Probabilistic Approaches for Resilience and Scalability in Extreme-Scale Computing

Bert Debusschere†, Khachik Sargsyan†, Francesco Rizzi†, Cosmin Safta†, Karla Morris†, Omar Knio‡, Habib Najm†

bjdebus@sandia.gov
†Sandia National Laboratories, Livermore, CA
‡Duke University, Durham, NC

SIAM UQ 2014 – Apr 3, 2014 – Savannah, GA

Supported by the US Department of Energy (DOE), Advanced Scientific Computing Research (ASCR)

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.
Outline

1. Background and Motivation
2. Algorithmic Approach
3. Application to 1D Linear Equation
4. Application to 1D Non-Linear Equation
5. Conclusions and Ongoing Work
6. Extra Material
Novel robust, scalable solvers are needed

- Exascale architectures present many daunting challenges to application codes
- Scalability will be key issue
  - Need to run on millions of cores
  - Communication expensive compared to floating point operations
- Robustness against wide range of faults required
  - Soft errors such as bit-flips introduce randomness
  - Loss of components due to hardware failures
  - Can not expect full machine to be up for any reasonable length of time
- Conventional approaches may not be effective
  - Time to save or restart from checkpoint may exceed MTBF (Mean Time Between Failures)
Treat faults as a source of uncertainty

- Probability Density Function (PDF) represents current state of knowledge about solution
  - Uncertainty from incomplete convergence
  - Uncertainty due to noisy or failed operations
- Targeted simulations refine knowledge
Probabilistic generalization of Schwarz coupling

- Decompose problem into overlapping subdomains
- Solve original problem with uncertain boundary conditions on each subdomain
Boundary maps accelerate convergence

- Write values of solution at boundaries as a function of each other
- Solution to system of equations gives full solution
- Similar to Aitken-like acceleration of Schwarz preconditioner [Garbey, SISC, 2005]

\[
\begin{align*}
y_{A,r} &= f_{B,A}(y_{B,l}) \\
y_{B,l} &= f_{A,B}(y_{A,r})
\end{align*}
\]
Bayesian inference of surrogate maps provides concurrency and resilience

\[ y_{A,r} = f_{B,A}(y_{B,l}; c_B) \]
\[ y_{B,l} = f_{A,B}(y_{A,r}; c_A) \]
\[ D = \{(y_{A,r}, y_{B,l})_n\}_{n=1}^N \]
\[ p(c|D) \propto p(D|c)p(c) \]

- Infer boundary maps from subdomain samples
- Bit-flips show up as noise in function evaluations
- Other faults (e.g. failing nodes) show up as missing data
Noise model needs to be appropriate

- $\ell_2$ noise model (Gaussian): assumes all data noisy
- $\ell_1$ noise model (Laplace): better suited when most data is exact but some points have large errors
- Analogy with compressed sensing: find solution with as few non-zero residuals as possible
With $\ell_1$ noise model, data corruption does not affect the response surface intersection.
Overall iteration process

- Sample the solution on subdomain boundaries over the range where solution is expected
  - Uniformly random samples
- Solve original equation on each subdomain for those sampled values
- Infer boundary maps on each subdomain from those samples
  - Parameterized as linear combination of Legendre polynomials
- Intersect the inferred boundary maps to get updated solution
  - Relying on Trilinos NOX and AztecOO solvers
- Set the boundary map range to the difference between the current and previous solution
  \[ \Delta_{k+1} = |y_k - y_{k-1}| \]
This approach readily generalizes to more subdomains

\[
\begin{align*}
    y_{B,r} &= f_{C,B}(y_{C,l}, y_{C,r}) \\
    y_{C,r} &= f_{D,C}(y_{D,l}, y_{D,r}) \\
    y_{D,l} &= f_{C,D}(y_{C,l}, y_{C,r}) \\
    y_{C,l} &= f_{B,C}(y_{B,l}, y_{B,r})
\end{align*}
\]

Same procedure but boundary maps are 2D
1D linear differential equation

- Linear 1D boundary value problem:
  \[ y''(x) + 2y'(x) + y(x) = x \cos(x) \]

- Solution:
  \[ y(x) = c_1 e^{-x} + c_2 xe^{-x} + 0.5 x \sin(x) - 0.5(\sin(x) - \cos(x)) \]
Linear problem converges in one iteration

- Use analytical solution on subdomains
- Linear boundary maps inferred from 15 samples per subdomain
- Error bars indicate sampling ranges
Bit-flip model perturbs sub-domain solver results

- Distribution resulting from bit-flip with probability $p$
- Flip randomly selected bit in 64 bit binary representation
- Discrete peaks correspond to flips in individual bits
- Emulates soft errors in sub-domain solver return results
- Remove values outside $[-1, 1]$ as outliers
Bit-flips do not affect convergence in linear case with $\ell_1$ noise model.

- Bit-flip probability, $p = 0.0015$, for both cases.
Probability of bit-flips does not affect convergence in linear case
With a discrete subdomain solver, error is bounded by discretization error

- 1D BVP discretized using 2nd-order centered differences
- Tridiagonal linear system: direct solver (Thomas algorithm)
- Bit-flip probability $p = 0.0015$ and $\ell_1$ model
With iterative subdomain solver, smaller subdomains improve convergence for fixed iteration count

- 401 total grid points
- Gauss-Seidel for tridiagonal system
- 300 iterations
- $p = 0.0015$
- $\ell_1$ noise model

Fewer grids points per subdomain gives better accuracy for fixed number of iterations, limited by discretization error
Consider non-linear differential equation:
\[(e^y)'' = \beta \sin(\alpha x), \alpha = 5 \text{ and } \beta = 10\]

Solution: 
\[y(x) = \log \left( -\frac{\beta}{\alpha^2} \sin(\alpha x) + ax + b \right)\]
The iterative approach converges for non-linear case

- Analytical solution on subdomains
- Linear boundary maps based on 15 samples
- Error bars show sampling ranges
Convergence is noisier with bit-flips

- Analytical solution used on subdomains
- 100 sample runs, 0.25, 0.5 and 0.75 quantiles
With discrete subdomain solver, accuracy is bounded by discretization error

- 10 subdomains with overlapping of 2 grid points
- Full grid: 51 points, \( \sim 8 \) per subdomain
- \( \Delta x = 0.06 \)
- 100 sample runs

- 200 Gauss-Seidel (GS) solver iterations on each subdomain
Smaller subdomains improve convergence for fixed iteration count

- Grid size 101 points
- Overlapping: 2 grid points
- $p = 0.0015$
- Gauss-Seidel
- 200 iterations

Fewer grids points per subdomain gives better accuracy for fixed number of iterations, limited by discretization error
Results remain good if number of subdomain samples reduced

- Minimum of $n = 4$ subdomain samples needed
  - linear 2D boundary map + hyperparameter
  - 12 samples corresponds to cost of three deterministic solves
- Bit-flip probability $p = 0.0015$
Conclusions and Ongoing Work

- Approach is designed to deal with faults
  - Silent / Soft errors such as bit-flips
  - Missing data due to communication issues or compute node failures
- Sampling approach provides concurrency
- Numerical examples show promising behavior in terms of convergence and resilience
- Ongoing work
  - More effective solution update
  - Extension to higher-dimensional computational domains
  - Detailed scalability and effectiveness study on more complex problems