The Resiliency of Multilevel Methods on Next Generation Computing Platforms: Probabilistic M

SAND2017-6952C

Christian Glusa
(Previously at Brown University, now at Sandia National Laboratories)

Joint work with Mark Ainsworth at Brown University

SIAM Annual Meeting 17
July 12, 2017

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Motivation - a problem of scale [10]

Nodes	Mean time to failure
1	100 years
1 million (exascale)	53 minutes

Faults

Classification of faults (Avižienis, Laprie, Randell, and Landwehr [3]):

- Hard faults - lead to program termination
  - Hardware failure
  - Software bugs
  - Petascale: every 16 hours
- Soft faults - corrupt application data and execution
  - Smaller transistors, energy efficient computing
  - Cosmic radiation causes bit flips
  - Petascale: every 5 hours

President Obama's executive order from summer 2015 to create a national strategic computing initiative:

[...] delivery of a capable exascale computing system that integrates hardware and software capability to deliver approximately 100 times the performance of current 10 petaflop systems [...]

Future large-scale computation will have to take faults into account.

Outline

1. Model for fault mitigation in iterative linear solvers
2. Products of random matrices
3. Analysis of fault-prone multigrid methods

Fault-prone Jacobi iteration & previous work

Global: $$\vec{x} \longrightarrow \vec{x} + D^{-1}\left(\vec{b} - A\vec{x}\right)$$

Distributed:

 $_1$ $$\vec{x}_1 \longrightarrow \vec{x}_1 + D_1^{-1}\left(\vec{b} - A\vec{x}\right)_1$$

 $_2$ $$\vec{x}_2 \longrightarrow \vec{x}_2 + D_2^{-1}\left(\vec{b} - A\vec{x}\right)_2$$

 $_3$ $$\vec{x}_3 \longrightarrow \vec{x}_3 + D_3^{-1}\left(\vec{b} - A\vec{x}\right)_3$$

Fault-prone Jacobi iteration & previous work

Global: $$\vec{x} \longrightarrow \vec{x} + D^{-1}\left(\vec{b} - A\vec{x}\right)$$

Distributed:

 1 $$\vec{x}_1 \longrightarrow \vec{x}_1 + \cancel{D_1^{-1}\left(\vec{b} - A\vec{x}\right)_1}$$

 2 $$\vec{x}_2 \longrightarrow \vec{x}_2 + D_2^{-1}\left(\vec{b} - A\vec{x}\right)_2$$

 3 $$\vec{x}_3 \longrightarrow \vec{x}_3 + D_3^{-1}\left(\vec{b} - A\vec{x}\right)_3$$

Previous work on Multigrid resilience

- Replication (Casas, Supinski, Bronevetsky, and Schulz [7]),
- Checkpointing (Calhoun, Olson, Snir, and Gropp [5]),
- Recovery (Huber, Gmeiner, Rüde, and Wohlmuth [11])

Fault-prone Jacobi iteration & previous work

Global: $\vec{x} \longrightarrow \vec{x} + D^{-1}\left(\vec{b} - A\vec{x}\right)$

Distributed:

 1 $\quad \vec{x}_1 \longrightarrow \vec{x}_1$

 2 $\quad \vec{x}_2 \longrightarrow \vec{x}_2 + D_2^{-1}\left(\vec{b} - A\vec{x}\right)_2$

 3 $\quad \vec{x}_3 \longrightarrow \vec{x}_3 + D_3^{-1}\left(\vec{b} - A\vec{x}\right)_3$

Our approach - *laissez-faire*

- Zero out values that are wrong or unavailable.
- Previous iterate is protected.

Model for laissez-faire mitigation

Fault-prone Jacobi iteration with laissez-faire mitigation:

$$\vec{x} \longrightarrow \vec{x} + \underbrace{\begin{pmatrix} \chi_1 & & & \\ & \chi_2 & & \\ & & \chi_3 & \\ & & & \ddots \end{pmatrix}}_{=:\boldsymbol{\mathcal{X}}} D^{-1}\left(\vec{b} - A\vec{x}\right)$$

- Diagonal fault matrix $\boldsymbol{\mathcal{X}}$ with entries

$$\chi_j = \begin{cases} 1 & \text{with probability } 1-\varepsilon \\ 0 & \text{with probability } \varepsilon \end{cases}$$

  and $\varepsilon \ll 1$ is the probability of a fault.
- χ_j can be dependent (node failure) or independent (detectable soft fault).
- Computation is subject to faults, but not data. (No persistent faults)

This is not a fault model, but a model for the remedial action taken by the laissez-faire approach.

Fault-prone two grid method for $A\vec{x} = \vec{b}$

Function $\mathcal{M}_{TG}\left(\textit{right-hand side } \vec{b},\ \textit{initial guess } \vec{x}\right)$

$\vec{x} \leftarrow \vec{x} + \theta D^{-1}\left(\vec{b} - A\vec{x}\right)$ `(Pre-smoothing on fine grid)`

$\vec{r} \leftarrow R\left(\vec{b} - A\vec{x}\right)$ `(Restriction to coarse grid)`

$\vec{d} \leftarrow A_C^{-1}\vec{r}$ `(Coarse-grid solve)`

$\vec{x} \leftarrow x + P\vec{d}$ `(Prolongation to fine grid)`

$\vec{x} \leftarrow \vec{x} + \theta D^{-1}\left(\vec{b} - A\vec{x}\right)$ `(Post-smoothing on fine grid)`

return $\vec{x}$

Fault-prone two grid method for $A\vec{x} = \vec{b}$

Function $\mathcal{M}_{TG}\left(\textit{right-hand side } \vec{b},\ \textit{initial guess } \vec{x}\right)$

$\vec{x} \leftarrow \vec{x} + \theta \boldsymbol{\mathcal{X}}^{(S,1)} D^{-1}\left(\vec{b} - A\vec{x}\right)$ `(Pre-smoothing on fine grid)`

$\vec{r} \leftarrow \boldsymbol{\mathcal{X}}^{(R)} R \boldsymbol{\mathcal{X}}^{(A)}\left(\vec{b} - A\vec{x}\right)$ `(Restriction to coarse grid)`

$\vec{d} \leftarrow A_C^{-1}\vec{r}$ `(Coarse-grid solve)`

$\vec{x} \leftarrow x + \boldsymbol{\mathcal{X}}^{(P)} P\vec{d}$ `(Prolongation to fine grid)`

$\vec{x} \leftarrow \vec{x} + \theta \boldsymbol{\mathcal{X}}^{(S,2)} D^{-1}\left(\vec{b} - A\vec{x}\right)$ `(Post-smoothing on fine grid)`

return $\vec{x}$

Fault-prone two grid method for $A\vec{x} = \vec{b}$

Function $\mathcal{M}_{TG}\left(\textit{right-hand side } \vec{b},\ \textit{initial guess } \vec{x}\right)$

$\vec{x} \leftarrow \vec{x} + \theta \boldsymbol{\mathcal{X}}^{(S,1)} D^{-1}\left(\vec{b} - A\vec{x}\right)$ (Pre-smoothing on fine grid)

$\vec{r} \leftarrow \boldsymbol{\mathcal{X}}^{(R)} R \boldsymbol{\mathcal{X}}^{(A)}\left(\vec{b} - A\vec{x}\right)$ (Restriction to coarse grid)

$\vec{d} \leftarrow A_C^{-1}\vec{r}$ (Coarse-grid solve)

$\vec{x} \leftarrow x + \boldsymbol{\mathcal{X}}^{(P)} P\vec{d}$ (Prolongation to fine grid)

$\vec{x} \leftarrow \vec{x} + \theta \boldsymbol{\mathcal{X}}^{(S,2)} D^{-1}\left(\vec{b} - A\vec{x}\right)$ (Post-smoothing on fine grid)

return $\vec{x}$

Random iteration matrix:

$$\boldsymbol{E_{TG}} = \left(I - \theta \boldsymbol{\mathcal{X}}^{(S,2)} D^{-1} A\right)\left(I - \boldsymbol{\mathcal{X}}^{(P)} P A_C^{-1} \boldsymbol{\mathcal{X}}^{(R)} R \boldsymbol{\mathcal{X}}^{(A)} A\right)\left(I - \theta \boldsymbol{\mathcal{X}}^{(S,1)} D^{-1} A\right)$$

Challenge: What is the asymptotic rate of convergence?
$\rightarrow$ Theory of products of random matrices

Lyapunov spectral radius [4, 8]

If E is non-random,

$$\rho(E) = \lim_{N\to\infty} \exp \frac{1}{N} \log \left\| \prod_{j=1}^{N} E \right\|. \qquad \text{(Gelfand's formula)}$$

If $\boldsymbol{E}$ is a random matrix with discrete support in $\mathrm{GL}_n(\mathbb{R})$, and $\boldsymbol{E_j}$ iid like $\boldsymbol{E}$.

$$\rho_L(\boldsymbol{E}) := \lim_{N\to\infty} \exp \mathbb{E}\left[\frac{1}{N} \log \left\| \prod_{j=1}^{N} \boldsymbol{E_j} \right\| \right] \qquad \text{(Lyapunov spectral radius)}$$

Lyapunov spectral radius [4, 8]

If E is non-random,

$$\rho(E) = \lim_{N\to\infty} \exp \frac{1}{N} \log \left\| \prod_{j=1}^{N} E \right\|. \qquad \text{(Gelfand's formula)}$$

If $\boldsymbol{E}$ is a random matrix with discrete support in $\mathrm{GL}_n(\mathbb{R})$, and $\boldsymbol{E_j}$ iid like $\boldsymbol{E}$.

$$\rho_L(\boldsymbol{E}) := \lim_{N\to\infty} \exp \mathbb{E}\left[\frac{1}{N} \log \left\| \prod_{j=1}^{N} \boldsymbol{E_j} \right\|\right] \qquad \text{(Lyapunov spectral radius)}$$

Theorem (Furstenberg and Kesten, 1960)

If $\mathbb{E}\left[\log^{+} \|\boldsymbol{E}\|\right] < \infty$, then $\rho_L(\boldsymbol{E})$ exists and with probability 1

$$\rho_L(\boldsymbol{E}) = \lim_{N\to\infty} \left\| \prod_{j=1}^{N} \boldsymbol{E_j} \right\|^{\frac{1}{N}}.$$

Convergence rate given by Lyapunov spectral radius of iteration matrix.

Behavior of the Lyapunov spectral radius

Example

Let $\boldsymbol{E}$ take values

$$\begin{pmatrix} 2 & 0 \\ 0 & \frac{1}{4} \end{pmatrix} \quad \text{w.p. } (1-\varepsilon) \qquad \text{and} \qquad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{w.p. } \varepsilon.$$

Then (by decomposition into cycles)

$$\rho_L(\boldsymbol{E}) = \begin{cases} 2 & \text{for } \varepsilon = 0, \\ \textbf{?} & \text{for } 0 < \varepsilon < 1, \\ 1 & \text{for } \varepsilon = 1. \end{cases}$$

Behavior of the Lyapunov spectral radius

Example

Let $\boldsymbol{E}$ take values

$$\begin{pmatrix} 2 & 0 \\ 0 & \frac{1}{4} \end{pmatrix} \quad \text{w.p. } (1-\varepsilon) \qquad \text{and} \qquad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{w.p. } \varepsilon.$$

Then (by decomposition into cycles)

$$\rho_L(\boldsymbol{E}) = \begin{cases} 2 & \text{for } \varepsilon = 0, \\ \sqrt{2}^{\varepsilon - 1} & \text{for } 0 < \varepsilon < 1, \\ 1 & \text{for } \varepsilon = 1. \end{cases}$$

- $\rho \geq 1$ for both matrices in the support of the random matrix, but $\rho_L < 1$
- ρ_L not necessarily continuous

Small probability of faults does not imply that $\rho_L(\boldsymbol{E}_{TG}) \approx \rho(E_{TG})$.

Finding the Lyapunov spectral radius (Crisanti, Paladin, and Vulpiani [8])

- Numerically: Simulation of trajectories of the Markov chain $\prod_{j=1}^{N} \boldsymbol{E_j} x_0$,
- Analytically: Only for small dimension or small support of $\boldsymbol{E}$.

Finding the Lyapunov spectral radius (Crisanti, Paladin, and Vulpiani [8])

- Numerically: Simulation of trajectories of the Markov chain $\prod_{j=1}^{N} \boldsymbol{E_j} x_0$,
- Analytically: Only for small dimension or small support of $\boldsymbol{E}$.

Lemma (Replica trick, Crisanti, Paladin, and Vulpiani [8])

Let $\boldsymbol{E}$ be a random square matrix. Then

$$\rho_L(\boldsymbol{E}) \leq \lim_{N\to\infty} \mathbb{E}\left[\left\|\prod_{j=1}^{N} \boldsymbol{E_j}\right\|^2\right]^{\frac{1}{2N}} = \sqrt{\rho\left(\mathbb{E}\left[\boldsymbol{E}\otimes\boldsymbol{E}\right]\right)} \leq \sqrt{\left\|\mathbb{E}\left[\boldsymbol{E}\otimes\boldsymbol{E}\right]\right\|}.$$

proof

Advantages:

- sub-multiplicative for independent matrices,
- almost sub-additive,
- second moment of a matrix instead of mean of log of matrix norm.

→ Plays nicely with classical multigrid analysis and fault matrices.

Fault-prone two grid method

Theorem (Ainsworth and CG)

Let A be the stiffness matrix of the finite-element discretization of a second order elliptic PDE on a quasi-uniform mesh of a C^2 domain.
Let usual conditions for two grid convergence be satisfied.
If prolongation, restriction, residual and damped Jacobi smoother are subject to component-wise detectable faults with probability ε, then

$$\rho_L\left(\boldsymbol{E}_{TG}\right) \leq \rho\left(E_{TG}\right) + C\varepsilon \begin{cases} n^{\frac{4-d}{2d}} & d < 4, \\ \sqrt{\log n} & d = 4, \\ 1 & d > 4, \end{cases}$$

where E_{TG} is the iteration matrix of the fault-free method, n is the number of degrees of freedom and d is the spatial dimension of the underlying PDE.

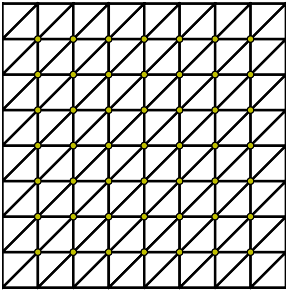
Does this capture the right behavior?

Test problem

Geometric two grid method for

$$\begin{cases} -\Delta u &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega. \end{cases}$$

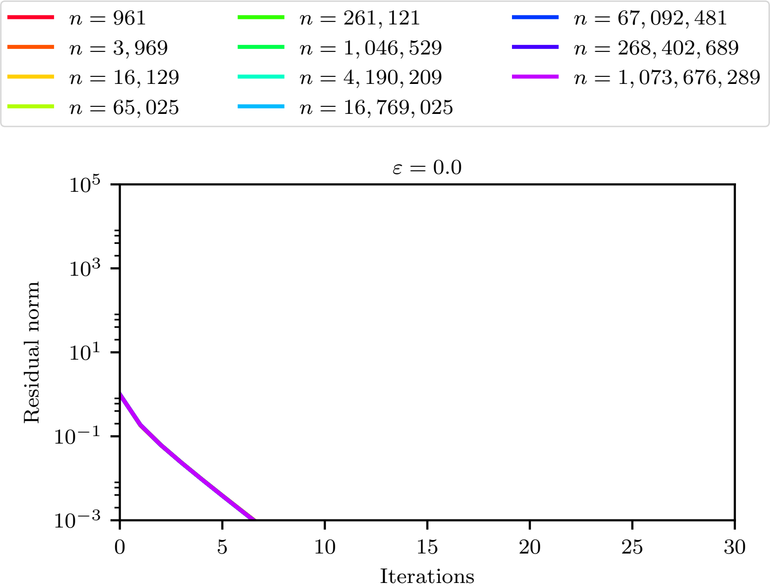
- piecewise linear FE,
- uniform mesh,
- damped Jacobi smoother.



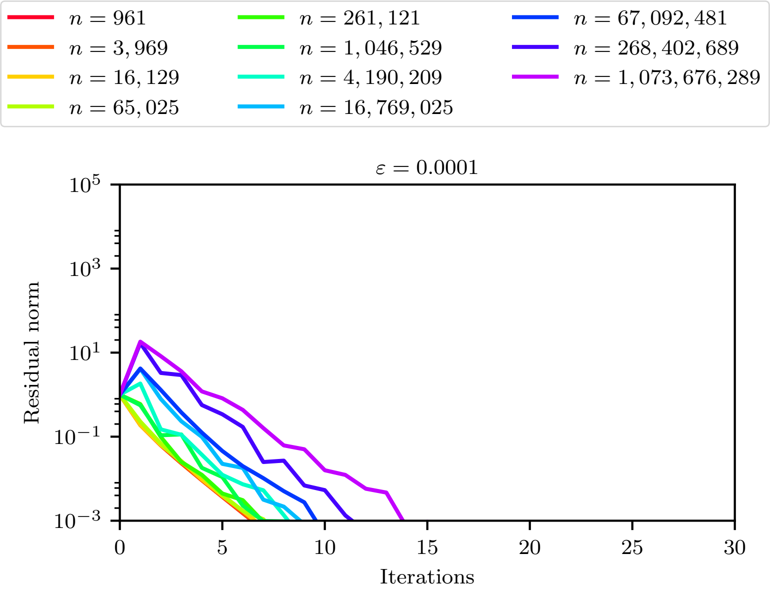
Bound on convergence rate:

$$\rho_L\left(\boldsymbol{E_{TG}}\right) \leq \rho\left(E_{TG}\right) + C\varepsilon\sqrt{n}.$$

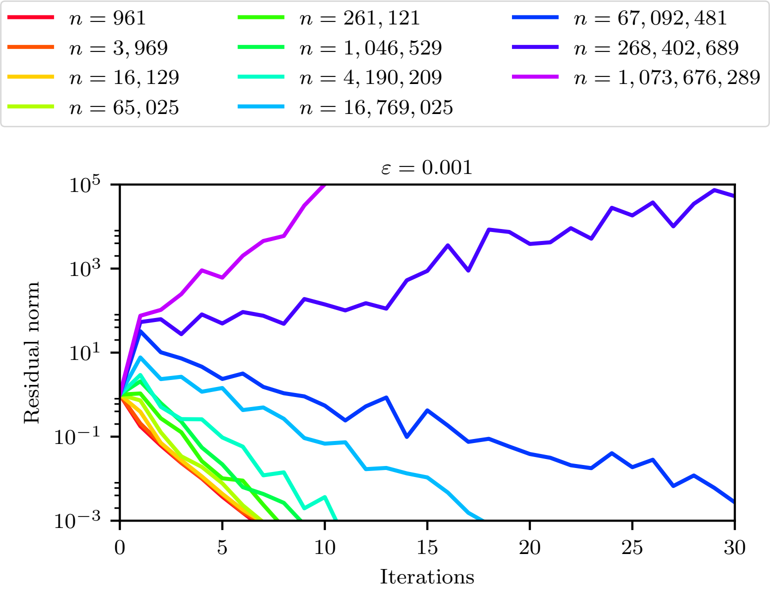
Evolution of residual norm



Evolution of residual norm

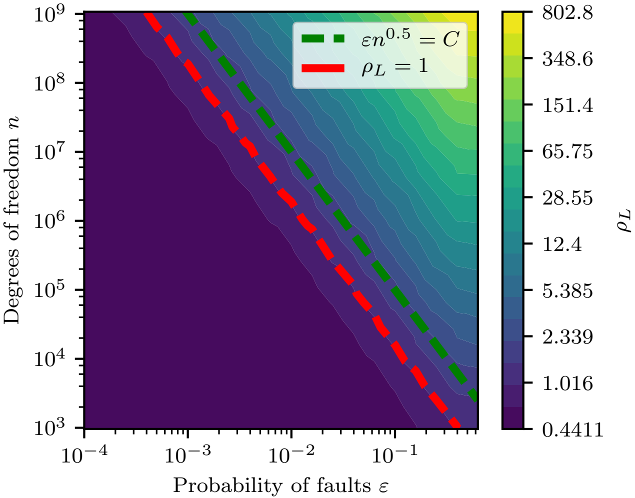


Evolution of residual norm



Rate of convergence $\rho_L\,(\boldsymbol{E_{TG}})$

11×20 data points $\times\ 1,000$ iterations $\approx 80,000$ CPU-hours



Bound is tight; method is not resilient.

Fault-prone multigrid method with protected prolongation

Theorem (Ainsworth and CG)

Let usual conditions for multigrid convergence be satisfied, so that

$$\|E_{TG}\|_2 \leq c < 1,$$

with constant c independent of n.
If restriction, residual and smoother are subject to corruption with fault probability ε, but the prolongation is protected, then

$$\rho_L\left(\boldsymbol{E_{TG}}\right) \leq \|E_{TG}\|_2 + C\varepsilon,$$

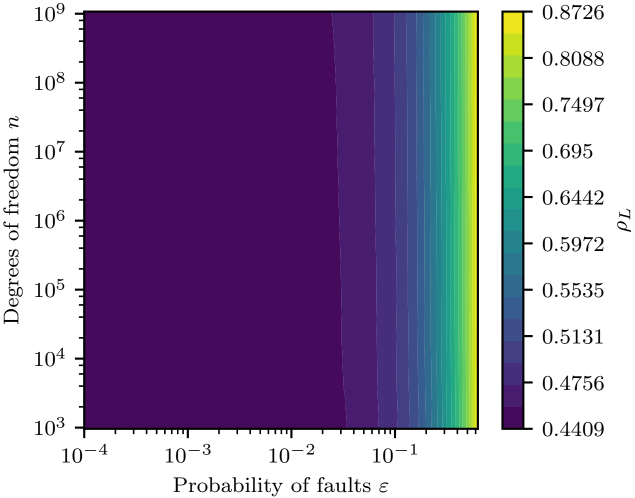
with constant C independent of n and ε.

Theorem (Ainsworth and CG)

Same holds for W-cycle multigrid.

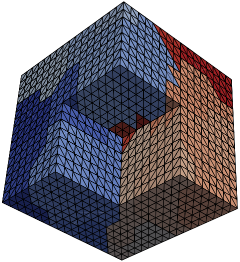
- Block smoothers are admissible if their structure matches the dependence pattern of the fault matrices.

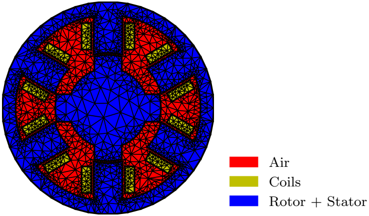
Rate of convergence $\rho_L\left(\boldsymbol{E_{TG}}\right)$ with protected prolongation



We observed similar behavior for

- V-cycle multigrid,
- higher order elements,
- adaptively refined meshes,
- solver hierarchies from algebraic multigrid,
- overlapping domain decomposition solvers.





Detection & Mitigation of Soft Faults in Practice

In practice, faults need to be

- detected – smoother, restriction, residual, prolongation
- zeroed out – smoother, restriction, residual
- corrected – prolongation

Detection in smoother, restriction and residual through replication

K replicas; accept value as correct, if all replicas match.

Detection and correction in prolongation through replication and voting

$K^{(P)}$ replicas; accept value as correct, if it is found by $M^{(P)}$ or more replicas. If no value has been accepted, proceed with laissez-faire mitigation.

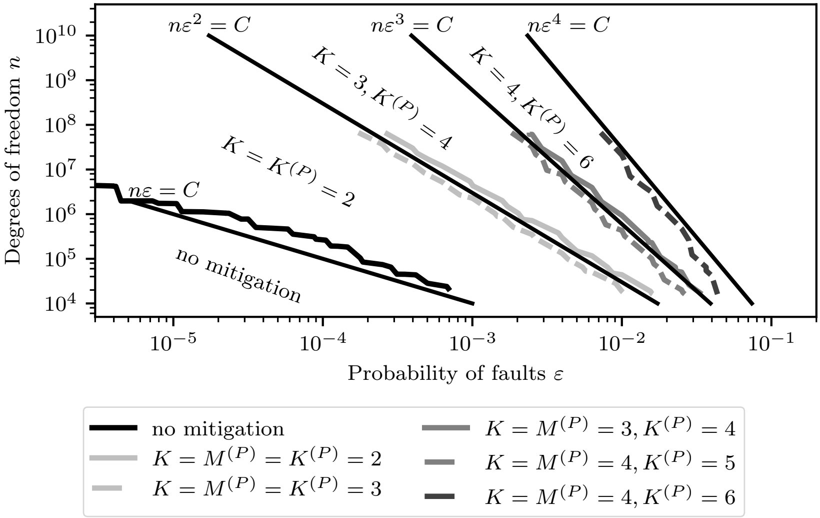
Work-optimal fault detection and protection strategy in 2D

If $n\varepsilon^{K-1} = \mathcal{O}(1)$, choose

$$K = M^{(P)}, \qquad K^{(P)} = 2K - 2,$$

to obtain convergence rate $\rho_L\left(\boldsymbol{E_{MG}}\right) \leq \|E_{MG}\|_2 + C\varepsilon$.

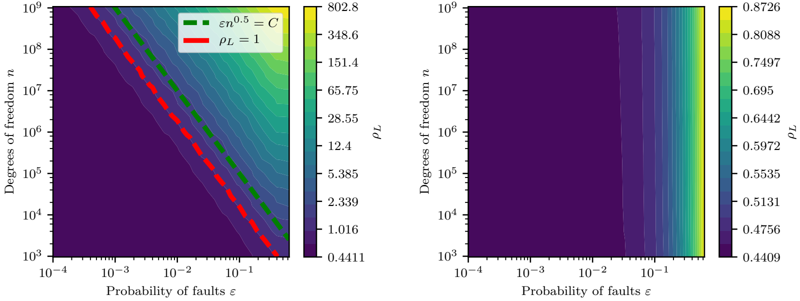
Level curves for different detection strategies, and optimal detection and protection parameters for each region.



Conclusion

- Fault-prone multigrid breaks down for large systems.
- Protecting the prolongation makes the algorithm fault resilient.
- Analytic convergence bounds in both cases.
- Theory informs choice of detection and protection strategies.

Thanks for listening!



References I

[1] Mark Ainsworth and Christian Glusa. "Is the multigrid method fault tolerant? The Multilevel Case.". In: *SIAM Journal on Scientific Computing* (Submitted).

[2] Mark Ainsworth and Christian Glusa. "Is the multigrid method fault tolerant? The Two Grid Case.". In: *SIAM Journal on Scientific Computing* (Accepted).

[3] Algirdas Avižienis et al. "Basic concepts and taxonomy of dependable and secure computing". In: *IEEE Transactions on Dependable and Secure Computing* 1.1 (2004), pp. 11–33.

[4] Philippe Bougerol and Jean Lacroix. *Products of random matrices with applications to Schrödinger operators*. Vol. 8. Progress in Probability and Statistics. Boston, MA: Birkhäuser Boston Inc., 1985, pp. xii+283.

[5] Jon Calhoun et al. "Towards a More Fault Resilient Multigrid Solver". In: *Proceedings of the High Performance Computing Symposium*. HPC 15. San Diego, CA, USA: Society for Computer Simulation International, 2015.

References II

[6] Franck Cappello et al. "Toward exascale resilience: 2014 update". In: *Supercomputing frontiers and innovations* 1.1 (2014), pp. 5–28.

[7] Marc Casas et al. "Fault Resilience of the Algebraic Multi-grid Solver". In: *Proceedings of the 26th ACM International Conference on Supercomputing*. ICS '12. San Servolo Island, Venice, Italy: ACM, 2012, pp. 91–100.

[8] Andrea Crisanti, Giovanni Paladin, and Angelo Vulpiani. *Products of random matrices*. Springer, 1993.

[9] Wolfgang Hackbusch. *Multi-grid methods and applications*. Vol. 4. Springer-Verlag Berlin, 1985.

[10] Thomas Herault and Yves Robert. *Fault-Tolerance Techniques for High-Performance Computing*. Springer, 2015.

[11] Markus Huber et al. "Resilience for Exascale Enabled Multigrid Methods". In: *CoRR* abs/1501.07400 (2015).

[12] Markus Huber et al. "Resilience for Multigrid Software at the Extreme Scale". In: *arXiv preprint arXiv:1506.06185* (2015).

References III

[13] Marc Snir et al. "Addressing failures in exascale computing". In: *International Journal of High Performance Computing Applications* 28.2 (2014), pp. 129–173.

Proof of replica trick

Let A a nonrandom square matrix. We write

$$\begin{aligned}
\|A\|_F^{2k} &= \operatorname{trace}\left(\left(AA^T\right)\right)^k \\
&= \operatorname{trace}\left(\left(AA^T\right)^{\otimes k}\right) \\
&= \operatorname{vec}\left(I^{\otimes k}\right) \cdot \operatorname{vec}\left(\left(AA^T\right)^{\otimes k}\right) \\
&= \operatorname{vec}\left(I^{\otimes k}\right) \cdot \operatorname{vec}\left(A^{\otimes k} I^{\otimes k} \left(A^T\right)^{\otimes k}\right) \\
&= \operatorname{vec}\left(I^{\otimes k}\right) \cdot A^{\otimes 2k} \operatorname{vec}\left(I^{\otimes k}\right).
\end{aligned}$$

Hence, if we call $\lambda_{j,\otimes 2k}$ and $v_{j,\otimes 2k}$ the eigenvalues and eigenvectors of $\mathbb{E}\left[\boldsymbol{A}^{\otimes 2k}\right]$, sorted in descending order with respect to their absolute value, we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\boldsymbol{A}^N\right\|_F^{2k}\right] &= \operatorname{vec}\left(I^{\otimes k}\right) \cdot \mathbb{E}\left[\left(\boldsymbol{A}^{\otimes 2k}\right)^N\right] \operatorname{vec}\left(I^{\otimes k}\right) \\
&= \operatorname{vec}\left(I^{\otimes k}\right) \cdot \mathbb{E}\left[\boldsymbol{A}^{\otimes 2k}\right]^N \operatorname{vec}\left(I^{\otimes k}\right) \\
&= \left[\operatorname{vec}\left(I^{\otimes k}\right) \cdot v_{1,\otimes 2k}\right]^2 \lambda_{1,\otimes 2k}^N + \mathcal{O}\left(\lambda_{2,\otimes 2k}^N\right).
\end{aligned}$$

back

Intuition and obtaining fault resilience

Split into low and high frequencies:

$$Av_{\text{low}} = \lambda_{\text{low}} v_{\text{low}}, \qquad PA_C^{-1}Rv_{\text{low}} \approx \frac{1}{\lambda_{\text{low}}} v_{\text{low}}, \qquad \boldsymbol{\mathcal{X}}^{(\bullet)} v_{\text{low}} \approx v_{\text{low}} + \epsilon v_{\text{high}},$$

$$Av_{\text{high}} = \lambda_{\text{high}} v_{\text{high}}, \qquad PA_C^{-1}Rv_{\text{high}} \approx 0, \qquad \boldsymbol{\mathcal{X}}^{(\bullet)} v_{\text{high}} \approx v_{\text{high}} + \epsilon v_{\text{low}}$$

Plug into fault-prone coarse-grid correction:

$$\left(I - \boldsymbol{\mathcal{X}}^{(P)} PA_C^{-1} R \boldsymbol{\mathcal{X}}^{(A)} A\right) v_{\text{high}} \approx \left(1 - \epsilon^2 \frac{\lambda_{\text{high}}}{\lambda_{\text{low}}}\right) v_{\text{high}} - \epsilon \frac{\lambda_{\text{high}}}{\lambda_{\text{low}}} v_{\text{low}}$$

- Low frequency error is removed by a single fault-free iteration.
- High frequency error accumulates.
- Protect an inexpensive operation: prolongation

$$\left(I - PA_C^{-1} R \boldsymbol{\mathcal{X}}^{(A)} A\right) v_{\text{high}} \approx v_{\text{high}} - \epsilon \frac{\lambda_{\text{high}}}{\lambda_{\text{low}}} v_{\text{low}}$$

Protecting the prolongation prevents accumulation of error (if intuition is correct).