SAND2011-1487C

# Uncertainty Quantification and Stochastic Dimension Reduction for Complex Coupled Systems

**Eric Phipps (etphipp@sandia.gov),**
**Roger Pawlowski, John Red-Horse**
**Sandia National Laboratories,**

**Roger Ghanem, Ramakrishna Tipireddy**
**University of Southern California, and**

**Maarten Arnst**
**University of Liège, Belgium**

**Workshop on Uncertainty Quantification for Multiphysics and Multiscale Systems**

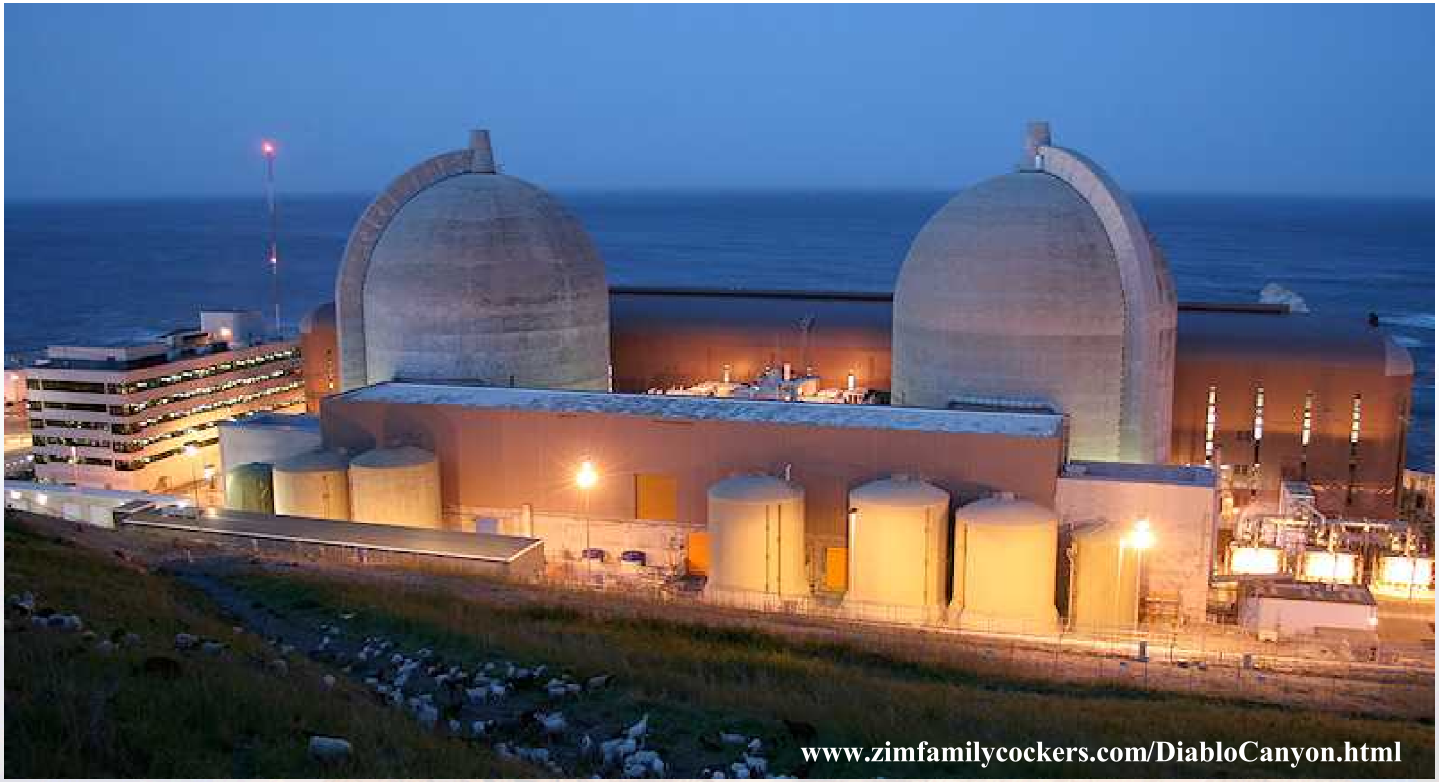
**March 7-8, 2011**

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.

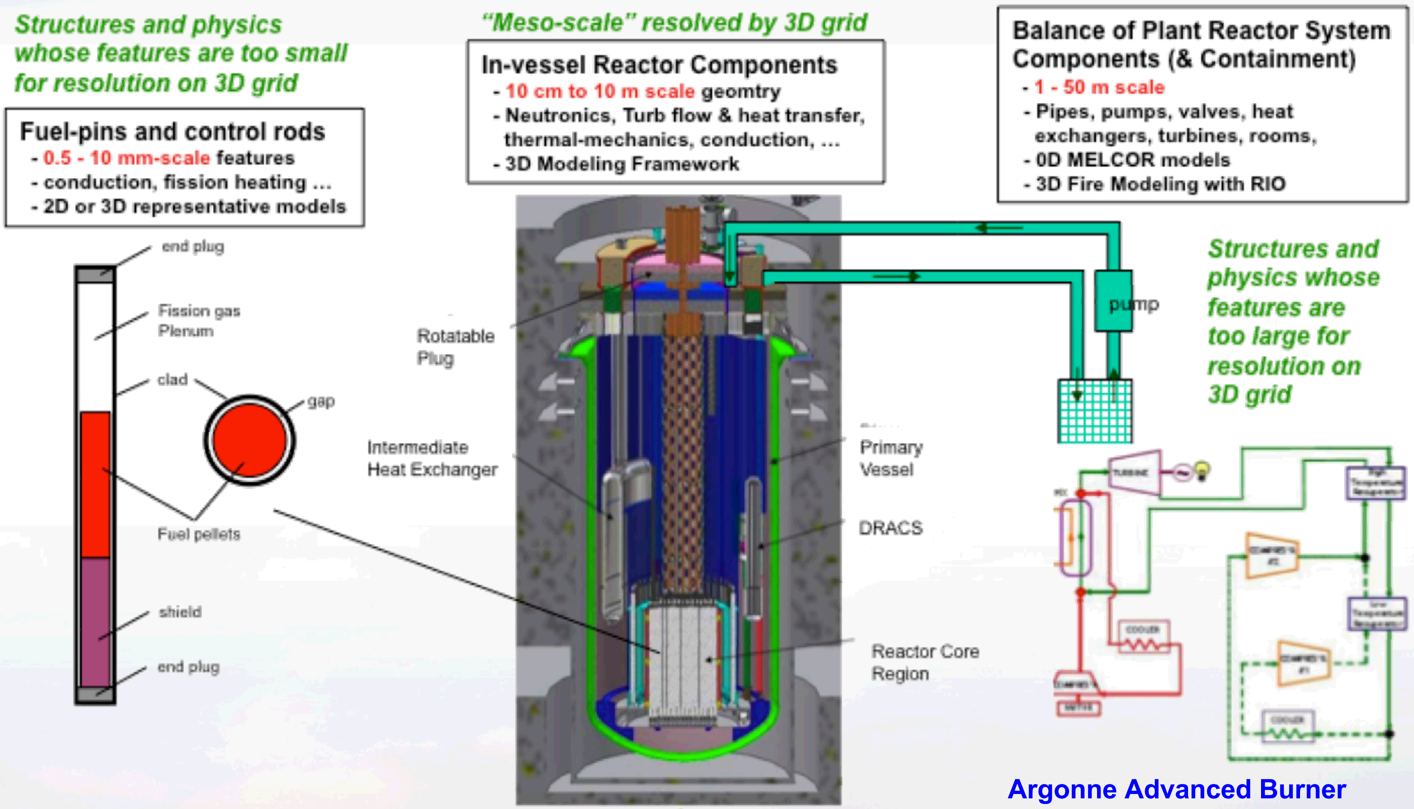
Sandia National Laboratories

# Uncertainty Quantification for Complex Coupled Systems

- Address *some* of the mathematical and computational challenges in predictive simulation of complex coupled systems such as…



# Challenges for UQ of Complex Coupled Systems

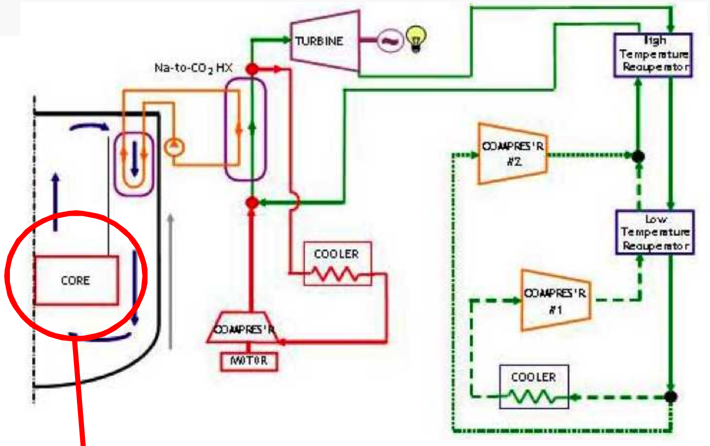



Argonne Advanced Burner Reactor Preconceptual Design

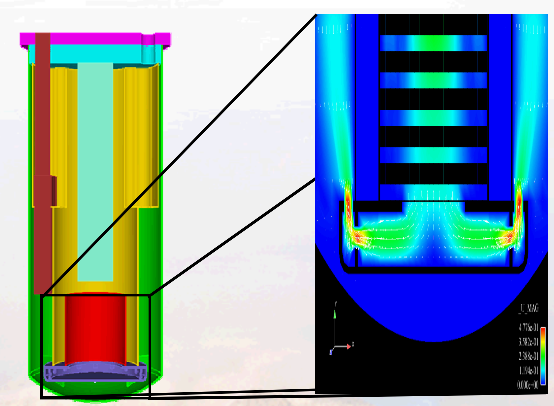
- Predictive simulation must capture critical couplings
- Coupling physics often necessitates reduction in model **fidelity**
- Reducing fidelity introduces additional uncertainty (component & interface)
- Strong coupling adds new dimensions of uncertainty to all components
- Cost of uncertainty quantification grows dramatically **with stochastic dimension**

# Multi-Physics, Multi-Fidelity, Heterogeneous Uncertainty Quantification Approach

- Component-level uncertainty propagation via general expansion methodology
  - Network or multi-physics component
- Stochastic dimension reduction at component interfaces
- Strongly coupled solver technology for coupled stochastic problems
- Stochastic up-scaling for low-fidelity models
- Stochastic sensitivities with respect to system components




Low-fidelity Network Plant Model



High-fidelity Multi-physics Component Model (Core)

Graphics courtesy: Rod Schmidt, BRISC project

# General Stochastic Expansion Uncertainty Quantification Framework

- Steady-state spatially finite-dimensional stochastic problem:

$$\text{Find } u(\xi) \text{ such that } f(u,\xi)=0,\ \xi:\Omega\rightarrow\Gamma\subset R^M,\ \text{density } \rho$$

- General stochastic expansion approximation:

$$Z=\text{span}\{\Psi_i : i=0,\ldots,P\}\subset L^2_\rho(\Gamma)\rightarrow u(\xi)\approx\hat{u}(\xi)=\sum_{i=0}^{P}u_i\Psi_i(\xi)$$

**Intrusive Stochastic Galerkin (SG), a.k.a. (Generalized) Polynomial Chaos**

- Orthogonal polynomial basis of total order at most N

$$\langle\Psi_i\Psi_j\rangle\equiv\int_\Gamma\Psi_i(x)\Psi_j(x)\rho(x)dx=\delta_{ij}\langle\Psi_i^2\rangle,\quad i,j=0,\ldots,P_{SG}$$

- Galerkin Projection

$$F_i(u_0,\ldots,u_P)\equiv\frac{1}{\langle\Psi_i^2\rangle}\int_\Gamma f(\hat{u}(x),x)\Psi_i(x)\rho(x)dx=0,\ i=0,\ldots,P_{SG}$$

**Non-Intrusive Polynomial Chaos (NIPC)**

$$u_i=\frac{1}{\langle\Psi_i^2\rangle}\int_\Gamma u(x)\Psi_i(x)\rho(x)dx\approx\frac{1}{\langle\Psi_i^2\rangle}\sum_{k=0}^{Q}w_k u_k\Psi_i(x_k),\ f(u_k,x_k)=0,\ i=0,\ldots,P_{SG},\ k=0,\ldots Q$$

**Non-Intrusive Stochastic Collocation (SC)**

- Interpolatory polynomial basis defined by collocation points $\{x_j\in\Gamma : j=0,\ldots,P_{SC}\}$

$$\Psi_i(x_j)=\delta_{ij},\quad f(u_j,x_j)=0,\quad i,j=0,\ldots,P_{SC}$$

# Two Challenges for Intrusive SG

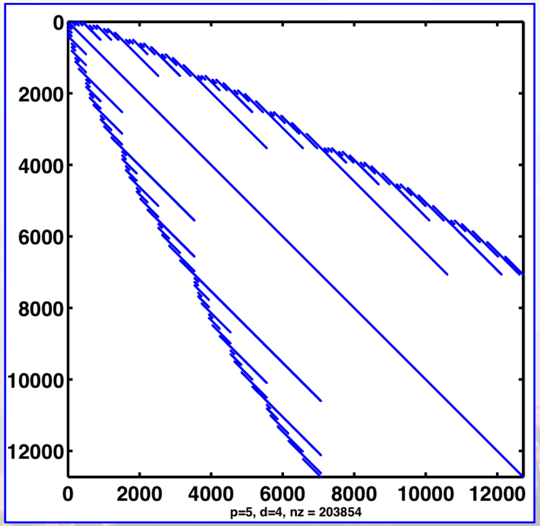
- **Generating SG residual & Jacobian entries in complex simulation codes:**

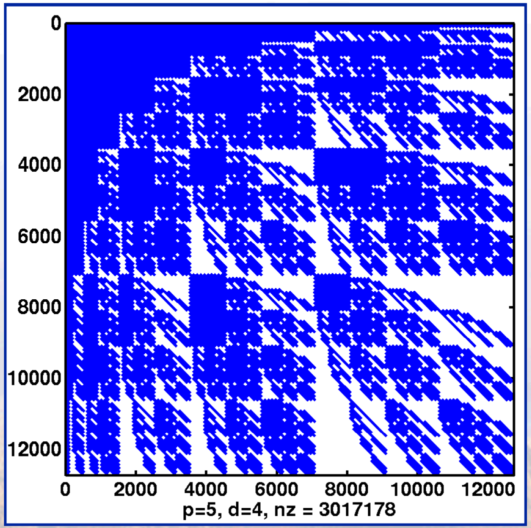
$$F_i = \int_\Gamma f(\hat{u}(y), y)\psi_i(y)\rho(y)dy, \quad \langle \cdot \rangle \equiv \int_\Gamma \cdot\, \rho(y)dy,$$

$$\frac{\partial f}{\partial u}(\hat{u}(y), y) \approx \sum_{k=0}^{P} J_k \psi_k(y), \; J_k = \frac{1}{\langle \psi_k^2 \rangle} \int_\Gamma \frac{\partial f}{\partial u}(\hat{u}(y), y)\psi_k(y)\rho(y)dy,$$

$$\Longrightarrow \frac{\partial F_i}{\partial u_j} = \int_\Gamma \frac{\partial f}{\partial u}(\hat{u}(y), y)\psi_i(y)\psi_j(y)\rho(y)dy \approx \sum_{k=0}^{P} J_k \langle \psi_i \psi_j \psi_k \rangle$$

- **Solving resulting fully-coupled spatial-stochastic problem:**



# Trilinos Package Stokhos

- Tools for generating SG residual and Jacobian entries
  - Polynomial basis definition
  - Quadrature methods
  - Triple product tensors
  - Expansion/approximation methods for nonlinear terms
  - Automatic differentiation (via Sacado AD Trilinos package)
- Tools for forming and solving SG linear systems
  - Product vectors
  - SG matrix operators
  - Preconditioning methods
- Nonlinear application code interfaces
  - Nonlinear solver
  - Time integration
  - Optimization
  - …

Trilinos

http://trilinos.sandia.gov

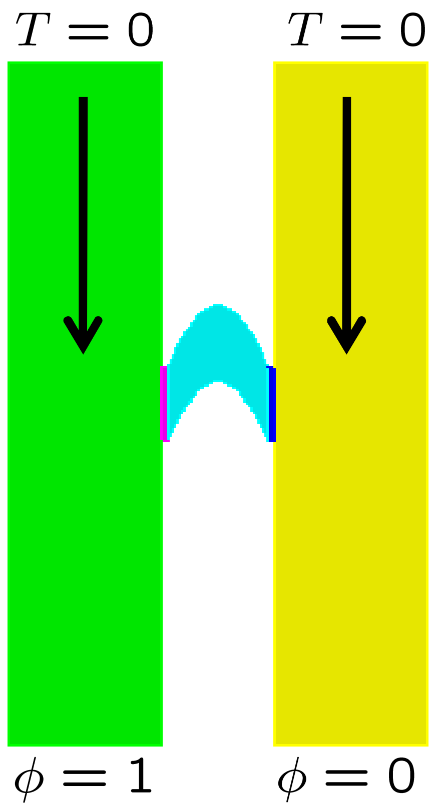
# Generating SG Residual and Jacobian Coefficients via Automatic Differentiation (AD)

- AD relies on known derivative formulas for all intrinsic operations plus chain rule
  - Implemented in C++ via operator overloading
  - Template your code on scalar type, replacing double's with AD type
- Similar approach possible for SG expansion

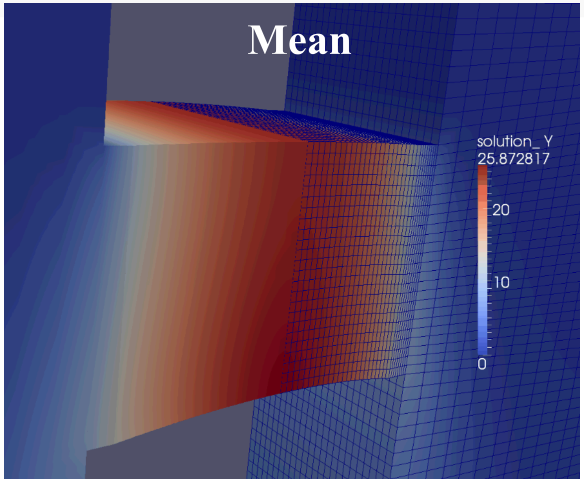
$$a = \sum_{i=0}^{P} a_i \psi_i, \; b = \sum_{j=0}^{P} b_j \psi_j, \; c = ab \approx \sum_{k=0}^{P} c_k \psi_k, \; c_k = \sum_{i,j=0}^{P} a_i b_j \frac{\langle \psi_i \psi_j \psi_k \rangle}{\langle \psi_k^2 \rangle}$$

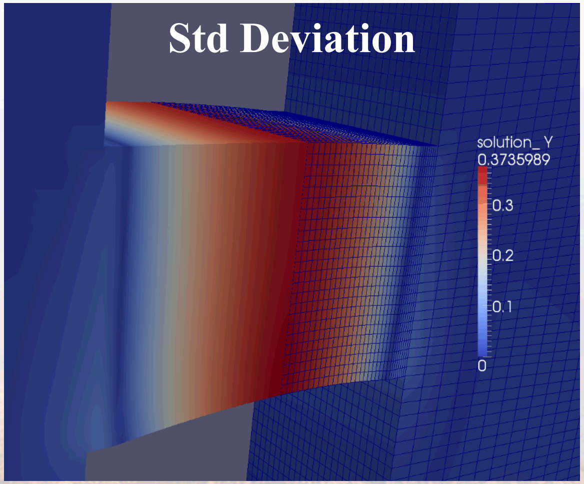
- Transcendental operations more difficult (see Debusschere *et al*, SISC, 2004)
  - Taylor series
  - Time integration
  - Sparse-grid quadrature
  - Research to be done here…(see Kevin Long's talk)
- Enables "easy" incorporation of SG calculations in codes that support AD
  - Stokhos provides Sacado "AD" data type for SG calculations

# UQ in an Electromagnetic Contact Demonstration SNL Albany Code (Salinger *et al*)




$$
\begin{aligned}
-\nabla \cdot \sigma \nabla \phi &= 0 \\
-\nabla \cdot \kappa \nabla T - \mathrm{v} \cdot \nabla T &= \sigma (\nabla \phi)^2 \\
\sigma(T) &= \sigma_0 / [1 + \beta (T - T_0)]
\end{aligned}
$$




Sandia National Laboratories

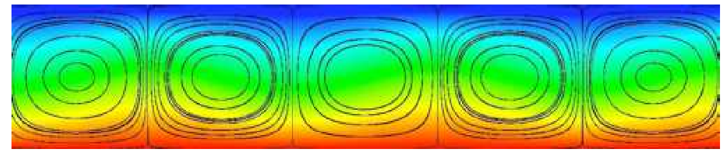
# Recent Advancements

- New sparse triple product data structures and fill
  - Drastically reduce calculation time
  - Data-structures optimized for linear solvers
- New KL-based Jacobian operator decomposition
  - Drastically reduces computational complexity of SG mat-vecs for nonlinear problems
  - Much analysis to be done here
- Parallelization over stochastic DOFs
  - Only for linear solves, not yet for residual/Jacobian fill
  - Can rebalance using Isorropia/Zoltan to minimize cost of mat-vecs
  - Stay tuned…
- Preliminary implementation of Stokhos/AD overloaded operators on Nvidia GPUs (via CUDA)
  - Intrusive SG may be an ideal algorithm for massive on-node threading
  - Much work to do from a software point of view to do
- New solver and preconditioner approaches
  - See Rama's talk later

# Coupled Nonlinear Systems

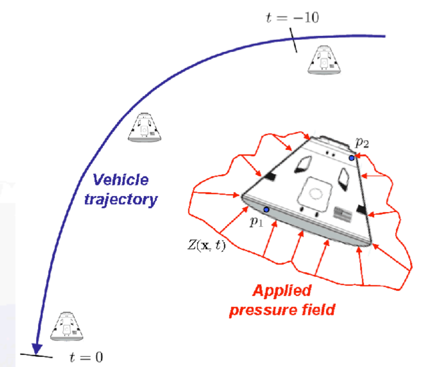
- Shared-domain multi-physics coupling
  - Equations coupled at each point in domain

$$\mathcal{L}_1(u_1(x), u_2(x)) = 0$$
$$\mathcal{L}_2(u_1(x), u_2(x)) = 0$$



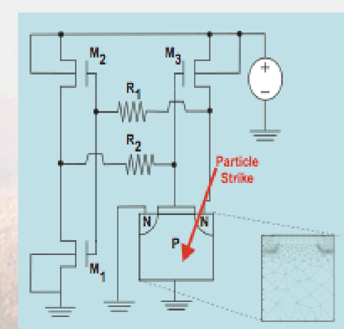
- Interfacial multi-physics coupling
  - Equations are coupled through boundaries

$$\mathcal{L}_1(u_1(x), v_2(x_2)) = 0, \quad v_2(x_2) = \mathcal{G}_2(u_2(x_2)), \quad x_2 \in \Gamma_2$$
$$\mathcal{L}_2(v_1(x_1), u_2(x)) = 0, \quad v_1(x_1) = \mathcal{G}_1(u_1(x_1)), \quad x_1 \in \Gamma_1$$




- Network coupling
  - Equations are coupled through a set of scalars

$$\mathcal{L}_1(u_1(x), v_2) = 0, \quad v_2 = \mathcal{G}_2(u_2)$$
$$\mathcal{L}_2(v_1, u_2(x)) = 0, \quad v_1 = \mathcal{G}_1(u_1)$$



# Finite Dimensional Coupled Nonlinear Systems

- All three forms can be written after discretization

$$f_1(u_1, v_2) = 0, \quad u_1 \in \mathbb{R}^{n_1}, \quad v_2 = g_2(u_2) \in \mathbb{R}^{m_2}, \quad f_1 : \mathbb{R}^{n_1+m_2} \to \mathbb{R}^{n_1}$$
$$f_2(v_1, u_2) = 0, \quad u_2 \in \mathbb{R}^{n_2}, \quad v_1 = g_1(u_1) \in \mathbb{R}^{m_1}, \quad f_2 : \mathbb{R}^{m_1+n_2} \to \mathbb{R}^{n_2}$$

  – Shared-domain multi-physics coupling:

$$m_1, m_2 \sim n_1, n_2$$

  – Interfacial multi-physics coupling:

$$1 \ll m_1, m_2 \ll n_1, n_2$$

  – Network coupling:

$$m_1, m_2 \sim 1$$

# Solution Strategies

- Successive substitution (Picard, Gauss-Seidel, …)
  - Appropriate for all three forms of coupled systems
  - Segregated solves

$$\text{Solve}\quad f_1(u_1^{(l+1)}, v_2^{(l)}) = 0 \quad \text{for} \quad u_1^{(l+1)}$$

$$\text{Solve}\quad f_2(v_1^{(l+1)}, v_2^{(l+1)}) = 0 \quad \text{for} \quad u_2^{(l+1)}$$

- Nonlinear elimination
  - Practical only for network coupling
  - Segregated solves

$$v_1 - g_1(u_1(v_2)) = 0 \quad \text{s.t.} \quad f_1(u_1, v_2) = 0$$

$$v_2 - g_2(u_2(v_1)) = 0 \quad \text{s.t.} \quad f_2(v_1, u_2) = 0$$

- Full Newton (including JFNK)
  - Appropriate for all three, but the most challenging to implement
  - No segregated solves

$$\begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial v_2}\frac{\partial g_2}{\partial u_2} \\ \frac{\partial f_2}{\partial v_1}\frac{\partial g_1}{\partial u_1} & \frac{\partial f_2}{\partial u_2} \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix} = - \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

# Stochastic Coupled Nonlinear Systems

- Introduce random variables:

$$f_1(u_1(\xi), v_2(\xi), \xi_1) = 0, \quad v_2(\xi) = g_2(u_2(\xi), \xi_2), \quad \xi = (\xi_1, \xi_2)$$
$$f_2(v_1(\xi), u_2(\xi), \xi_2) = 0, \quad v_1(\xi) = g_1(u_1(\xi), \xi_2), \quad |\xi_1| = M_1, \quad |\xi_2| = M_2$$

- Introduce stochastic expansion approximation:

$$\hat{u}_i(\xi) = \sum_{j=0}^{P} u_{i,j}\Psi_j(\xi), \quad \hat{v}_i(\xi) = \sum_{j=0}^{P} v_{i,j}\Psi_j(\xi)$$

Intrusive

$$\frac{1}{\langle \Psi_j^2 \rangle}\langle f_1(\hat{u}_1(\xi), \hat{v}_2(\xi), \xi_1)\Psi_j(\xi)\rangle = 0$$
$$\frac{1}{\langle \Psi_j^2 \rangle}\langle f_2(\hat{v}_1(\xi), \hat{u}_2(\xi), \xi_2)\Psi_j(\xi)\rangle = 0$$

Non-Intrusive

$$u_{i,j} = \sum_{k=0}^{Q} w_k u_i^k \Psi_j(\xi^k) \qquad v_{i,j} = \sum_{k=0}^{Q} w_k v_i^k \Psi_j(\xi^k)$$
$$\text{s.t.} \quad f_1(u_1^k, v_2^k, \xi_1^k) = 0, \quad f_2(v_1^k, u_2^k, \xi_2^k) = 0$$

$$F_1(U_1, V_2) = 0, \quad V_2 = G_2(U_2)$$
$$F_2(V_1, U_2) = 0, \quad V_1 = G_1(U_1)$$

- Corresponding coupled system for stochastic DOFs
  - Direct analog of the deterministic solution strategies

# Curse of Dimensionality

- Because system is coupled, each component must compute approximation over full stochastic space:

$$\hat{u}_1(\xi_1, \xi_2) = \hat{u}_1(\hat{v}_2(\xi_1, \xi_2), \xi_1)$$

  – For segregated methods, requires solving sub-problems of larger dimensionality, e.g.,

$$\text{Solve} \quad \frac{1}{\langle \Psi_j^2 \rangle} \langle f_1(\hat{u}_1(\xi), \hat{v}_2(\xi), \xi_1) \Psi_j(\xi) \rangle = 0 \quad \text{for} \quad \{u_{1,j}\} \quad \text{given} \quad \{v_{2,j}\}$$

  – Adding more components, or more sources of uncertainty in other components, increases cost of each sub-problem

- For network problems, use interface to define new random variables

$$\hat{u}_1(\xi_1, \xi_2) = \sum_{j=0}^{P} u_{1,j} \Psi_j(\xi_1, \xi_2) \longrightarrow \tilde{u}_1(\eta_2, \xi_1) = \sum_{j=0}^{\tilde{P}_1} \tilde{u}_{1,j} \Phi_j(\eta_2, \xi_1), \quad \eta_2 = \hat{v}_2(\xi_1, \xi_2)$$

- Challenges:
  – Measure transformation; Generating polynomials orthogonal w.r.t. joint PDF of $\eta_2$ and $\xi_1$
  – Extending approach to other types of coupling

# Dimension Reduction in Multi-Physics Problems

- Consider the shared-domain multi-physics problem:

$$\begin{aligned} f_1(\hat{u}_1(\xi), \hat{u}_2(\xi), \xi_1) &= 0 \\ f_2(\hat{u}_1(\xi), \hat{u}_2(\xi), \xi_2) &= 0 \end{aligned}, \quad \hat{u}_i(\xi) = \sum_{k=0}^{P} u_{i,k} \Psi_k(\xi)$$

- Introduce truncated Karhunen-Loeve (KL) decomposition

$$\tilde{u}(\eta(\xi)) = u_0 + \sum_{k=0}^{\tilde{M}} \sqrt{\lambda_k} \varphi_k \eta_{(k)}(\xi), \quad \eta = (\eta_{(1)}, \dots, \eta_{(\tilde{M})})$$

- KL eigenvectors/values (eigenvalue problem):

$$(CC^T)\varphi_k = \lambda_k \varphi_k, \quad C = [u_1 \dots u_P]$$

- KL random variables (given by PCE):

$$\eta_{(k)}(\xi) = \frac{\varphi_k^T(\hat{u} - u_0)}{\sqrt{\lambda_k}} = \sum_{l=1}^{P} \frac{\varphi_k^T u_l}{\sqrt{\lambda_k}} \Psi_l(\xi)$$

- Denote this transformation by:

$$\eta = g(\hat{u})$$

# Dimension Reduction in Multi-Physics Problems

- Applying this to each sub-problem yields:

$$f_1(\hat{u}_1(\xi), \eta_2(\xi), \xi_1) = 0, \quad \eta_2(\xi) = g_2(\hat{u}_2(\xi))$$
$$f_2(\eta_1(\xi), \hat{u}_2(\xi), \xi_2) = 0, \quad \eta_1(\xi) = g_1(\hat{u}_1(\xi))$$

- At this point, all we have done is introduce error
  - For computational savings, we also need measure transformation

$$\hat{u}_1(\xi_1, \xi_2) = \sum_{k=0}^{P} u_{1,k}\Psi_k(\xi_1, \xi_2) \longrightarrow \bar{u}_1(\xi_1, \eta_2) = \sum_{k=0}^{\tilde{P}_1} \bar{u}_{1,k}\Phi_{1,k}(\xi_1, \eta_2)$$

  - Significant savings can be realized if

$$N_1 + M_2 < N_1 + N_2 \implies \tilde{P}_1 \ll P$$

# Coupled neutron-transport and heat transfer demonstration

$$\frac{d}{dx}\left(D(T)\frac{d\Phi}{dx}\right) - \left(\Sigma_{\mathrm{a}}(T) - \nu\Sigma_{\mathrm{f}}(x,T)\right)\Phi = -s, \quad \text{with} \quad \frac{d\Phi}{dx}(0) = \frac{d\Phi}{dx}(L) = 0,$$

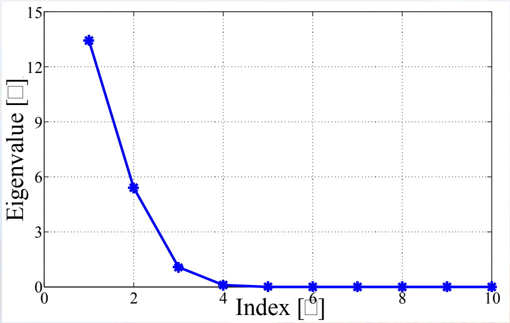
$$\frac{d}{dx}\left(k\frac{dT}{dx}\right) - h(T - T_\infty) = -q(T,\Phi), \quad \text{with} \quad \frac{dT}{dx}(0) = \frac{dT}{dx}(L) = 0,$$

$$\nu\Sigma_{\mathrm{f}}(x, T(x), \xi) = \nu\Sigma_{\mathrm{f}}^{\mathrm{ref}}(x,\xi)\sqrt{\frac{T^{\mathrm{ref}}(x)}{T(x)}}$$

$$\Longrightarrow \begin{cases} \mathcal{L}_1(T,\Phi,x,\xi) = 0 \\ \mathcal{L}_2(T,\Phi) = 0 \end{cases}$$

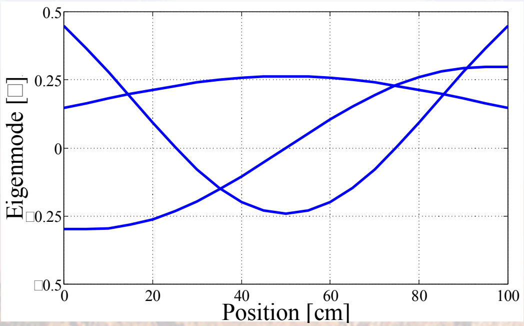
- Gauss-Seidel solution strategy
- Non-intrusive PCE

- Reference fission cross section modeled by a truncated KL expansion for a uniform random field:

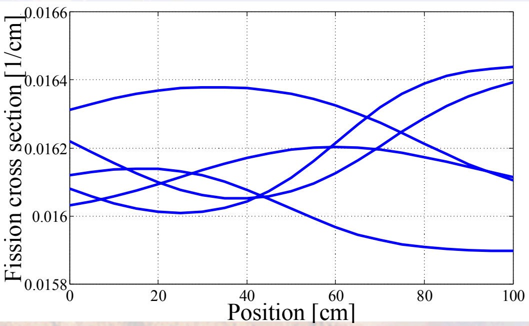



**KL Eigenvalues**

$\Longrightarrow \quad M = 3$



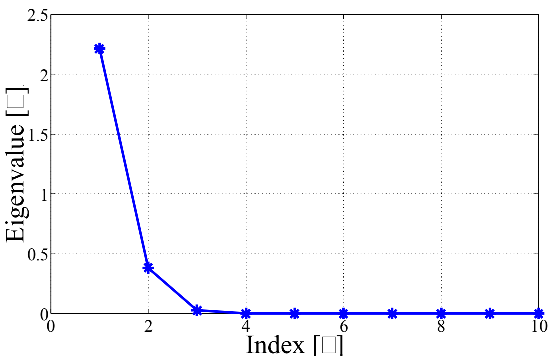

**KL Eigenmodes**




**Sample paths**

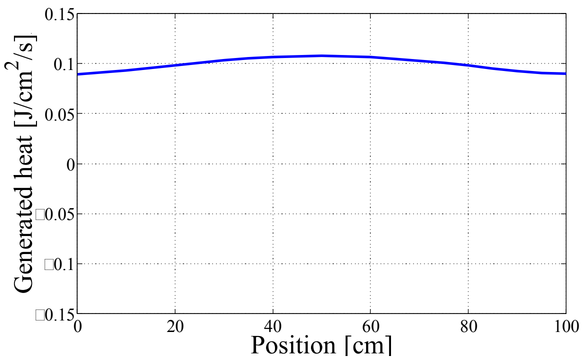
# Truncation Error Controlled by KL Terms

- KL decomposition of temperature:

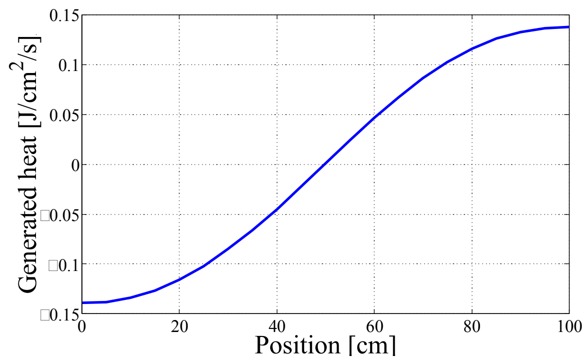



Eigenvalues

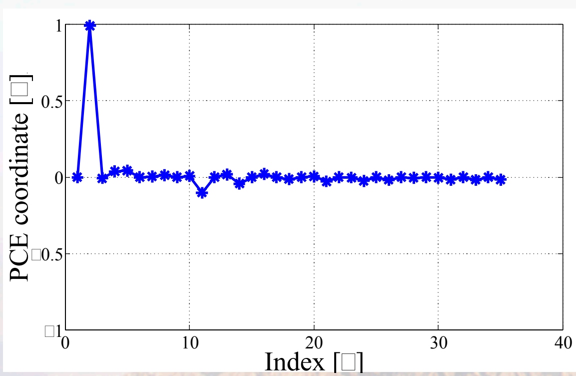
$\Longrightarrow \quad \tilde{M} = 2$



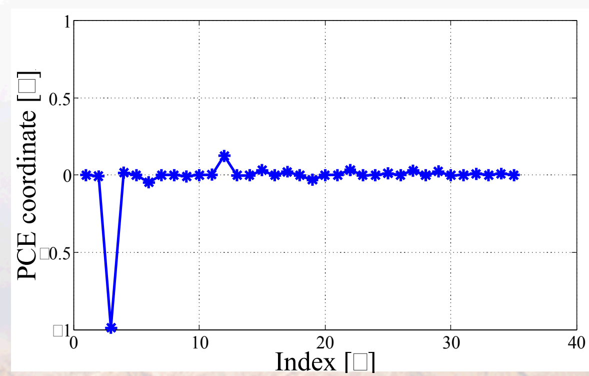

Eigenmode 1




Eigenmode 2




PCE $(M = 3,\ p = 4)$, of $\eta_1$




PCE $(M = 3,\ p = 4)$, of $\eta_2$

- Only need a few KL terms

# The Key is Measure Transformation

- Must generate orthogonal polynomials and quadrature rules for joint measure of $(\eta, \xi)$
  - Components are dependent
  - We don't have the joint measure

- What we can compute is expectation, given a quadrature rule for $\xi$

$$\int f(\eta)d\eta = \int f(\eta(\xi))d\xi \approx \sum_{k=0}^{Q} w_k f(\eta(\xi^k)) = \sum_{k=0}^{Q} w_k f(\eta^k)$$

  - Unclear how accurate of a rule this is
  - Not useful for a non-intrusive approach since it doesn't reduce the number of samples
  - Can be used to generate orthogonal polynomials via Gram-Schmidt
    - Enables intrusive approach, but too expensive

# Approach Based on Point-wise Surrogate

## (Inspired by Wan & Karniadakis, CMAME 2009)

- Generate 1-D polynomials orthogonal w.r.t. marginal density
  - Discretized Stieltjes procedure (Gautschi)
  - Use above quadrature rule to estimate necessary integrals
    - Can only generate limited order of polynomials
- Form tensor product of 1-D polynomials

$$\{\tilde{\Phi}_j(\eta_2, \xi_1) : 0 \leq j \leq \tilde{P}\}$$

  - Orthogonal w.r.t. to product-of-marginals measure
  - Generate corresponding Smolyak sparse-grid quadrature rule
- Solve sub-problem in this basis

$$f_1(\tilde{u}_1(\eta_2, \xi_1), \eta_2, \xi_1) = 0 \rightarrow \tilde{u}_1 = \sum_{j=0}^{\tilde{P}} \tilde{u}_{1,j} \tilde{\Phi}_j(\eta_2, \xi_1)$$

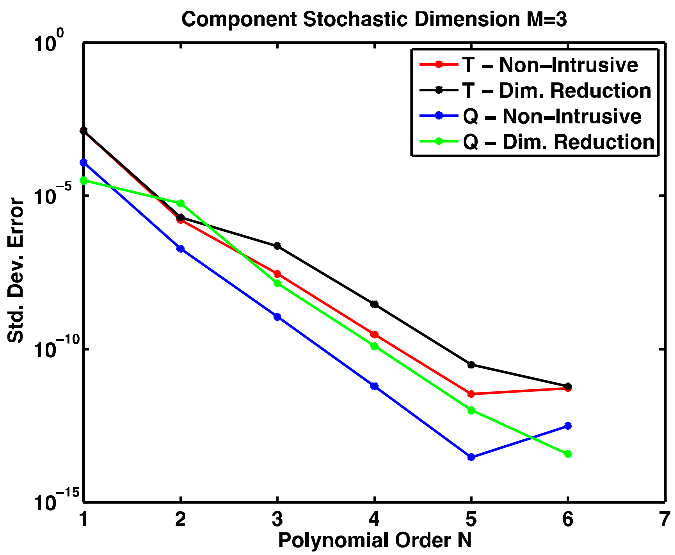
  - Intrusive or non-intrusive
- Sample this solution to compute expansion in original basis

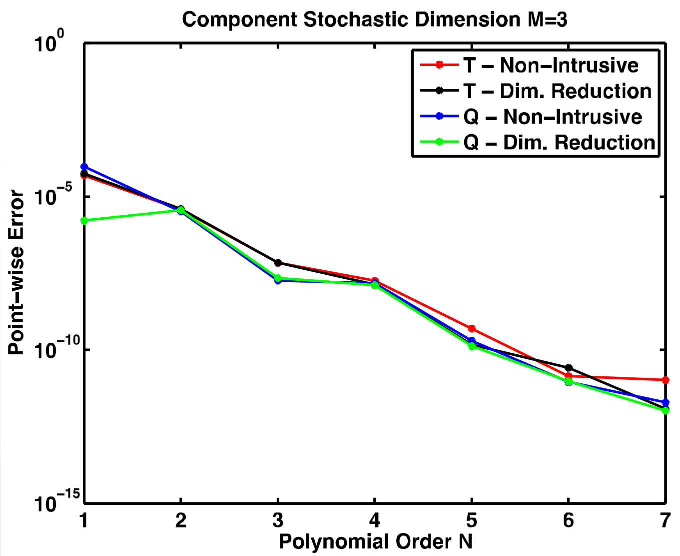
$$\hat{u}_1(\xi) = \sum_{j=0}^{P} u_{1,j} \Psi_j(\xi) \rightarrow u_{1,j} \approx \frac{1}{\langle \Psi_j^2 \rangle} \sum_{k=0}^{Q} w_k \tilde{u}_1(\eta_2(\xi^k), \xi_1^k)$$

  - Relying on point-wise convergence

# Demonstration

- Slight variant of coupled neutron/heat transfer
  - Hayes Stripling, Texas A&M, 2009 CSRI summer student
  - “Network” coupling
  - Nonlinear elimination




- Additional error in second moment

# Generating a Multi-Variate Quadrature Rule
## (Inspired by Xiao and Gimbutas 2010)

- This is all Maarten's idea…
- Start with Gram-Schmidt orthogonal basis

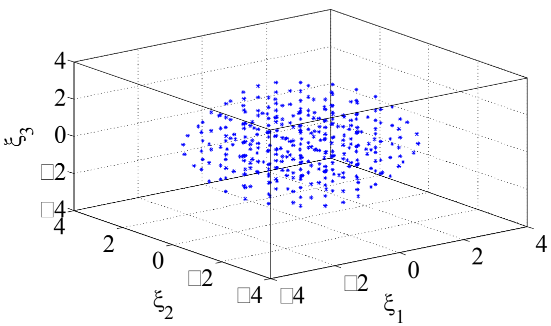
$$\{\Phi_j(\eta_2, \xi_1) : 0 \le j \le \tilde{P}\}$$

- Require quadrature rule to integrate basis exactly

$$\begin{bmatrix} \Phi_0(\eta_2^0, \xi_1^0) & \Phi_0(\eta_2^1, \xi_1^1) & \dots & \Phi_0(\eta_2^Q, \xi_1^Q) \\ \Phi_1(\eta_2^0, \xi_1^0) & \Phi_1(\eta_2^1, \xi_1^1) & \dots & \Phi_1(\eta_2^Q, \xi_1^Q) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{\tilde{P}}(\eta_2^0, \xi_1^0) & \Phi_{\tilde{P}}(\eta_2^1, \xi_1^1) & \dots & \Phi_{\tilde{P}}(\eta_2^Q, \xi_1^Q) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{\tilde{P}} \end{bmatrix} = \begin{bmatrix} \int \Phi_0(\eta_2, \xi) d(\eta_2, \xi) \\ \int \Phi_1(\eta_2, \xi) d(\eta_2, \xi) \\ \vdots \\ \int \Phi_{\tilde{P}}(\eta_2, \xi) d(\eta_2, \xi) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

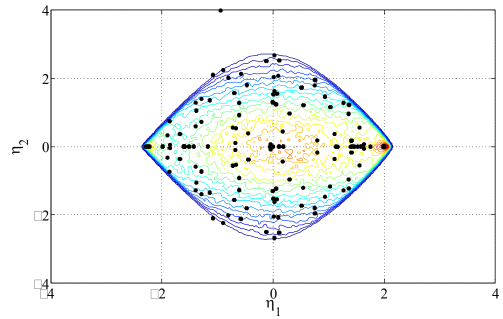
  - Underdetermined system of equations
- Start with quadrature rule for $\xi$ and $\xi \to \eta_2$ mapping
- Extract smallest set of columns with full row rank
  - QR with column pivoting
  - This defines the points
- Invert resulting linear system to obtain corresponding weights
- Use basis and quadrature rule in a non-intrusive approach

# Applied to Heat-transfer/Neutron Diffusion Problem

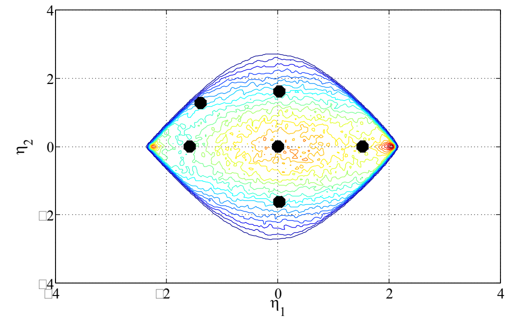



Sparse Grid Quadrature Rule

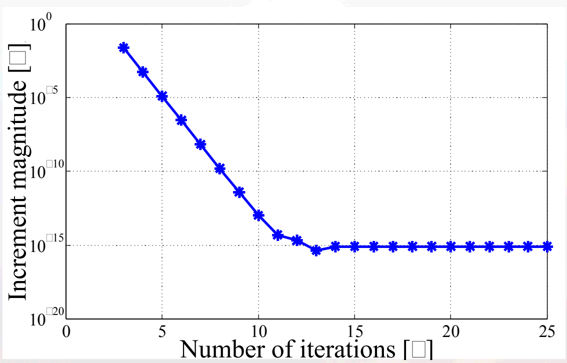
$\{\xi^k : 1 \leq k \leq 165\}$



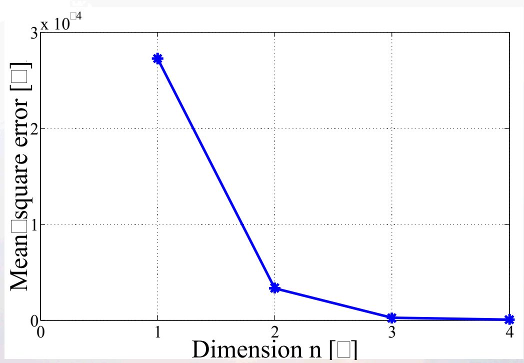

$\{\eta(\xi^k) : 1 \leq k \leq 165\}$




Reduced Quadrature Rule




Nonlinear Solver Convergence




Convergence w.r.t. reduced dimension

# Open Questions

- This QR-based quadrature approach is largely unexplored
  - Efficiency
  - Accuracy/conditioning issues
- Several unresolved questions for the Stieltjes approach
  - Accuracy of calculations of integrals in Stieltjes procedure?
  - Can this be improved by estimating density directly (e.g., kernel density estimation)
  - Effects of point-wise convergence of intermediate expansions on overall error?
- We would like error analysis/estimates to tell us
  - How many terms to keep in the KL
  - What order to compute expansions in the transformed basis
- Can this be incorporated into other solver strategies?
  - Full Newton or JFNK?
- Can we further reduce cost by not transforming component responses back to original basis?
  - How would convergence be measured?

# Auxiliary Slides

# General Stochastic Expansion Uncertainty Quantification Framework

- Stochastic collocation and non-intrusive polynomial chaos are *essentially* the same when the collocation points are the same as the quadrature points
  - Differences amount to a change of basis for similar, *but not identical*, spaces
- All three methods exploit regularity of solution w.r.t. random parameters to achieve much faster convergence rates than Monte Carlo
  - Cost grows rapidly with number of stochastic dimensions
- All three methods prefer independent random variables
  - Stochastic Galerkin: Polynomials are tensor products of 1-D polynomials of total order N
  - Stochastic Collocation/NIPC: Quadrature/collocation point grid built from tensor products or Smolyak sparse grids derived from Gaussian quadrature points from above 1-D polynomials
- Stochastic Galerkin requires forming and solving a new coupled spatial-stochastic nonlinear problem

$$0 = F(U) = \begin{bmatrix} F_0 \\ F_1 \\ \vdots \\ F_{P_{SG}} \end{bmatrix}, \quad U = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{P_{SG}} \end{bmatrix}, \quad P_{SG} + 1 = \frac{(M+N)!}{M!N!}$$

- Stochastic collocation/NIPC only require solving a sequence of deterministic nonlinear problems

$$f(u_i, x_i) = 0, \quad i = 0, \ldots, P_{SC} = Q$$

- However:

$$Q = P_{SC} > P_{SG}$$

# Computing SG Residuals/Jacobians via Automatic Differentiation (AD)

- Technology for computing analytic derivatives in simulation codes
  - Propagates derivatives at the scalar-operation level
  - Good tools available
- Provides deep interface into application code
- Leverage AD interface for any computation that can be done in an operation by operation manner

$$y = \sin(e^x + x \log x), \quad x = 2$$

| | | $x$ | $\frac{d}{dx}$ |
|---|---|---|---|
| $x \leftarrow 2$ | $\frac{dx}{dx} \leftarrow 1$ | 2.000 | 1.000 |
| $t \leftarrow e^x$ | $\frac{dt}{dx} \leftarrow t\frac{dx}{dx}$ | 7.389 | 7.389 |
| $u \leftarrow \log x$ | $\frac{du}{dx} \leftarrow \frac{1}{x}\frac{dx}{dx}$ | 0.301 | 0.500 |
| $v \leftarrow xu$ | $\frac{dv}{dx} \leftarrow u\frac{dx}{dx} + x\frac{du}{dx}$ | 0.602 | 1.301 |
| $w \leftarrow t + v$ | $\frac{dw}{dx} \leftarrow \frac{dt}{dx} + \frac{dv}{dx}$ | 7.991 | 8.690 |
| $y \leftarrow \sin w$ | $\frac{dy}{dx} \leftarrow \cos(w)\frac{dw}{dx}$ | 0.991 | -1.188 |

# Sacado: AD Tools for C++ Applications

- AD via operator overloading and C++ templating
  - Transform to template code & instantiate on Sacado AD types
  - Easy to add new AD types to a code
- Designed for use in complex C++ codes
  - Sacado::FEApp example demonstrates approach
- Very successful in enabling analytic sensitivity calculations in large-scale simulation codes
  - Charon, Aria, Xyce, Alegra, LAMMPS, Albany

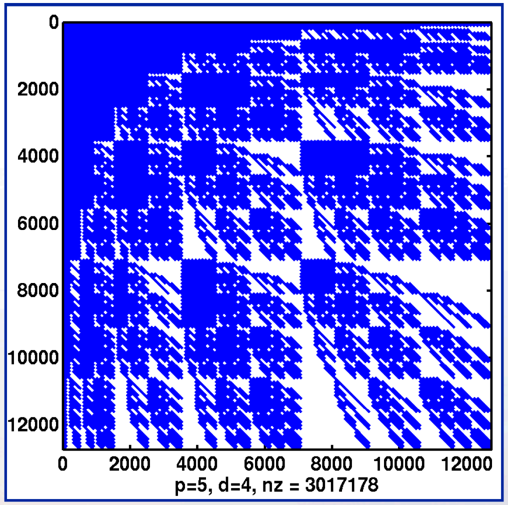
Trilinos

- http://trilinos.sandia.gov
- Algorithms and enabling technologies
- Large-scale scientific and engineering applications
- C++ Object oriented framework

# Stokhos: Trilinos Tools for Intrusive Stochastic Galerkin UQ Methods

- Eric Phipps, Chris Miller, Habib Najm, Bert Debusschere, Omar Knio
- AD overloaded operators for SG propagation
  - Sacado: Trilinos AD tools for C++ applications
- Tools solving SG linear systems
  - Jacobian-free (Ghanem) or fully assembled
  - Mean-based preconditioning
  - Hooks to Trilinos parallel linear solvers
- Nonlinear SG application code interface
  - Interface to nonlinear solver, time integrator, optimizer
  - Global quadrature SG propagation method
- Enabling investigation of SG methods in complex applications

http://trilinos.sandia.gov




Sandia National Laboratories

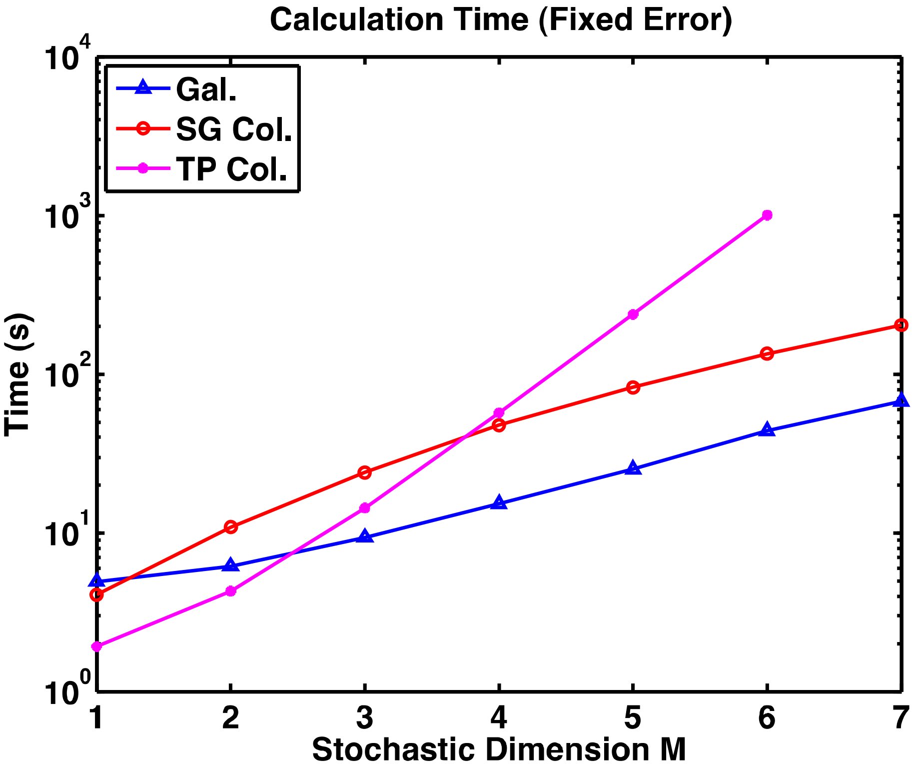
# Comparing Linear and Nonlinear PDEs

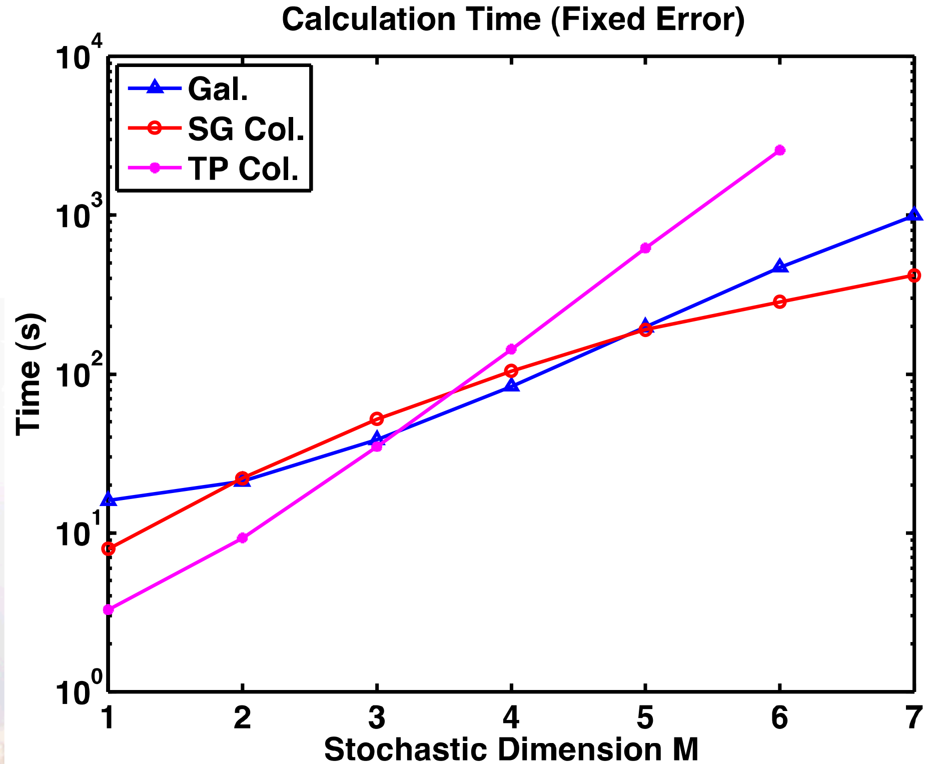
$$-\nabla \cdot (a(x,\xi)\nabla u) = 1, \; x \in [0,1] \times [0,1]$$

$$a(x,\xi) = \mu + \sigma \sum_{k=1}^{M} \sqrt{\lambda_k} f_k(x) \xi_k$$

$$-\nabla \cdot (a(x,\xi)\nabla u) = \alpha u^2, \; x \in [0,1] \times [0,1]$$

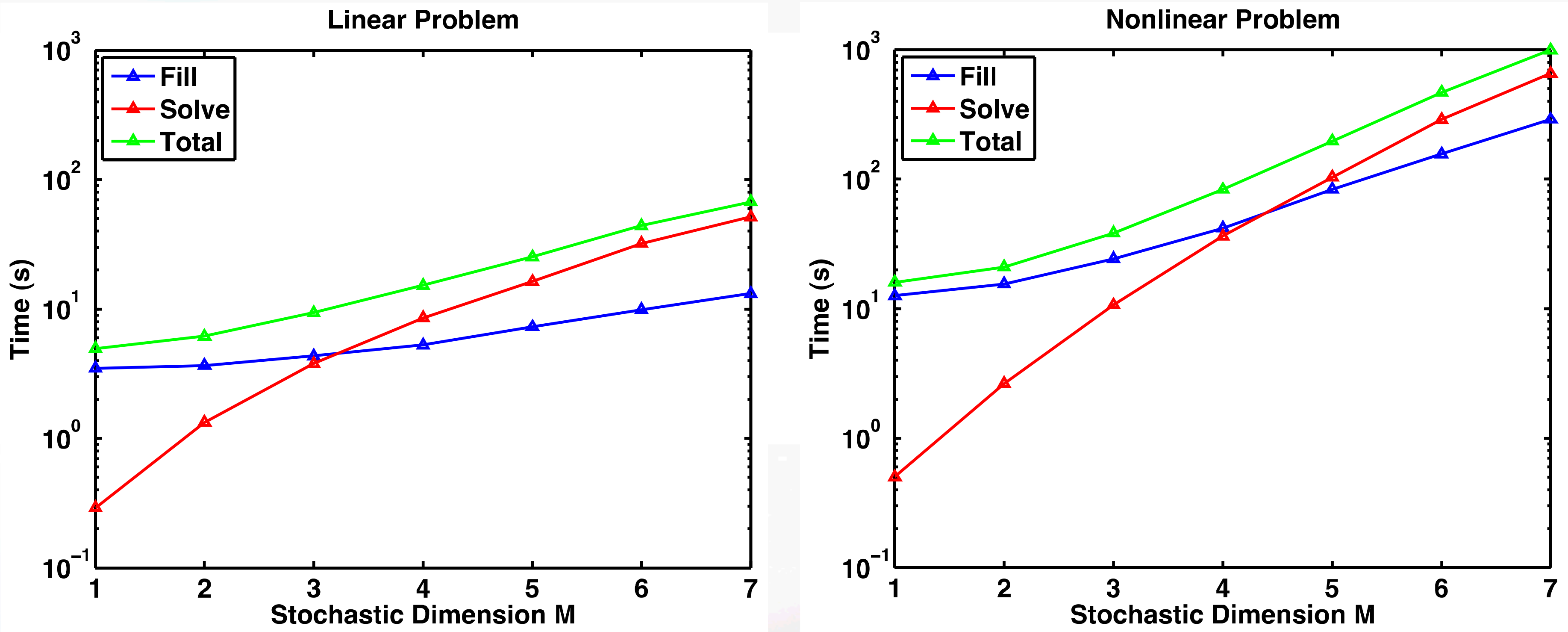
$$a(x,\xi) = \mu + \sigma \sum_{k=1}^{M} \sqrt{\lambda_k} f_k(x) \xi_k$$




**DAKOTA tensor product (Gauss-Legendre) and sparse grid stochastic collocation (Gauss-Patterson, Burkardt/Eldred)**

# Analysis of Intrusive SG Computational Cost




- Increased cost due to two sources
  - Filling nonlinear SG residual and Jacobian
  - Linear solve for each Newton iteration
    Matrix-vector product scales as $O(P^2)$ versus $O(MP)$

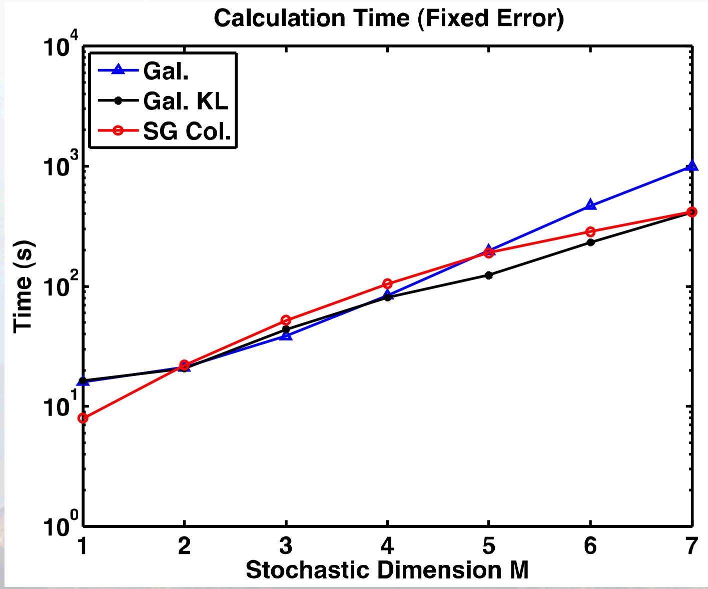
# KL Expansion of SG Jacobian Operator

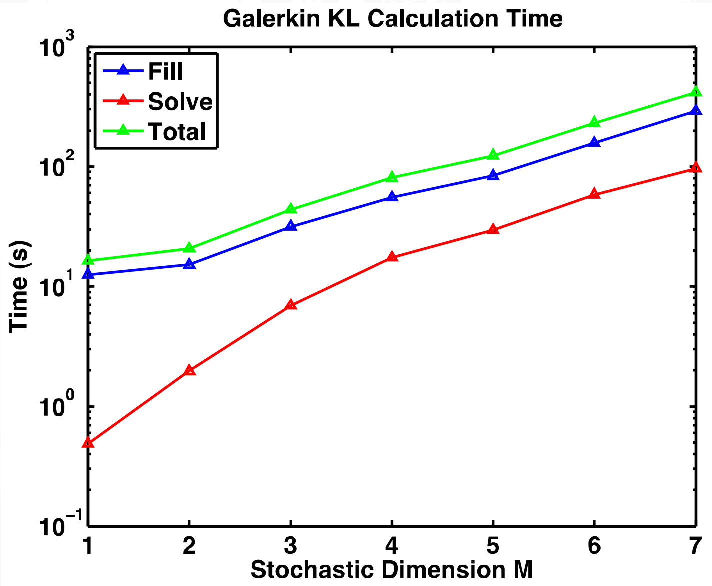
- SG Jacobian operator can be approximated by a truncated KL expansion:

$$\frac{\partial f}{\partial u}(\hat{u}(\xi), \xi) \approx \sum_{k=0}^{P} J_k \psi_k(\xi) \approx J_0 + \sum_{j=1}^{\bar{M}} \sqrt{\lambda_j} B_j \eta_j$$

$$\eta_j = \frac{1}{\sqrt{\lambda_j}} \sum_{k=1}^{P} \mathrm{vec}(B_j)^T \mathrm{vec}(J_k) \psi_k(\xi), \quad (ZZ^T)\mathrm{vec}(B_j) = \lambda_j \mathrm{vec}(B_j), \quad Z = [\mathrm{vec}(J_1) \ldots \mathrm{vec}(J_P)]$$

- Reduces matrix-vector product cost to $\sim O(\bar{M}P)$



# Nonlinear Elimination for Network Coupled Systems

**Component 1**

$$v_2 = G_1(v_1, p_1) = g_1(u_1(v_1), p_1) \quad \text{s.t.} \quad f_1(u_1, v_1, p_1) = 0$$

$v_1$     $v_2$

**Component 2**

$$v_1 = G_2(v_2, p_2) = g_2(u_2(v_2), p_2) \quad \text{s.t.} \quad f_2(u_2, v_2, p_2) = 0$$

## Nonlinear elimination

### Equations

$$v_2 - G_1(v_1, p_1) = 0$$
$$v_1 - G_2(v_2, p_2) = 0$$

### Newton Step

$$\begin{bmatrix} -dG_1/dv_1 & 1 \\ 1 & -dG_2/dv_2 \end{bmatrix} \begin{bmatrix} \Delta v_1 \\ \Delta v_2 \end{bmatrix} = - \begin{bmatrix} v_2 - G_1(v_1, p_1) \\ v_1 - G_2(v_2, p_2) \end{bmatrix}$$

$$\frac{dG_i}{dv_i} = -\frac{\partial g_i}{\partial u_i} \left( \frac{\partial f_i}{\partial u_i} \right)^{-1} \frac{\partial f_i}{\partial v_i}$$

# (Semi-) Intrusive UQ for Network/Nonlinear Elimination Coupled Systems

Define: $\xi = (\xi_1, \xi_2), \quad \hat{u}_i(\xi) = \sum_{j=0}^{P} u_{ij} \Psi_j(\xi), \quad \hat{v}_i(\xi) = \sum_{j=0}^{P} v_{ij} \Psi_j(\xi)$

Where coefficients for $\hat{u}_i(\xi)$ are computed by any UQ method, e.g.,

Intrusive: $\frac{1}{\langle \Psi_j^2 \rangle} \langle (f_i(\hat{u}_i(\xi), \hat{v}_i(\xi), \xi_i) \Psi_j(\xi) \rangle = 0$

Non-intrusive: $u_{ij} = \frac{1}{\langle \Psi_i^2 \rangle} \sum_{k=0}^{Q} w_k u_i^k \Psi_j(x_k), \quad f_i(u_i^k, \hat{v}_i(x_k), x_k) = 0$

Let $\hat{G}_i(\xi) = \sum_{j=0}^{P} G_{ij} \Psi_j(\xi), \quad G_{ij} = \frac{1}{\langle \Psi_j^2 \rangle} \langle g_i(\hat{u}_i(\xi), \xi_i) \Psi_j \rangle$

Then the intrusive SG network system is

$$\left.\begin{array}{r} \frac{1}{\langle \Psi_j^2 \rangle} \langle (\hat{v}_2(\xi) - \hat{G}_1(\xi)) \Psi_j(\xi) \rangle = 0 \\ \frac{1}{\langle \Psi_j^2 \rangle} \langle (\hat{v}_1(\xi) - \hat{G}_2(\xi)) \Psi_j(\xi) \rangle = 0 \end{array}\right\} \Longrightarrow \left.\begin{array}{r} v_{2j} - G_{1j} = 0 \\ v_{1j} - G_{2j} = 0 \end{array}\right\}, \quad j = 0, \ldots, P$$

Which can be solved via a nonlinear elimination.

# Stieltjes Procedure (see Gautschi)

- Assume $\eta(\xi) = \hat{v}(\xi) = \sum_{k=0}^{P} v_i \Psi_i(\xi)$ given and $|\eta| = 1$
- Let $\{\phi_i : i = 0, \ldots, \tilde{P}\}$ be (1-D) polynomials orthogonal w.r.t. measure of $\eta$:

$$\langle \phi_i \phi_j \rangle_\eta = \int_{\mathbb{R}} \phi_i(y) \phi_j(y) \rho_\eta(y) dy = \langle \phi_i^2 \rangle_\eta \delta_{ij}, \quad i, j = 0, \ldots, \tilde{P}$$

- Polynomials defined a 3-term recurrence:

$$\phi_{i+1}(y) = (y - \alpha_i)\phi_i(y) - \beta_i \phi_{i-1}(y), \quad i = 0, 1, 2, \ldots$$
$$\phi_{-1}(y) = 0, \quad \phi_0(y) = 1$$

where

$$\alpha_i = \frac{\int y \phi_i^2(y) \rho_\eta(y) dy}{\int \phi_i^2(y) \rho_\eta(y) dy}, \quad i = 0, 1, 2, \ldots,$$
$$\beta_i = \frac{\int \phi_i^2(y) \rho_\eta(y) dy}{\int \phi_{i-1}^2(y) \rho_\eta(y) dy}, \quad i = 1, 2, \ldots,$$
$$\beta_0 = 1$$

# An Appproach for Approximating Integrals w.r.t. Unknown Measure

- By measure transformation theorem:

$$\int_{\mathbb{R}} \phi_i^2(y)\rho_\eta(y)dy = \int_\Gamma \phi_i^2(\eta(x))\rho_\xi(x)dx$$

- Approximate new basis in terms of old:

$$\phi_i(\eta(\xi)) \approx \sum_{j=0}^{P} \phi_{ij}\Psi_j(\xi), \quad \phi_{ij} = \frac{1}{\langle \Psi_j^2 \rangle_\xi} \int_\Gamma \phi_i(\eta(x))\Psi_j(x)\rho_\xi(x)dx$$

$$\phi_{ij} \approx \frac{1}{\langle \Psi_i^2 \rangle_\xi} \sum_{k=0}^{Q} w_k \phi_i(\eta(x_k))\Psi_j(x_k)$$

- Then

$$\int_{\mathbb{R}} \phi_i^2(y)\rho_\eta(y)dy \approx \int_\Gamma \Big(\sum_{j=0}^{P} \phi_{ij}\Psi_j\Big)^2 \rho_\xi(x)dx = \sum_{j=0}^{P} \phi_{ij}^2 \langle \Psi_j^2 \rangle_\xi$$

- Similarly

$$\int_{\mathbb{R}} y\phi_i^2(y)\rho_\eta(y)dy \approx \sum_{j,k,l=0}^{P} \phi_{ij}\phi_{ik}v_l \langle \Psi_j \Psi_k \Psi_l \rangle_\xi$$

# Multi-Variate Basis and Dependence

- Multi-variate tensor product polynomials:

$$\tilde{\Phi}_i(\eta, \xi_1) = \phi^1_{i_1}(\eta_1) \ldots \phi^L_{i_L}(\eta_L) \psi^{11}_{j_1}(\xi_{11}) \ldots \psi^{1M_1}_{j_{M_1}}(\xi_{1M_1}), \quad |\eta| = L, \quad |\xi_1| = M_1$$

- In general, these polynomials not orthogonal w.r.t. joint PDF of $(\eta, \xi_1)$

$$\rho_{(\eta,\xi_1)}(y, x_1) \neq \rho_{\eta_1}(y_1) \ldots \rho_{\eta_L}(y_L) \rho_{\xi_{11}}(x_{11}) \ldots \rho_{\xi_{1M_1}}(x_{1M_1}) \equiv \tilde{\rho}(y, x_1)$$

- First approach: Orthogonalize this basis using Gram-Schmidt

$$\Phi_i = \tilde{\Phi}_i - \sum_{j=0}^{i-1} \frac{\langle \tilde{\Phi}_i \Phi_j \rangle_\xi}{\langle \Phi_j^2 \rangle_\xi} \Phi_j, \quad \langle \tilde{\Phi}_i \Phi_j \rangle_\xi = \int_\Gamma \tilde{\Phi}_i(\eta(x), \pi_1(x)) \Phi_j(\eta(x), \pi_1(x)) \rho_\xi(x) dx$$

  - Don't know how to define a good set of quadrature points for this basis (so no non-intrusive approach)
  - Intrusive Galerkin algorithm is much more expensive, e.g., $C_{ijk} = \langle \Phi_i \Phi_j \Phi_k \rangle$ is dense.