Embedded Sampling–Based Uncertainty Quantification Approaches for Emerging Computer Architectures

SAND2015-1763C

Marta D'Elia[1] & Eric Phipps[1] & H. Carter Edwards[2] & Jonathan J. Hu[2] & Siva Rajamanickam[2]

[1]Optimization and Uncertainty Quantification, [2]Scalable Algorithms, Sandia National Laboratories, NM

Objectives

- Enable accurate predictive simulation for many problems that are intractable today using uncertainty quantification (UQ) as a scientific driver for pushing to the exascale.
- Improve the performance of UQ approaches that repeatedly sample deterministic simulation codes at different realizations of the input data, resulting in performance limited to that of each realization.
- Achieve high performance in the simulation of PDEs on multicore (CPU/GPU/Accelerator) architectures addressing the following problems:
  – Random, uncoalesced memory accesses,
  – Inability to exploit consistent vectorization.

Approach

Many UQ approaches are based on the solution of a PDE for several realizations of input parameters, sampled from a probability density function, e.g.

$$\nabla \cdot (a(\mathbf{x}, \mathbf{y}_i)\nabla u) = g \quad \text{in } \Omega \quad \text{for some samples } \{\mathbf{y}_i\}_{i=1}^M. \tag{1}$$

In many cases, the code path, processor instructions, and memory access patterns are very similar from realization to realization.

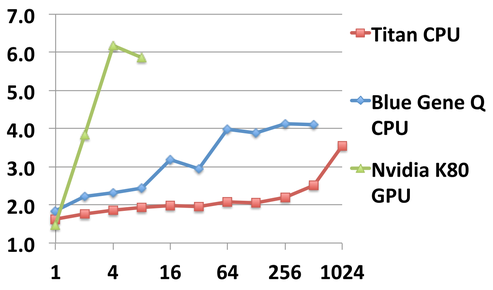
Idea: Propagate a collection (**ensemble**) of samples together through the forward simulation: given the PDEs $f(u, \mathbf{y}_i) = 0$, $i = 1, ...m$

- discretize via finite elements (FE): $F(U, Y_i) = 0$, $i = 1, ...m$
- propagate: $\mathbf{F}(\mathbf{U}, \mathbf{Y}) = \mathbf{0}$, where

  Ensemble FE solution: $\mathbf{U} = \sum_{i=1}^m U_i \otimes \mathbf{e}_i$
  Ensemble input data: $\mathbf{Y} = \sum_{i=1}^m Y_i \otimes \mathbf{e}_i$
  Ensemble FE residual: $\mathbf{F} = \sum_{i=1}^m F(U_i, Y_i) \otimes \mathbf{e}_i$

The new approach:

- Turns sample–dependent parameters into small arrays;
- Increases the fine–grained parallelism: each sample within the ensemble can be assigned to a vector lane/CUDA thread;
- Improves memory access patterns: random memory accesses become block accesses (coalesced/packed);
- Enables sharing of non–sample–dependent data (e.g., mesh) between samples to reduce memory bandwidth;
- Amortizes MPI communication latency across ensemble.



Approach applied to C++ PDE codes via template–based generic programming: ***Stokhos*** *embedded uncertainty quantification library on top of* ***Kokkos*** *portable manycore performance library (H.C. Edwards, D. Sunderland, C. Trott).*

Figure: Speed-up vs compute nodes on a 64x64x64 mesh for the solution of $\mathbf{F}(\mathbf{U}, \mathbf{Y}) = \mathbf{0}$ with $m = 32$.

Problem setting

Solvers

- For the solution of $F(U, Y_i) = 0$ and $\mathbf{F}(\mathbf{U}, \mathbf{Y}) = \mathbf{0}$ we use the Conjugate Gradient (CG) method.
- We precondition the system using a smoothed aggregation algebraic multigrid (AMG) preconditioner implemented in the Trilinos package ML (R. Tuminaro, C. Siefert, J. Hu).
  NOTE: though very robust, the convergence of the AMG preconditioned CG may deteriorate if **1.** the computational grid is unstructured/anisotropic, **2.** the diffusion parameter is highly non–smooth.

The diffusion parameter

Given the covariance function $cov(\mathbf{x}, \mathbf{x}')$ with eigenvalues and eigenfunctions λ_n and $v_n(\mathbf{x})$, the parameter $a(\mathbf{x}, \mathbf{y})$ is defined via truncated Karhunen–Loève (KL) expansion:

$$a(\mathbf{x}, \mathbf{y}) = a_0 + e^{\sum_{n=1}^N \sqrt{\lambda_n} v_n(\mathbf{x})\, y_n} \quad \text{with} \quad cov(\mathbf{x}, \mathbf{x}') = e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{L^2}}.$$

Single–Parameter results

For $\Omega = [0, 1]^2$, $L = 0.1$, $N = 3$, and $\mathbf{y} \in \mathbb{R}^N$ such that $y_n \sim \mathcal{U}(-25, 25)$, we solve $F(U, Y) = 0$. In the table below we report the realizations $\mathbf{y}_i$, $\Delta a(\mathbf{y}_i) = \max_\Omega a(\mathbf{x}, \mathbf{y}_i) - \min_\Omega a(\mathbf{x}, \mathbf{y}_i)$, the condition number of the FE diffusion matrix $\kappa(D_i)$ and the number of CG iterations $\texttt{its}_i$ to reach convergence. In the figures below we report, for $\mathbf{y}_6$, $\mathbf{y}_8$, and $\mathbf{y}_9$, $a(\mathbf{x}, \mathbf{y})$ and $u(\mathbf{x}, \mathbf{y})$.

sample	$\Delta a(\mathbf{y}_i)$	$\kappa(D_i)$	$\texttt{its}_i$
$\mathbf{y}_1 = (8, -21, 18)$	2.1600e+03	1.1062e+07	47
$\mathbf{y}_2 = (-11, -16, 9)$	5.9669e+03	1.8700e+07	69
$\mathbf{y}_3 = (-14, -8, -11)$	2.5747e+03	5.9703e+06	61
$\mathbf{y}_4 = (22, -15, 4)$	2.9942e+00	3.7685e+04	27
$\mathbf{y}_5 = (-20, 12, 14)$	5.8283e+04	1.1907e+08	97
$\mathbf{y}_6 = (-1, 4, 6)$	1.3962e+01	7.7359e+04	28
$\mathbf{y}_7 = (-14, -22, 9)$	6.5345e+04	1.9956e+08	93
$\mathbf{y}_8 = (-23, 22, 11)$	7.6637e+05	1.7142e+09	113
$\mathbf{y}_9 = (-18, 6, 1)$	8.1776e+03	9.3598e+05	57
$\mathbf{y}_{10} = (1, -1, -7)$	7.3972e+00	4.9348e+04	27
$\mathbf{y}_{11} = (-6, 20, 16)$	2.0276e+04	7.8380e+07	80
$\mathbf{y}_{12} = (-13, -10, -9)$	1.9646e+03	4.8397e+06	58
$\mathbf{y}_{13} = (-9, -22, -4)$	9.1691e+03	3.1937e+07	77
$\mathbf{y}_{14} = (11, 17, 23)$	1.6592e+03	8.7855e+06	43
$\mathbf{y}_{15} = (2, -5, -22)$	9.0591e+03	3.9393e+06	35
$\mathbf{y}_{16} = (6, -19, 16)$	1.2703e+03	6.4215e+06	43

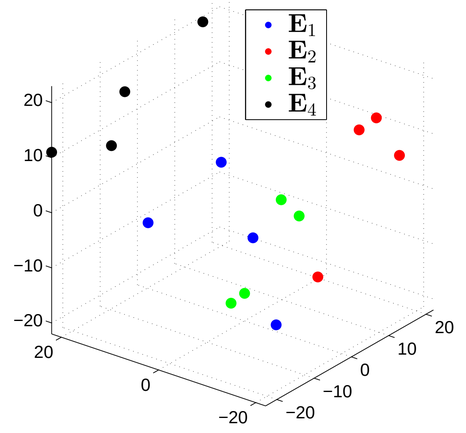
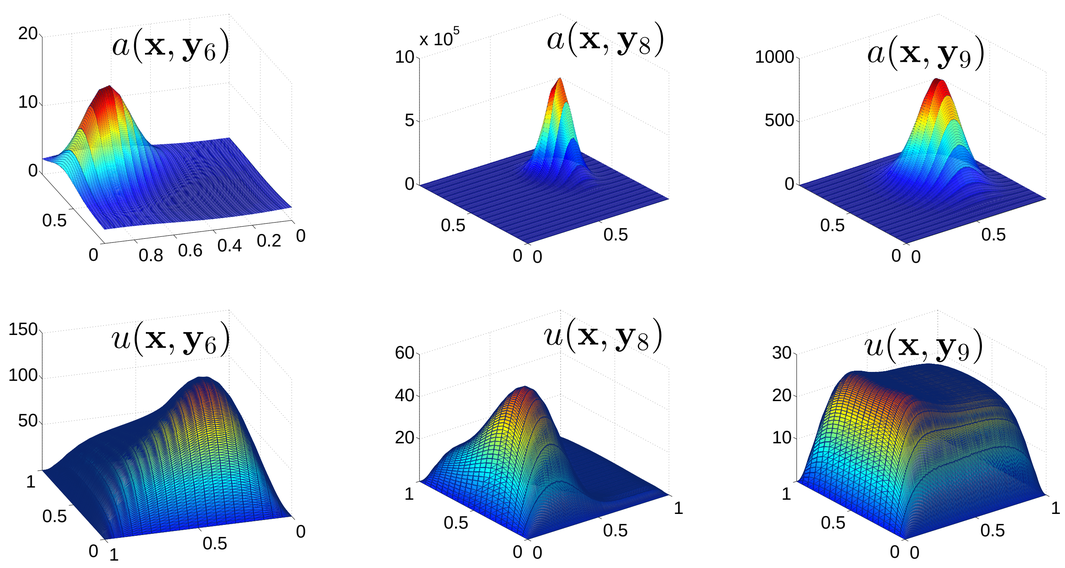


Figure: Samples $\mathbf{y}_1$–$\mathbf{y}_{16}$ grouped by mutual distance.



Note: when we form the ensemble system, the convergence of the linear solver is dictated by the sample that requires the highest number of iterations in a single–sample solve and, in most of the cases, is even worse.
GOAL: find an efficient **grouping** strategy.
Idea: use a grouping strategy that aggregates samples requiring similar number of iterations. We introduce three grouping strategies based on

1. *Variation of the diffusion parameter over* Ω;
2. *Condition number of the FE diffusion matrix*;
3. *Mutual distance of the samples in the space*.

Ensemble results

Using the strategies **1.**, **2.**, and **3.** we group the samples in 4 ensembles, $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathbf{E}_3$, $\mathbf{E}_4$, of size $m = 4$. We report the results in the table below. Strategy **0.** refers to grouping based on the number of iterations; r_j is related to the ratio between the cost of the ensemble approach and the single–sample's

$$\frac{m\ \texttt{its}_j}{\left(\text{speed–up}\right) \sum_{i=1}^m \texttt{its}_i} = \frac{r_j}{\left(\text{speed–up}\right)}.$$

	Ensembles	$\texttt{its}_j$	r_j	$\kappa(D_j)$
0.	$\mathbf{E}_1 = \{10, 4, 6, 15\}$	34	1.16	$\kappa(D_E)$=7.6198e+06
	$\mathbf{E}_2 = \{14, 16, 1, 9\}$	59	1.24	$\kappa(D_E)$=1.1562e+07
	$\mathbf{E}_3 = \{12, 3, 2, 13\}$	99	1.49	$\kappa(D_E)$=3.1937e+07
	$\mathbf{E}_4 = \{11, 7, 5, 8\}$	188	1.96	$\kappa(D_E)$=2.9381e+09
1.	$\mathbf{E}_1 = \{4, 10, 6, 9\}$	56	1.91	$\kappa(D_E)$=7.5931e+06
	$\mathbf{E}_2 = \{15, 16, 14, 12\}$	59	1.32	$\kappa(D_E)$=1.0875e+07
	$\mathbf{E}_3 = \{1, 3, 2, 13\}$	100	1.57	$\kappa(D_E)$=4.9514e+07
	$\mathbf{E}_4 = \{11, 5, 7, 8\}$	188	1.96	$\kappa(D_E)$=2.9381e+09
2.	$\mathbf{E}_1 = \{4, 10, 6, 9\}$	56	1.91	$\kappa(D_E)$=7.5931e+06
	$\mathbf{E}_2 = \{15, 12, 3, 16\}$	65	1.32	$\kappa(D_E)$=1.3379e+07
	$\mathbf{E}_3 = \{14, 1, 2, 13\}$	95	1.61	$\kappa(D_E)$=5.1755e+07
	$\mathbf{E}_4 = \{11, 5, 7, 8\}$	188	1.96	$\kappa(D_E)$=2.9381e+09
3.	$\mathbf{E}_1 = \{10, 9, 15, 6\}$	57	1.55	$\kappa(D_E)$=4.7551e+06
	$\mathbf{E}_2 = \{13, 16, 1, 4\}$	79	1.63	$\kappa(D_E)$=8.3199e+07
	$\mathbf{E}_3 = \{2, 7, 12, 3\}$	105	1.49	$\kappa(D_E)$=2.0333e+08
	$\mathbf{E}_4 = \{8, 5, 14, 11\}$	170	2.04	$\kappa(D_E)$=4.1230e+09

CONCLUSION

- Our approach allows us to achieve a significant speed–up for several architectures and different ensemble sizes.
- To exploit the speed–up an effective grouping strategy is mandatory.
- Strategy **1.** seems to be the most promising when the smoothness of parameters affects the convergence of linear solvers.

Future work: application to hierarchical stochastic collocation methods for UQ.

National Nuclear Security Administration
U.S. DEPARTMENT OF ENERGY | Office of Science

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.