A Library for Large Scale Parallel Modeling and Simulation of Spatial Rand(SAND2012-2360C)

Brian Carnes
bcarnes@sandia.gov
John Red-Horse
jrredho@sandia.gov

Sandia National Laboratories
Albuquerque, NM

April 2-6, 2012
SIAM UQ 2012
SAND2012-****

Sandia National Laboratories

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

National Nuclear Security Administration

Contents

- Motivation for the RF library
- KL eigensolver: a preprocessor for RFs
- A posteriori error estimation
- Applications

Random Variables or Random Fields?

- Often propagation of uncertainty can be done using scalar parameters, for bulk material properties.
- However, uncertainty can be distributed within a material, along a boundary, etc.

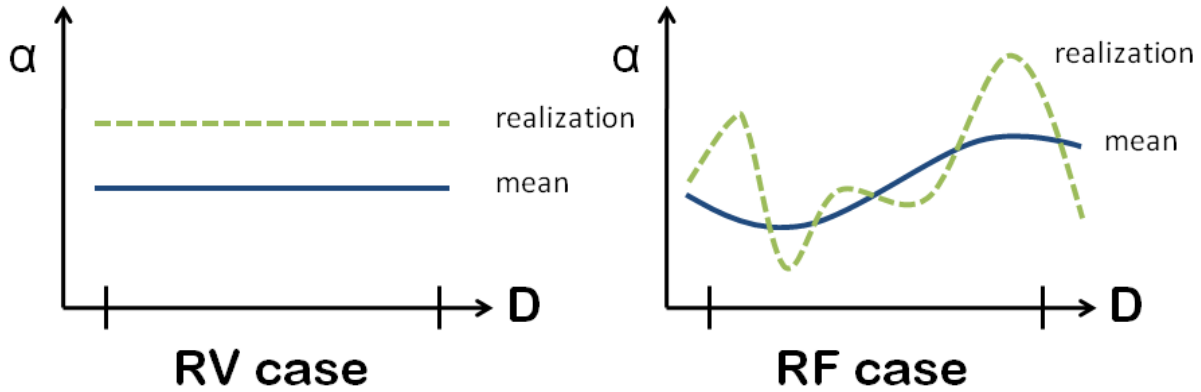


Figure: Random fields enable spatial/temporal stochastic variables.

Motivation

- Karhunen-Loeve (KL) series are general-purpose random field models
- Exact solutions are known only for simple geometries (bricks) and covariance kernels
- Goal of our library: provide a numerical KL solver that can generate KL series from
  - general domains D that have been meshed
  - general covariance kernel $r(x, y)$ for any $x, y \in D \times D$
- Assemble realizations of RFs (KL, PCE) for uncertainty propagation
- Support large scale parallelism (MPI)

Non-intrusive UQ with Random Fields

- RF Simulator added to traditional non-intrusive UQ coupling
- RV coeffs $\{\xi_j\}$ are now DoFs for random field $\alpha(x, \xi)$, such as through a Karhunen-Loeve (KL) series

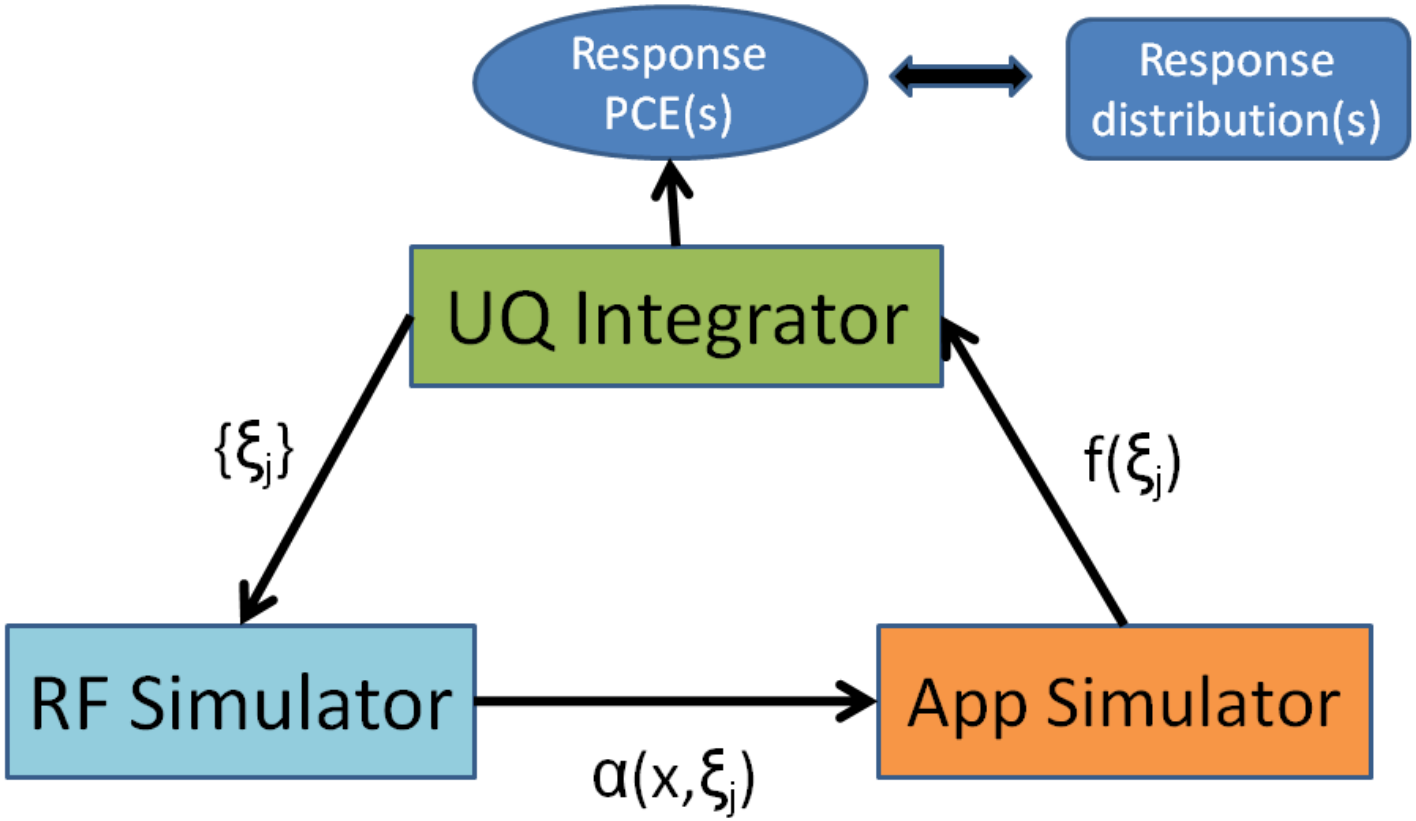


Figure: Overview of non-intrusive UQ coupled multiphysics.

Overview: KL Series Representations

- The Karhunen-Loeve expansion of a random field takes the form

$$\alpha(x,\omega) = m(x) + \sum_{j=1}^{\infty} \xi_j(\omega)\,\sqrt{\lambda_j}\,\phi_j(x) \tag{1}$$

- It requires the solution of the KL eigenproblem: find $\{\lambda_j, \phi_j\}$

$$K\,\phi_j = \lambda_j\,\phi_j, \tag{2}$$

$$(K\,v)(x) \equiv \int_D r(x,y)\,w(y)\,dy, \quad x \in D \tag{3}$$

- The covariance function r is defined by

$$r(x,y) \equiv E[(\alpha(x) - m)\,(\alpha(y) - m)]$$

- The uncorrelated RV coefficients ξ_j are defined by

$$\xi_j \equiv \frac{1}{\sqrt{\lambda_j}} \int_D (\alpha(x) - m)\,\phi_j(x)\,dx$$

Discrete KL Solver

- We discretize the spatial domain into N_e elements.
- A basis for $V_h \subset L^2(D)$ is made from discontinuous polynomials.
- FE approximation in V_h leads to generalized eigenproblem:

$$A\,\Phi = \lambda\, B\Phi$$

- Dense matrix A, sparse matrix B (both SPD)

$$A_{ij} \equiv \int_D \int_D r(x,y)\, v_i(y)\, v_j(x)\, dy\, dx, \qquad B_{ij} \equiv \int_D v_i(x)\, v_j(x)\, dx$$

- Assembly of B matrix is standard (mass matrix)
- Assembly of dense A matrix requires double element loop

Eigensolver and Preconditioner

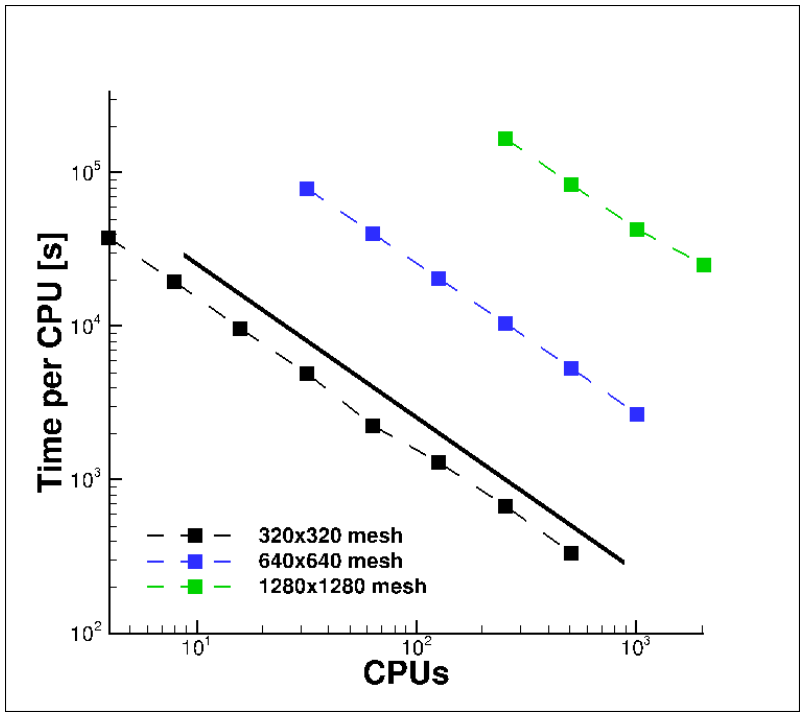
- The eigensolver is an iterative block Davidson solver [1]
- Implemented in the Trilinos/Anasazi library [2]
- Requires only matvec operation
- Computes only a subset of eigenvalues (largest)
- Can leverage preconditioner of dense matrix A
- Matrix free option to prevent memory overflow
  - Avoid $O(N^2)$ storage (recall: sparse storage is $O(N)$).

[1] Arbenz et. al., "A Comparison of Eigensolvers for Large-scale 3D Modal Analysis Using AMG-Preconditioned Iterative Methods", IJNME, 64 (2005)

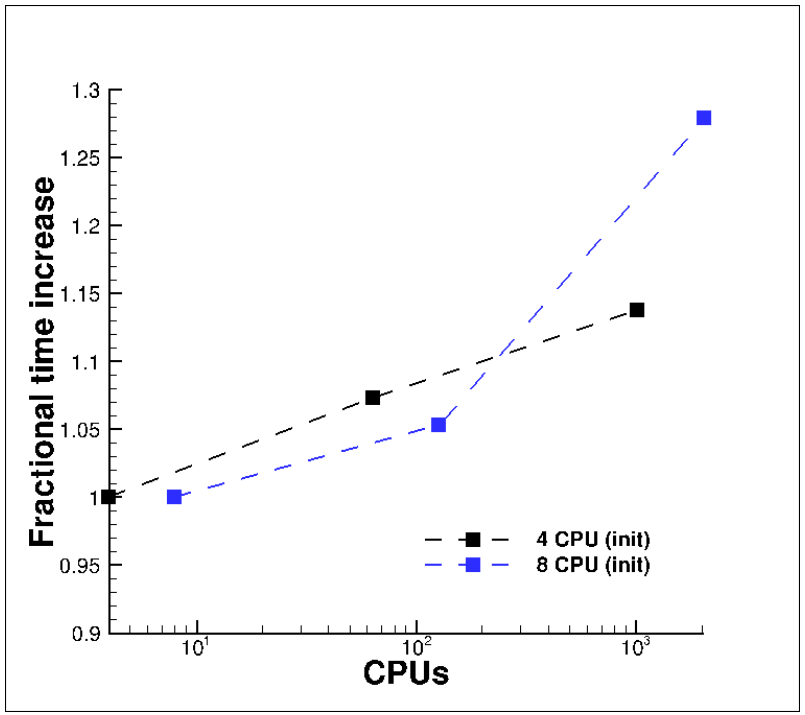
[2] trilinos.sandia.gov

KL Eigensolver: Parallel Scaling Study

Figure: Strong/weak scaling of KL eigensolver (2048 procs, 1.6M elems) on 2D domains using matrix-free implementation and iterative eigensolver



Strong scaling



Weak scaling

KL Eigensolver: Verification Test (1/2)

- We used a 1D problem on $D = (-1, 1)$ with $r(x, y) \equiv \exp(-|x - y|)$.
- The exact KL eigensolution can be computed.
- The RV coefficients ξ_j are assumed to be independent standard Gaussians.

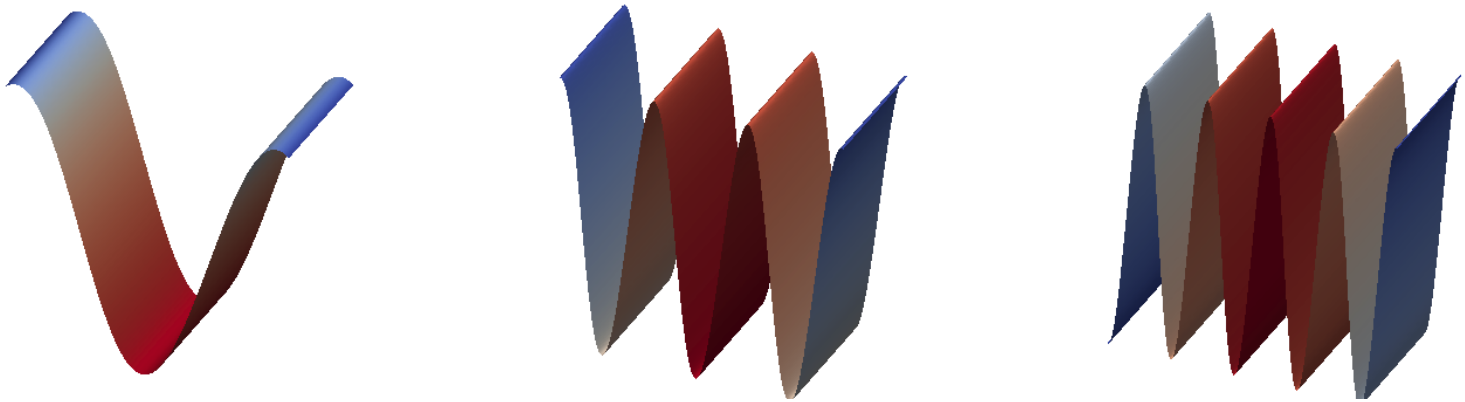
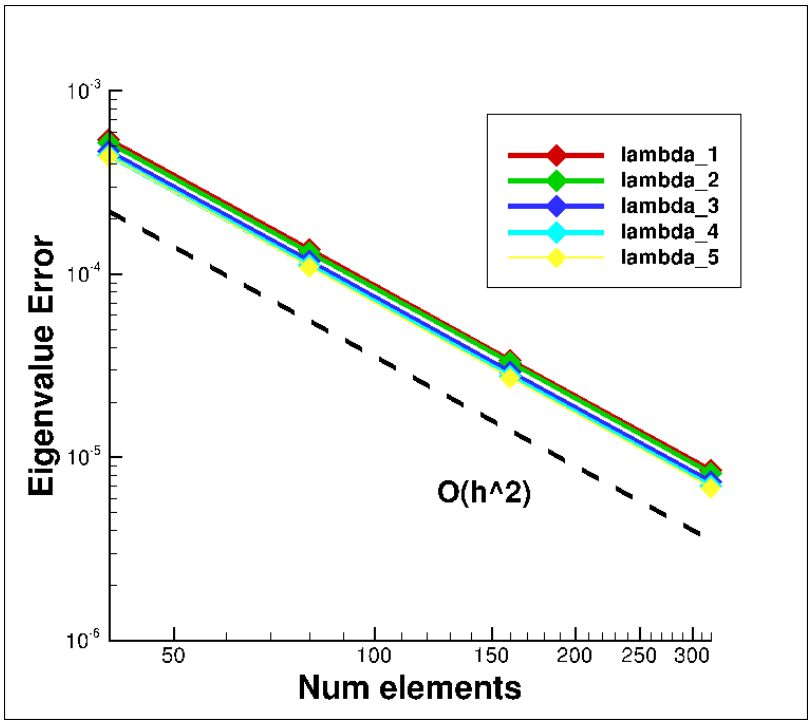
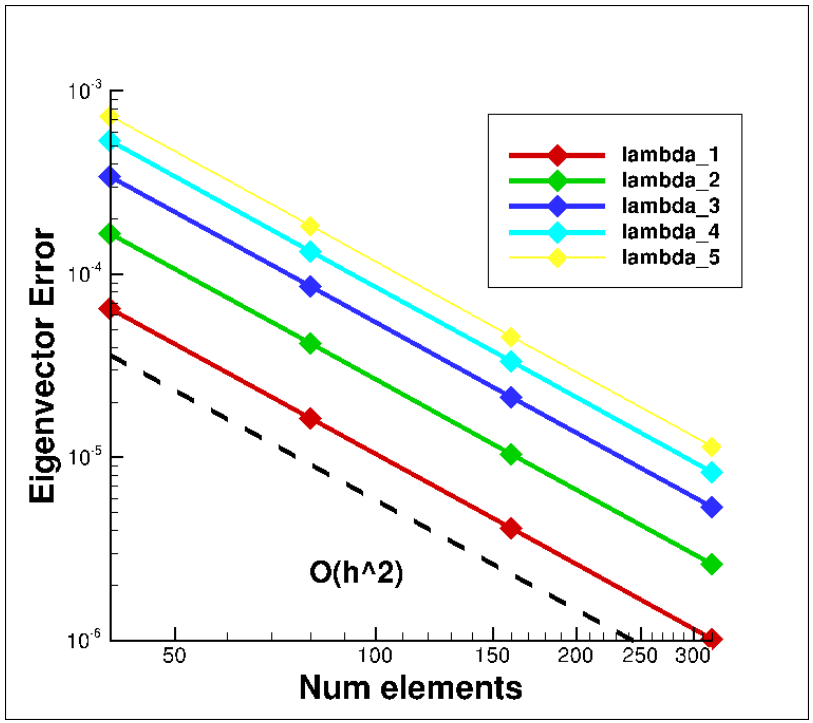


Figure: Eigenfunctions 3, 7, and 10.

KL Eigensolver: Verification Test (2/2)

- Eigenfunctions converge $O(h^2)$ with midpoint integration.
- Midpoint values can be average to continuous nodal functions.





RF Realizations: KL and PCE

- RF realizations can be output to disk using ExodusII files
- This uses existing mesh partitions if the same as KL solver
- We provide for a workset level assembly of RFs over a collection of mesh elements
- This enables integration with the application code for performance
- We also can assemble Polynomial Chaos expansions using orthogonal polynomials from the DAKOTA/Pecos library [3].

[3] http://dakota.sandia.gov/

A Posteriori Error Estimation for KL Eigenproblem

- The residual (for a given eigenpair) is

$$R \equiv R(\phi_h, \lambda_h) \equiv -K\,\phi_h + \lambda_h\,\phi_h$$

- Defining errors $e \equiv \phi - \phi_h$ and $\mu \equiv \lambda - \lambda_h$, we have derived estimates for individual eigenvalue/eigenfunction pairs:

$$\|e\| \approx \frac{\|R\|}{\lambda}, \quad \mu \approx \frac{\|R\|^2}{\lambda}$$

- These estimates capture both the dependence on mesh size h and on the eigenvalues.

A Posteriori Error Estimation for KL Series

- The numerical KL solver creates approximation error.
- The truncated approximate KL series is

$$A(x,\omega) = M(x) + \sum_{j=1}^{N} \xi_j(\omega)\,\sqrt{\Lambda_j}\,\Phi_j(x)$$

- For non-intrusive UQ, we are given a realization of the RVs $\{\xi_j\}$.
- The error estimation problem is then deterministic and can be solved as a preprocessor step.

A Posteriori Error Estimation for KL Series (2)

- The error $e_\alpha = \alpha - A$ can be linearized as:

$$\|e_\alpha\| \approx E \equiv \left\| \sum_{i=1}^{N} \xi_i \left(\frac{1}{2\sqrt{\Lambda_i}} \mu_i \Phi_i + \sqrt{\Lambda_i}\, e_i \right) \right\|$$

- The simplest estimate is to use the triangle inequality:

$$E \leq \sum_{i=1}^{N} |\xi_i| \left(\frac{1}{2\sqrt{\Lambda_i}} |\mu_i| + \sqrt{\Lambda_i}\, \|e_i\| \right) \quad \text{(drop } \mu_i\text{)}$$

$$\approx \sum_{i=1}^{N} |\xi_i| \sqrt{\Lambda_i}\, \|e_i\| \approx \sum_{i=1}^{N} |\xi_i| \frac{\|R_i\|}{\sqrt{\Lambda_i}} \quad \text{(apply } e_i \text{ estimate)}$$

An Optimal Error Estimate

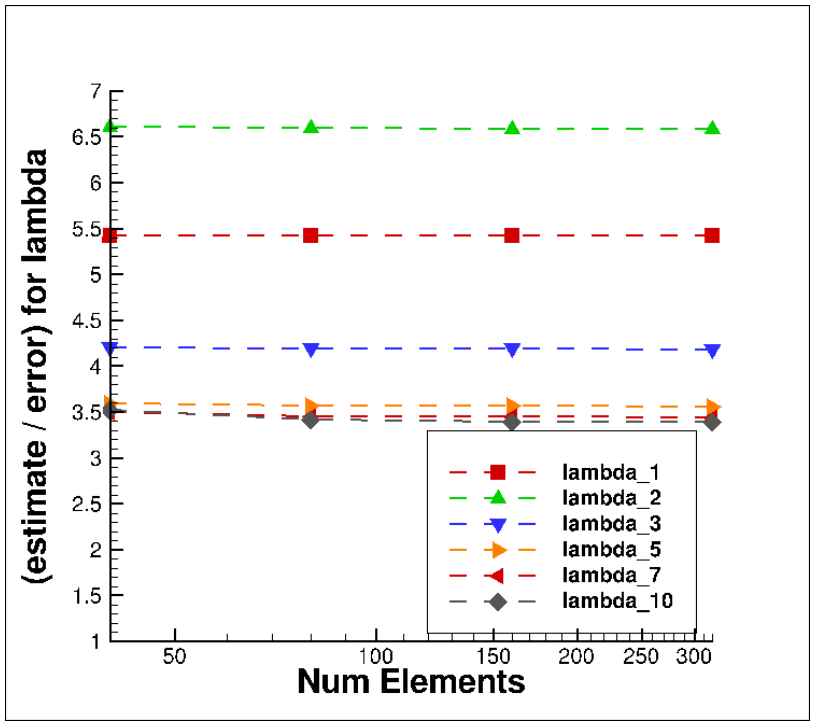
- The previous error estimate significantly overestimates the error for large N.
- We can improve on this by approximating all (e_i, e_j) terms and allowing for cancellation:

$$\|E\|^2 \approx \sum_{i,j=1}^{N} \xi_i \, \xi_j \, \frac{(R_i, R_j)}{\sqrt{\Lambda_i}\sqrt{\Lambda_j}} = \left\| \sum_{i=1}^{N} \xi_i \, \frac{R_i}{\sqrt{\Lambda_i}} \right\|^2$$

- The estimate requires storage of the R_i or (R_i, R_j) since cannot be evaluated without the ξ_i's.

Effectiveness of KL Error Indicator

- We test the error indicator using the 1D verification test.
- Error in ϕ is asymptotically exact; error in λ overestimates by factor of 3 to 6.



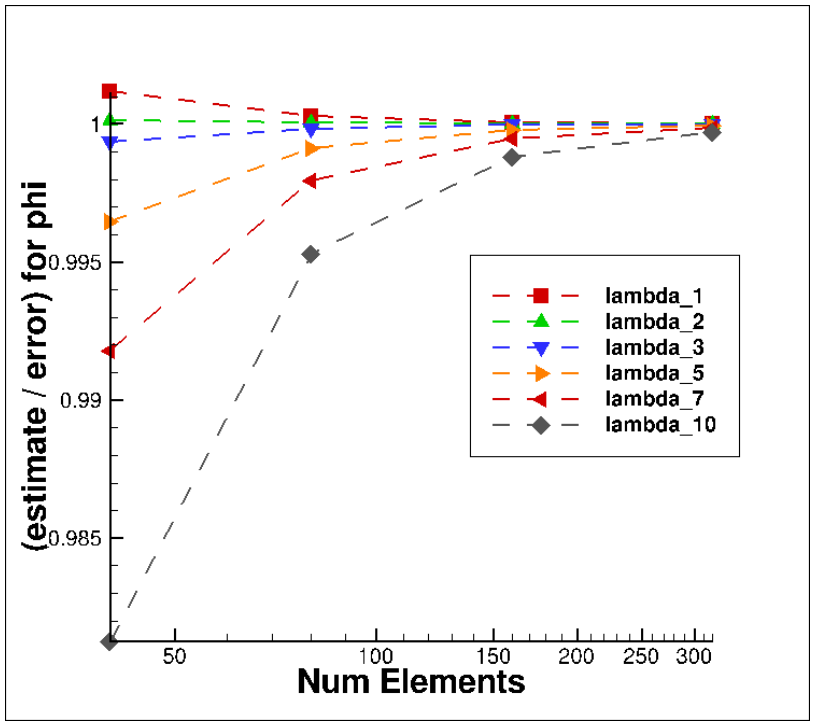


Figure: Ratio of error indicator to true error for eigenvalues/eigenvectors.

Effectiveness of KL Error Indicator

- We test the KL error indicator also using the 1D verification test.
- Here we take ten terms with $\xi_i = 1$ for all terms.

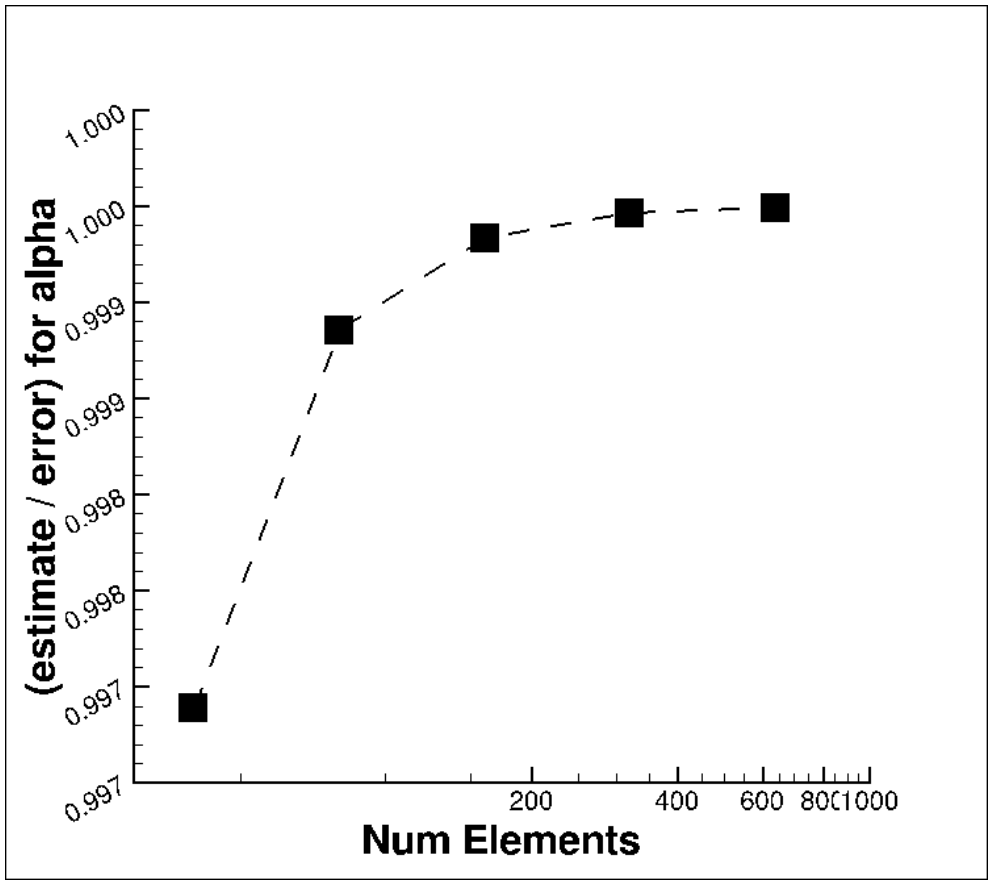
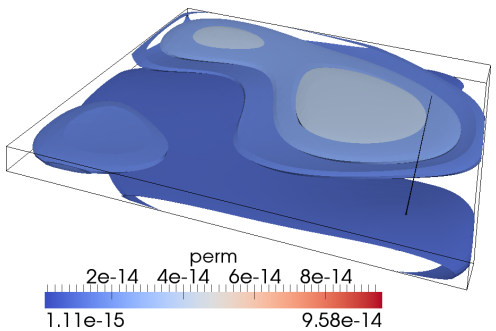
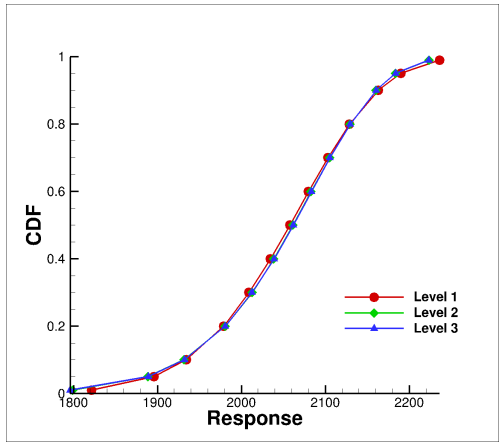


Figure: Ratio of error indicator to true error for 1D random field realization.

Application: Flow in Heterogeneous Porous Media

- The permeability RF had anisotropic correlation length scales.
- A CDF curve for average well pressure was computed using DAKOTA's sparse grid integrator.





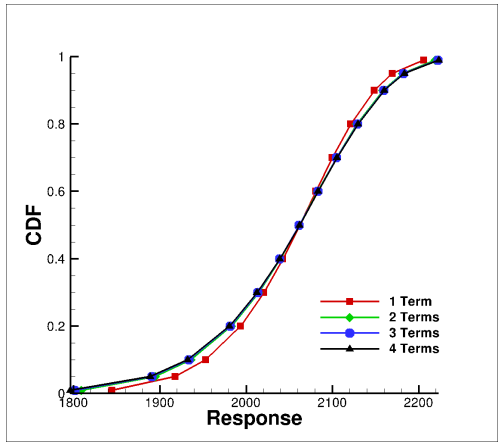


Figure: Sample realization of permeability, convergence of CDF under sparse grid level and number of KL terms. Simulations done using SIERRA/Aria.

Application: Heterogeneous Foam Structural Analysis

- A random field models the Young's Modulus of a foam component.
- The expansion uses 12 terms with $0.138 \leq \lambda \leq 26.59$.
- A CDF curve for maximum acceleration was computed using DAKOTA's sparse grid integrator.

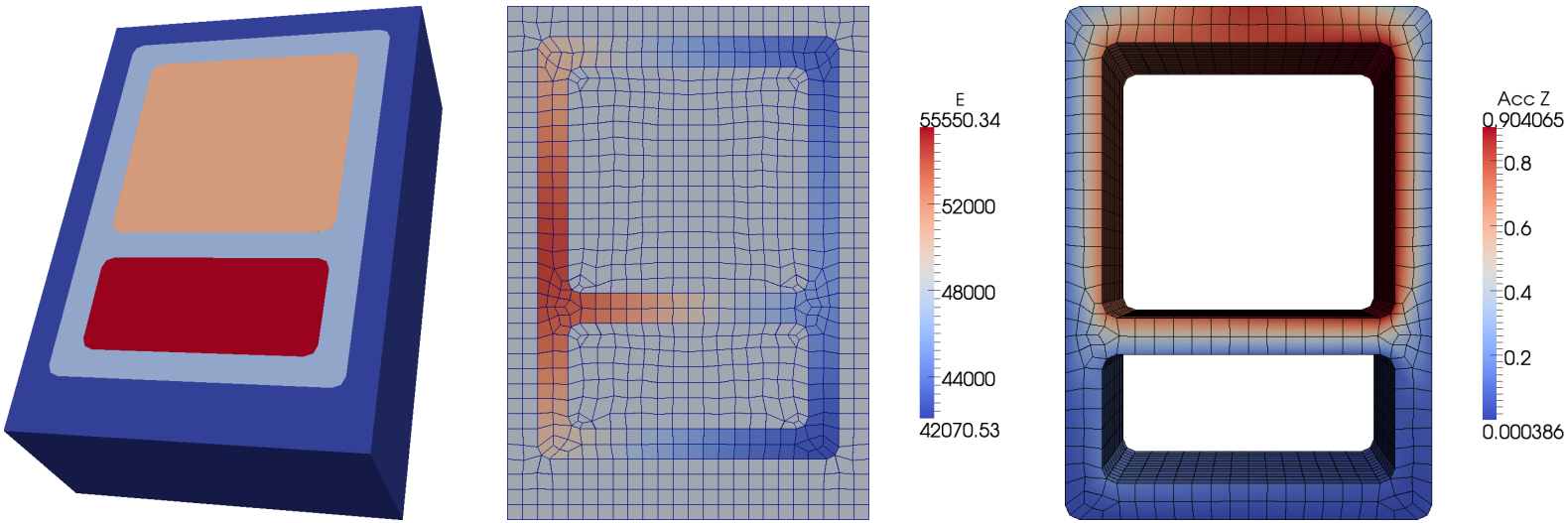


Figure: Problem domain (foam in light blue), RF realization, and computed acceleration. Calculations were done in the Salinas code.

Summary

- We have presented a parallel library for RF discretization using KL series.
- A posteriori error estimation techniques for the KL preprocessor and realizations have been demonstrated.
- Future work:
  - Interface to experimental data for end-to-end RF discretization
  - Release of Open Source code (Trilinos or DAKOTA/Pecos)