Stokhos: Trilinos Tools for Embedded Stochastic-Galerkin Uncertainty Quantification Methods

Eric Phipps
etphipp@sandia.gov
Ray Tuminaro, Chris Miller (U. of Maryland)

Trilinos User Group 2009

SAND 2009-7541C
Models of Uncertainty

• Predictive simulation means making a rigorous statement about the world based on computational simulation
  – Must understand uncertainty in simulation input data and its effects on simulations

• Two broad classes of uncertainty
  – Aleatory or irreducible uncertainty: “inherent randomness”
  – Epistemic or reducible uncertainty: “lack of knowledge”

• Two basic sources of uncertainty
  – Finite-dimensional simulation parameters
  – Infinite-dimensional correlated random fields/stochastic processes

• Assume a finite dimensional representation given by random variables/vectors with known probability distributions
  – Generating such models is a research area in its own right
Stochastic Galerkin Uncertainty Quantification Methods  
(aka Polynomial Chaos, Spectral Galerkin, Stochastic Finite Elements)

• Stochastic, steady-state problem (possibly after spatial discretization):

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

• Let \( Z \) be a finite-dimensional subspace of \( L^2_{\rho}(\Gamma) \) with basis \( \{\psi_i : i = 0, \ldots, N_{PC}\} \) orthogonal with respect to inner product

\[ \langle fg \rangle \equiv \int_{\Gamma} f(y)g(y)\rho(y)dy \]

• Stochastic Galerkin method (Ghanem, ...):

\[ \hat{u}(\xi) = \sum_{i=0}^{P} u_i\psi_i(\xi) \rightarrow F_i(u_0, \ldots, u_P) = \int_{\Gamma} f(\hat{u}(y), y)\psi_i(y)\rho(y)dy = 0, \ i = 0, \ldots, P \]

• Typically basis polynomials are tensor products of 1-D orthogonal polynomials of total degree \( N \)
  – Assumes independence of random parameters
  – Named polynomials for common densities (Hermite, Legendre, ...)
  – Polynomials can be generated numerically for any density (Gautschi)

• Exponential convergence in \( N \) when solution is sufficiently smooth
Stochastic Galerkin Nonlinear System

- Method generates new coupled spatial-stochastic nonlinear problem

\[ 0 = F(U) = \begin{bmatrix} F_0 \\ F_1 \\ \vdots \\ F_P \end{bmatrix}, \quad U = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_P \end{bmatrix} \]

- Dimension grows rapidly with degree or dimension

\[ P = \frac{(M + N)!}{M!N!} \]

- Block system Jacobian:

<table>
<thead>
<tr>
<th>Stochastic dimension</th>
<th>Polynomial degree</th>
<th>Number of terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>56</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>286</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>3003</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>1,771</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>~53,000</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
<td>~177,000</td>
</tr>
<tr>
<td>100</td>
<td>5</td>
<td>~96,000,000</td>
</tr>
</tbody>
</table>
Trilinos Package Stokhos

• Tools for generating SG residual and Jacobian entries
  – Automatic differentiation with Sacado
    \[ F_i = \int_{\Gamma} f(\hat{u}(y), y) \psi_i(y) \rho(y) dy, \quad \langle \cdot \rangle = \int_{\Gamma} \cdot \rho(y) dy \]
    \[ \frac{\partial F_i}{\partial u_j} \approx \sum_{k=0}^{P} J_k \langle \psi_i \psi_j \psi_k \rangle, \quad J_k = \frac{1}{\langle \psi_k^2 \rangle} \int_{\Gamma} \frac{\partial f}{\partial u}(\hat{u}(y), y) \psi_k(y) \rho(y) dy \]

• Nonlinear SG application code interface through EpetraExt::ModelEvaluator
  – New In/Out Args for SG expansions (x_sg, xdot_sg, f_sg, W_sg, …)
  – ModelEvaluator that implements calculation of SG in/out args via quadrature
  – ModelEvaluator adaptor translating SG expansion to block nonlinear problems using EpetraExt::BlockVector, etc… (f_sg --> block f)

• Epetra operators for solving block SG linear systems
  – Jacobian-free operator with mean-based preconditioning, and fully assembled Jacobian
    \[ \frac{\partial F_i}{\partial u_j} \approx \sum_{k=0}^{N_{PC}} J_k \langle \psi_i \psi_j \psi_k \rangle \quad \Rightarrow \quad \left( \frac{\partial \bar{F}}{\partial \bar{u}} \right)_i = \sum_{j,k=0}^{N_{PC}} J_{kj} \langle \psi_i \psi_j \psi_k \rangle \]
Generating SG Residual/Jacobian Entries Through Automatic Differentiation (AD)

• Trilinos package Sacado provides AD capabilities to C++ codes
  – AD relies on known derivative formulas for all intrinsic operations plus chain rule
  – AD data types & overloaded operators
  – Replace scalar type in application with Sacado AD data types

• Similar approach can be used to apply SG projections in an operation by operation manner

  Given \( a(y) = \sum_{i=0}^{P} a_i \psi_i(y) \), \( b = \sum_{i=0}^{P} b_i \psi_i(y) \), find \( c(y) = \sum_{i=0}^{P} c_i \psi_i(y) \)

  such that \( \int_{\Gamma} \left( c(y) - \phi(a(y), b(y)) \right) \psi_i(y) \rho(y) dy = 0, \quad i = 0, \ldots, P \)

  – Simple formulas for addition, subtraction, multiplication, division
  – Transcendental operations are more difficult
    • Taylor series and time integration (Fortran UQ Toolkit by Najm, Debusschere, Ghanem, Knio)
    • Tensor product and sparse-grid quadrature (Dakota)
AD Approach Generally Works Well

$$u = \log \left( \frac{1}{1 + (e^x)^2} \right)$$

- AD approach is usually accurate
- Truncation error can cause catastrophic failure
2-D Linear Diffusion Problem

(Chris Miller – 2009 CSRI Summer Student & Ray Tuminaro)

\[-\nabla \cdot \left( a(x, \xi) \nabla u \right) = 1\]

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

- 2-D finite difference discretization
- ML multi-level mean preconditioner
- Exponential covariance kernel
- Truncated Gaussian random variables (Rys polynomials)
2-D Nonlinear Diffusion Problem

\[ \nabla^2 u + \frac{\alpha_1 + \cdots + \alpha_M}{M} u^2 = 0 \]

- New Albany code (Salinger et al)
- 2-D finite element discretization
- RILU mean preconditioner
- Uniform random variables (Legendre polynomials)
Summary of Stokhos Capabilities

• Stokhos provides a complete set of tools for stochastic Galerkin UQ problems
  – Sacado overloaded operators for generating SG residual/Jacobian entries via AD
  – Nonlinear application code interface for SG nonlinear problems (EpetraExt)
  – Operators and preconditioners for SG linear systems (Epetra)

• Optional third-party libraries
  – Fortran UQ toolkit for Taylor/time integration approaches for SG expansion
  – Dakota for sparse-grid quadrature (brought in through TriKota)

• Demonstration provided by Sacado’s FEApp

• Stokhos is fully functional, but still primarily a research tool
  – Dakota provides robust UQ capabilities
What We’re Working on

• Software
  – Building Stokhos capabilities from PECOS
  – Thyra interfaces

• Algorithmic
  – Linear solver/preconditioner methods for block SG linear systems
  – Reducing high linear solve cost for nonlinear problems
  – Stochastic Galerkin methods for multi-physics systems

• Continuing investigation in target applications
  – Charon, Albany

• Finishing the release
  – Internal for Trilinos 10
  – External once copyright is completed (3 months and counting!)