9. This shows the polynomial order of the stochastic Galerkin expansion for the cylinder in cross flow problem over six levels of refinement using the nodal indicator and marking strategy. Plot of the L^2 convergence for this sequence can be seen in Fig. 5.7 labeled "Nodal".

Chapter 6

Conclusions

We have demonstrated a novel spatially varying approach exploiting the finer granularity of embedded uncertainty quantification methods. This method does not require a tensor product between the stochastic approximation and the deterministic finite element basis as is typically used in both black-box and embedded uncertainty propagation methods. Focusing on polynomial chaos (PC) expansions as the stochastic basis, we developed an adaptive method for forming these discretizations. This was demonstrated on two convection-diffusion problems including a high Peclet number 2D convection diffusion problem and a model heat transfer problem. These demonstrated that for problems with spatial locality this method can reduce the number of unknowns required to achieve a similar amount of accuracy using uniform stochastic refinement.

This report demonstrated that the spatially varying discretizations can be useful for uncertainty quantification. However, much work needs to be done to refine the method before it is practically useful. In particular adaptive refinement continues to prove difficult. This is because of the challenge of developing high quality cheap error indicators and robust marking schemes. Some specific options are to use adjoint-based error estimates. This has the advantage of localizing both because of the forward problem but also because of the possibly local quantity of interest. These methods often localize in space for adaptive mesh refinement. Another issue that must be considered is implementation for a number of parameter dimensions greater then one. This could lead to substantial improvement because the shear number of unknowns required for the full stochastic expansion is extremely large. Finally, in order to be practical, good load balancing algorithms and linear solvers must be developed for a fast parallel implementation.

References

[1] D. Braess. *Finite elements*. Cambridge University Press, 2007.

[2] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*, volume 15. Springer, 2008.

[3] L. Chen, M. Holst, and J. Xu. Convergence and optimality of adaptive mixed finite element methods. *Mathematics of Computation*, 78(265):35–53, 2009.

[4] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Classics in applied mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2002.

[5] R. Codina. On stabilized finite element methods for linear systems of convectiondiffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 188(1-3):61 – 82, 2000.

[6] W. Dörfler. A convergent adaptive algorithm for poissons equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124, 1996.

[7] H. C. Elman, C. W. Miller, E. T. Phipps, and R. S. Tuminaro. Assessment of collocation and galerkin approaches to linear diffusion equations with random data. *International Journal for Uncertainty Quantification*, 1(1), 2011.

[8] G. Fishman. *Monte Carlo: concepts, algorithms, and applications*. Springer, 1996.

[9] R. G. Ghanem and P. D. Spanos. *Stochastic finite elements: a spectral approach*. Dover Publications, Inc., 2003.

[10] O. P. Le Maître and O. M. Knio. *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*. Springer, 2010.

[11] O. P. Le Maıtre, H. N. Najm, R. G. Ghanem, and O. M. Knio. Multi-resolution analysis of wiener-type uncertainty propagation schemes. *Journal of Computational Physics*, 197(2):502–531, 2004.

[12] F. Nobile, R. Tempone, and C. G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2411–2442, 2008.

[13] F. Nobile, R. Tempone, and C. G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345, 2008.

[14] T. M. Smith, J. N. Shadid, R. P. Pawlowski, E. C. Cyr, and P. D. Weber. Reactor core sub-assembly simulations using a stabilized finite element method. 2011.

[15] P. Solin, K. Segeth, and I. Dolezel. *Higher-order finite element methods*, volume 41. Chapman and Hall/CRC, 2003.

[16] X. Wan and G. E. Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209(2):617–642, 2005.

[17] D. Xiu and G. Karniadakis. The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations. *SIAM J. Sci. Comput.*, 24(2), 2002.

DISTRIBUTION:

1	MS 0899	Technical Library, 9536 (electronic copy)
1	MS 0359	D. Chavez, LDRD Office, 1911


Sandia National Laboratories