SAND2013-7953C

QUEST UQ Methods & Tools

— Connections to SUPER Research Areas —

Habib N. Najm

Sandia National Laboratories,
Livermore, CA

SUPER SciDAC Institute Workshop
Sep 19, 2013
Oakland, CA

Acknowledgement

Contributions by:

- Mike Eldred, Bert Debusschere
  – Sandia National Laboratories, NM & CA
- Omar Ghattas, Chris Simmons
  – University of Texas, Austin

This work was supported by:

- US Department of Energy (DOE), Office of Advanced Scientific Computing Research (ASCR), Scientific Discovery through Advanced Computing (SciDAC)

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

Outline

Key Elements of the QUEST UQ strategy

- Probabilistic framework
  - Uncertainty is represented using probability theory
- Parameter Estimation, Model Calibration
  - Experimental measurements
  - Regression, Bayesian Inference
- Forward propagation of uncertainty
  - Polynomial Chaos (PC) Stochastic Galerkin methods
    - Intrusive/non-intrusive
  - Stochastic Collocation methods
- Model comparison, selection, and validation
- Model averaging
- Experimental design and uncertainty management

QUEST Team

Institution	Participants
SNL	**H. Najm**, M. Eldred, B. Debusschere, J. Jakeman, K. Chowdhary, C. Safta, K. Sargsyan
USC	**R. Ghanem**
Duke	**O. Knio**, O. Le Maître, F. Rizzi, J. Winokur
UT	**O. Ghattas**, R. Moser, C. Simmons, A. Alexanderian T. Bui-Thanh, K. E.-Hiroms, N. Petra, G. Stadler
LANL	**D. Higdon**, J. Gattiker
MIT	**Y. Marzouk**, P. Conrad, T. Cui, A. Gorodetsky, M. Parno

Team Expertise and Capabilities

Institution	Expertise	Tools
SNL	Forward and inverse UQ methods, design under uncertainty	DAKOTA UQTK
USC	Intrusive UQ methods probabilistic modeling	
Duke	Sparse adaptive forward UQ methods	
UT	Large scale inverse problems validation, inverse UQ	QUESO
LANL	Gaussian process modeling, inverse UQ	GPMSA
MIT	Calibration, adaptive sampling, inverse UQ, experimental design	MUQ

QUEST UQ Software tools

DAKOTA

- Optimization and calibration
- Non-intrusive UQ
- Global Sensitivity Analysis
- > 10K registered downloads

QUESO

- Bayesian Inference
- Parallel MultiChain MCMC
- Bayesian Model Analysis
- Model Calibration

GPMSA

- Bayesian Inference
- Gaussian Process Emulation
- Model Calibration
- Model discrepancy analysis

UQTk

- Intrusive PC UQ
- Non-intrusive sampling
- Customized sparse PCE
- Random fields

Computational Character of UQ Codes

- Bayesian inference
  - MCMC algorithms, serial, parallelism
  - many model samples
  - high dimensional parameter space
  - local maxima
- Intrusive UQ methods
  - new math, algorithms, software
  - compiler relevance for automatic source code transformation
- non-intrusive UQ methods
  - deterministic sampling – resilience / fault detection
  - high-dimensional integrals, classification/clustering
  - embarrassingly parallel but I/O bus contention possible

Challenges in PC UQ – High-Dimensionality

- Dimensionality n of the PC basis: $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_n\}$
  - number of degrees of freedom
  - $P + 1 = (n + p)!/n!p!$ grows fast with n
- Impacts:
  - Size of intrusive system
  - # non-intrusive (sparse) quadrature samples
- Generally $n \approx$ number of uncertain parameters
- Reduction of n:
  - Sensitivity analysis
  - Dependencies/correlations among parameters
  - Dominant eigenmodes of random fields
  - Manifold learning: Isomap, Diffusion maps
  - Sparsification: Compressed Sensing, LASSO

PC Quadrature in hiD

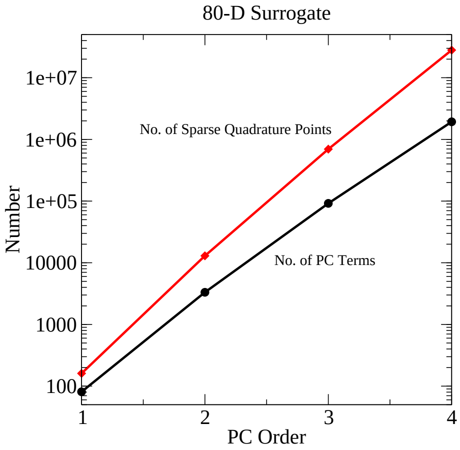
Full quadrature: $N = (N_{1\mathrm{D}})^n$

Sparse Quadrature

- Wide range of methods
- Nested & hierarchical
- Clenshaw-Curtis: $N = \mathcal{O}(n^p)$
- Adaptive – greedy algorithms

Number of points can still be excessive in hi-D

- Large no. of terms
- Reduction/sparsity



Areas of Mutual Interest – 1 – Performance Eng’g

- Non-intrusive UQ methods – sampling
  - Worksflow mgmt – managing samples, I/O – PEGASUS
  - Tiling – Mapping of N runs to N partitions, M-cores each
  - Managing I/O bus contention
    - Large number of runs with identical memory & file system access patterns
    - Filesystem virtualization
  - Parallel profiling DAKOTA/QUESO/UQTk/GPMSA – HPCToolkit
- Intrusive UQ methods
  - Profiling linear and non-linear solvers for specialized problem and matrix structures

Areas of Mutual Interest – 2 – Fault Tolerance

- MPI error handling
- Filesystem overload due to massive job concurrency
- pMPI communicator redefinition
  - Finer grain information for fault mgmt
  - Identifying and using healthy subsets of MPI partition
- Mechanisms for detecting both dead *and* slow nodes
- Dead sample runs, once detected, are easy to ignore, and proceed-without, in random sampling UQ methods
- Not so in quadrature, deterministic-sampling, UQ methods
  - Sparse quadrature integration with missing samples?
- Dead samples would have to be repeated in serial Markov Chain Monte Carlo (MCMC) algorithms
  - Parallel-chain MCMC methods

Areas of Mutual Interest – 3 – Compilers

- Opportunities for automatic source code transformation for enabling intrusive UQ methods
- Intrusive UQ methods: Original deterministic application code is modified to solve for the uncertain solution, examples:
  - Adjoint methods
  - Polynomial Chaos methods
- Key potential advantage: computational efficiency
- Significant challenges can exist in math and algorithms, depending on the problem at hand
- For some problems, the challenge is re-writing legacy codes $\rightarrow$ source transformation

DAKOTA – Performance – non-intrusive UQ

Concurrency in multiprocessor simulations

- Black box interfaces
  - job tiling — N runs, each on a partition with M cores
  - support varies depending on HPC system
- Library embedding
  - Share MPI communicator with application code
  - Application needs to be MPI-modular, using sub-partitions
  - No I/O, all in memory

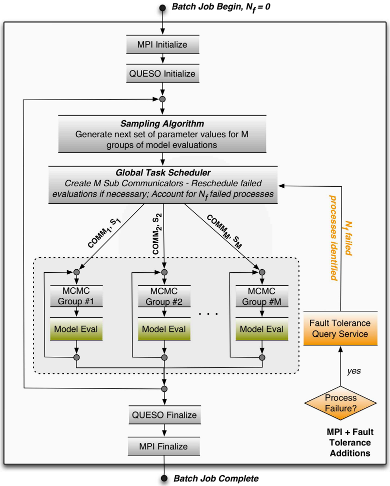
DAKOTA – Fault Tolerance – non-intrusive UQ

Resilience: Failure detection and mitigation

- Simulation failures due to parameter variation
  - 30% failures observed in an actual system code
  - Online recovery
    - failure capture $\rightarrow$ abort, retry, recover, continue
    - fault tolerant random sampling & regression-based PC (ignore sample)
  - Offline recovery
    - UQ & simulation restart
- File system overload to massive job concurrency
- Hardware & system software faults
  - need better MPI resilience

QUESO – Fault Tolerance

- UT interest in robustness to faults in statistical inverse problems
    - parallel MCMC chains
    - Intrusive/embedded UQ algorithms – adjoints, Hessians
- QUESO – Detect slow/dead nodes, recreate node lists for MCMC chains
    - Handles sub-communicators



Closure

- Broad range of ongoing work on UQ software and algorithms development
- A number of SciDAC partnership activities
- Potential areas of mutual interest with SUPER
  - Compilers
  - Performance
  - Fault tolerance
- Existing capabilities in DAKOTA & QUESO