On the resilience of conjugate gradient and multigrid methods to node failures

Carlos Pachajoa Wilfried Gansterer

University of Vienna

August 28, 2017

1. Introduction
2. Model setup
3. Algorithms
4. Experimental setup
5. Results
6. Conclusions

Table of Contents

1. Introduction
2. Model setup
3. Algorithms
4. Experimental setup
5. Results
6. Conclusions

Motivation

- This work starts with the question "How resilient is multigrid to node failures, and how do the recovery strategies used influence its resilience?"
- We use multigrid to solve a problem defined by a positive semidefinite matrix. We use the conjugate gradient method as a reference.
- Multigrid is optimal for some problems, while CG is the solver of choice for positive-definite matrices

Objectives

- Understand the influence of the of the location of the node failure, its size, the initial guess for the solution and the kind of interpolation used for the reconstruction of the vectors on the cost of recovery for multigrid and conjugate gradient.
- Evaluate the preconditioned conjugate gradient method also for more general problems taken from the SuitSparse Matrix Collection (Davis and Hu 2011).

Previous work I

Silent data corruption

(Mishra and Banerjee 2003)

- Application of Algorithm-Based Fault Tolerance to the detection of faults introduced in the interpolation and restriction operations of MG
- Two dimensional Poisson

(Casas et al. 2012)

- Impact of silent data corruption in an algebraic MG solver

(Ainsworth and Glusa 2017)

- Two-grid model that takes silent data corruption into account

(Altenbernd and Göddeke 2017)

- Detection and correction of soft failures for MG (Full approximation scheme) by exploiting invariants in the maps between levels

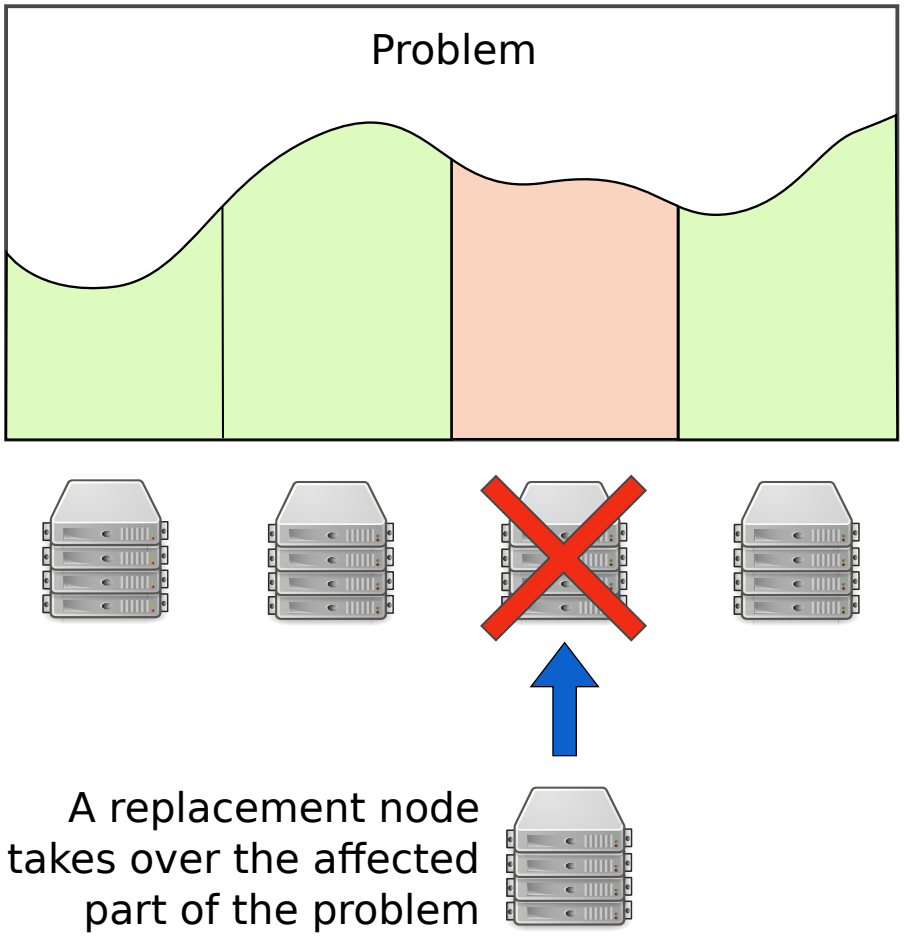
Previous work II

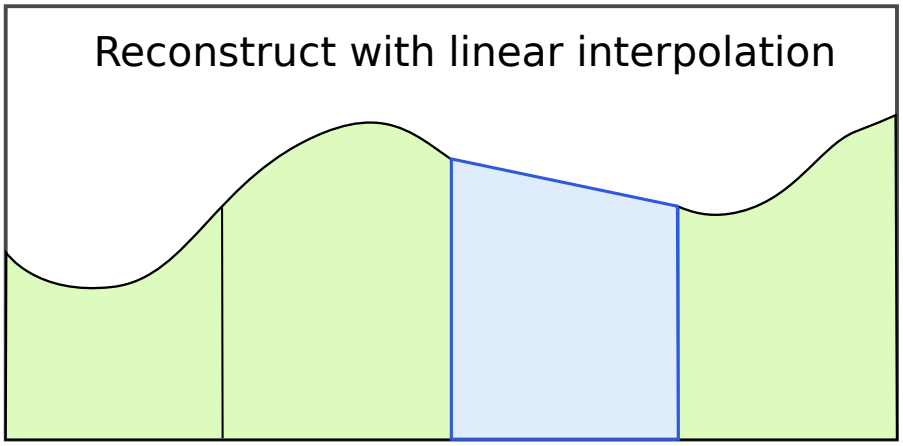
Node failures

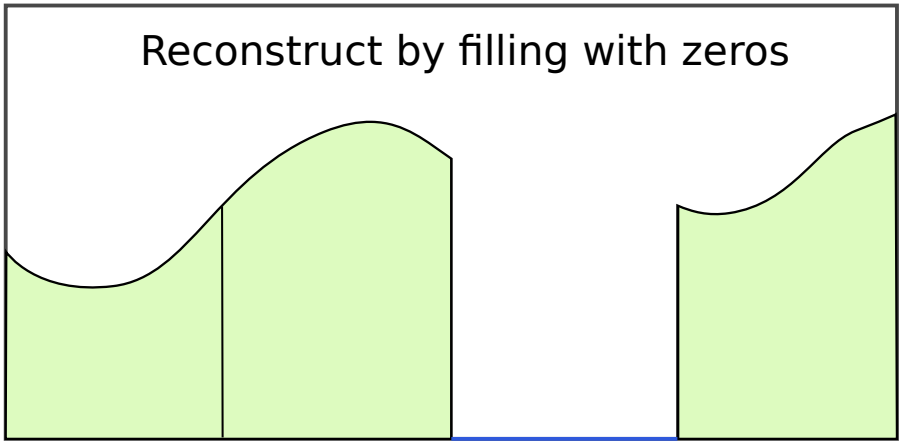
(Langou et al. 2007; Agullo et al. 2013; Agullo et al. 2016)

- Interpolation strategies to accelerate the recovery of Krylov solvers from node failures
- The strategies are costly but the reconstructed approximation has a smaller error norm than the approximation before the node failure
- We want to examine the impact of some of these interpolation strategies on MG

Problem description







Evaluation constellation

Different scenarios are considered for investigating the influence of the following parameters:

- Location of the node failure (Edge or center)
- Size of the set affected by the node failure
- Reconstruction strategy to be used (Linear interpolation or zero)
- Initial guess (Zero or uniformly distributed random numbers)

Table of Contents

1. Introduction
2. Model setup
3. Algorithms
4. Experimental setup
5. Results
6. Conclusions

Parallel model

- We assume that different nodes of a parallel machine work on contiguous, non overlapping sets of indices
- A node failure is represented as the loss of the information for the corresponding set of indices in all vectors
- We assume that the matrix and the right-hand side vector can be recovered from safe storage after a node failure

Data distribution among nodes

Poisson problem

- $2^k + 1$ grid points
- 2^m nodes, $m < k$
- The colors represent assignment to a node

SuitSparse matrices

- Vector size N is fixed

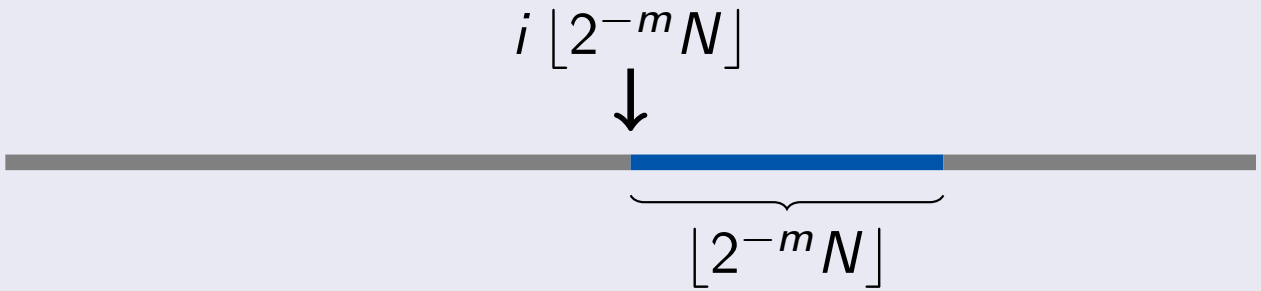
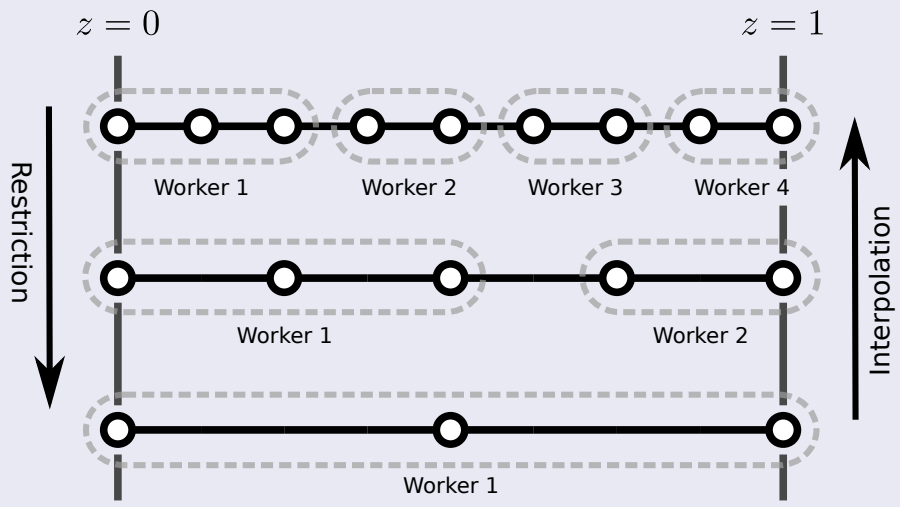


Table of Contents

1. Introduction
2. Model setup
3. Algorithms
4. Experimental setup
5. Results
6. Conclusions

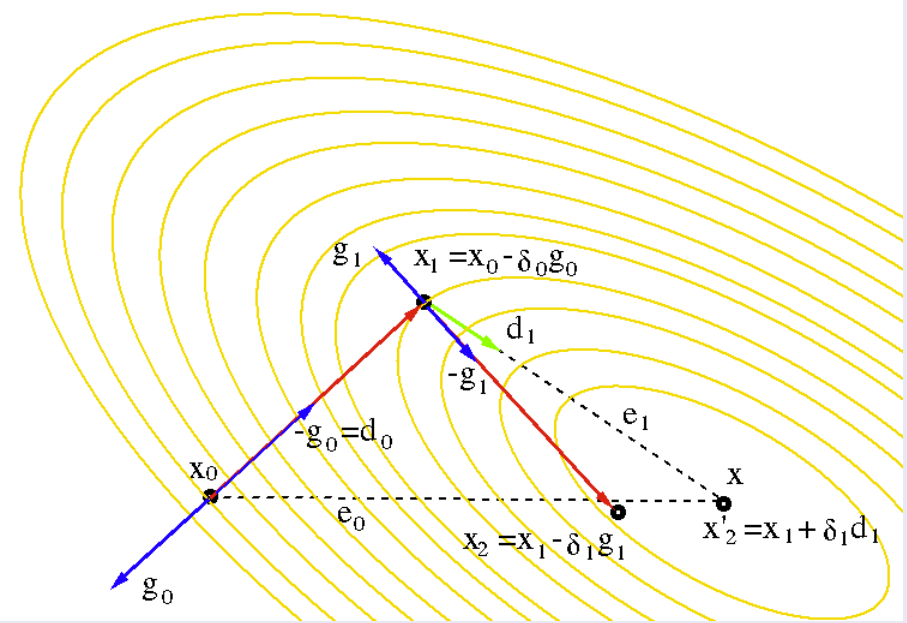
Algorithms

Geometric multigrid



- Faster solution by mapping the error between levels

Conjugate gradient



Taken from the lecture notes for *Numerical Methods in Engineering* (e176) of the harvey Mudd College. Nikos Drakos and Ross More. `http://fourier.eng.hmc.edu/e176/lectures/NM/node29.html`

- Solver of choice for SPD matrices

Table of Contents

1. Introduction
2. Model setup
3. Algorithms
4. Experimental setup
5. Results
6. Conclusions

Test problems: Poisson problem

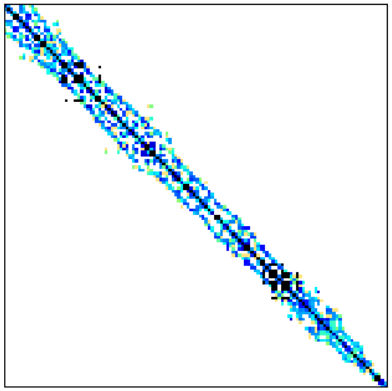
- A one-dimensional linear diffusion problem.

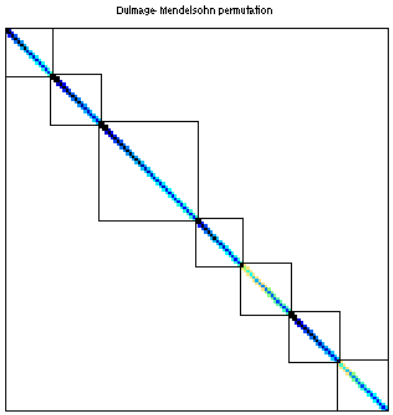
$$x : [0,1] \to \mathbb{R}, \quad \Delta x = f, \quad x(0) = x(1) = 0. \tag{1}$$

- Discretized with the three-point stencil $\frac{1}{h^2}\begin{bmatrix}1 & -2 & 1\end{bmatrix}$
- Problem size fixed at $2^{16} + 1$ grid points
- Right-hand side is a vector of ones
- Starting vector set to zero or drawn from a uniform probability distribution in the interval $[-0.5, 0.5]$
- Parallel workers with a set of grid points of size 2^6, 2^{10} and 2^{14}
- Terminate when the relative residual is below 10^{-5}
- Using MG, coarser grids are not reconstructed
- Also using non-preconditioned CG

Test problems: SuitSparse Matrix Collection

Name	bcsstk28	mhd4800b
Nickname	Wide-band	Narrow-band
Application	Solid mechanics	Magnetohydrodynamics
Rows $\times$ columns	4410×4410	4800×4800
Non-zeros	219024	27520





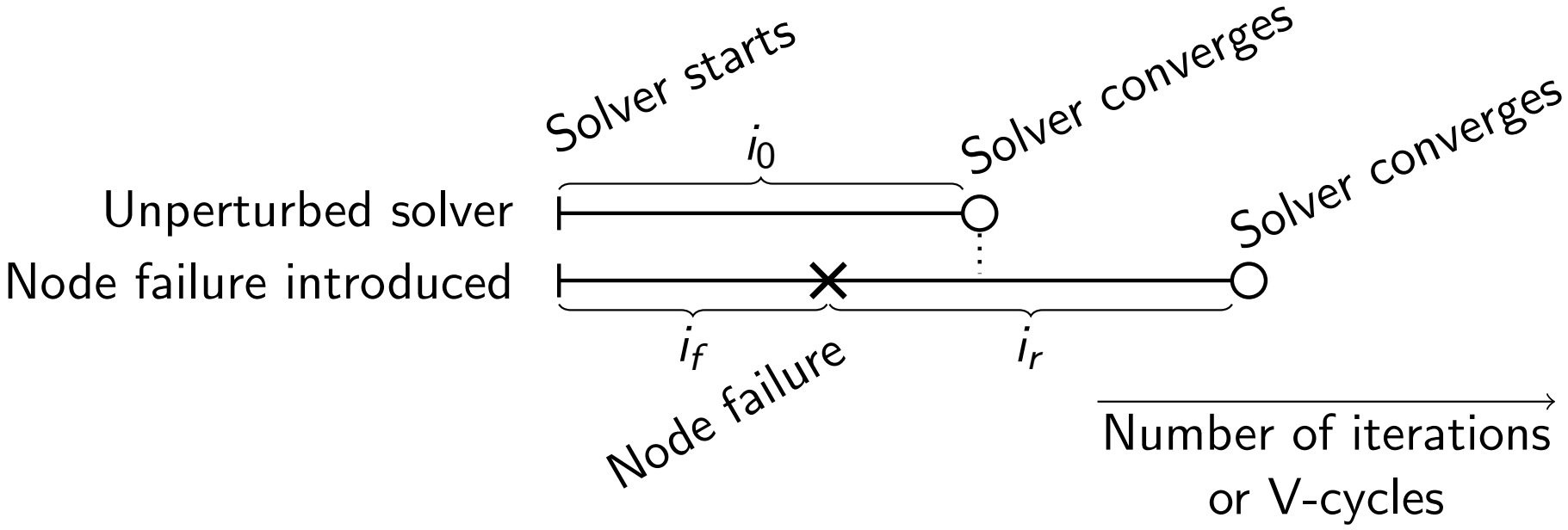
These matrices were selected because

- they represent one dimensional problems
- are positive definite

Test problems: SuitSparse Matrix Collection

- Solved with preconditioned CG
  - Jacobi preconditioner
  - Gauss-Seidel preconditioner
- Right-hand side is a vector of ones
- Starting vector set to zero or drawn from a uniform probability distribution in the interval $[-0.5, 0.5]$
- Lost set of grid points have a size of $1/4$, $1/8$ or $1/16$ of the number of grid points of the vector
- Terminate when the relative residual is below 10^{-5}
- Solved using PETSc

Metric

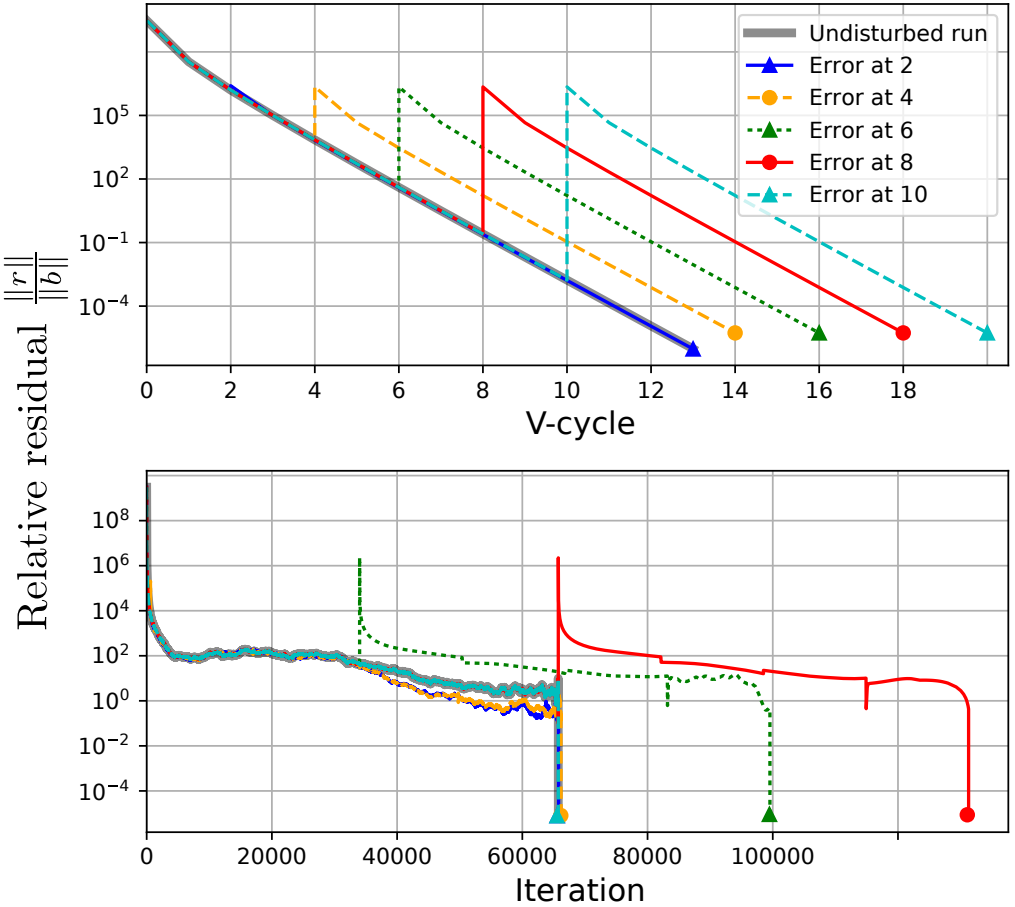


- *Relative overhead due to a node failure* $= \frac{i_f + i_r - i_0}{i_0}$
- Independent of the runtime

Table of Contents

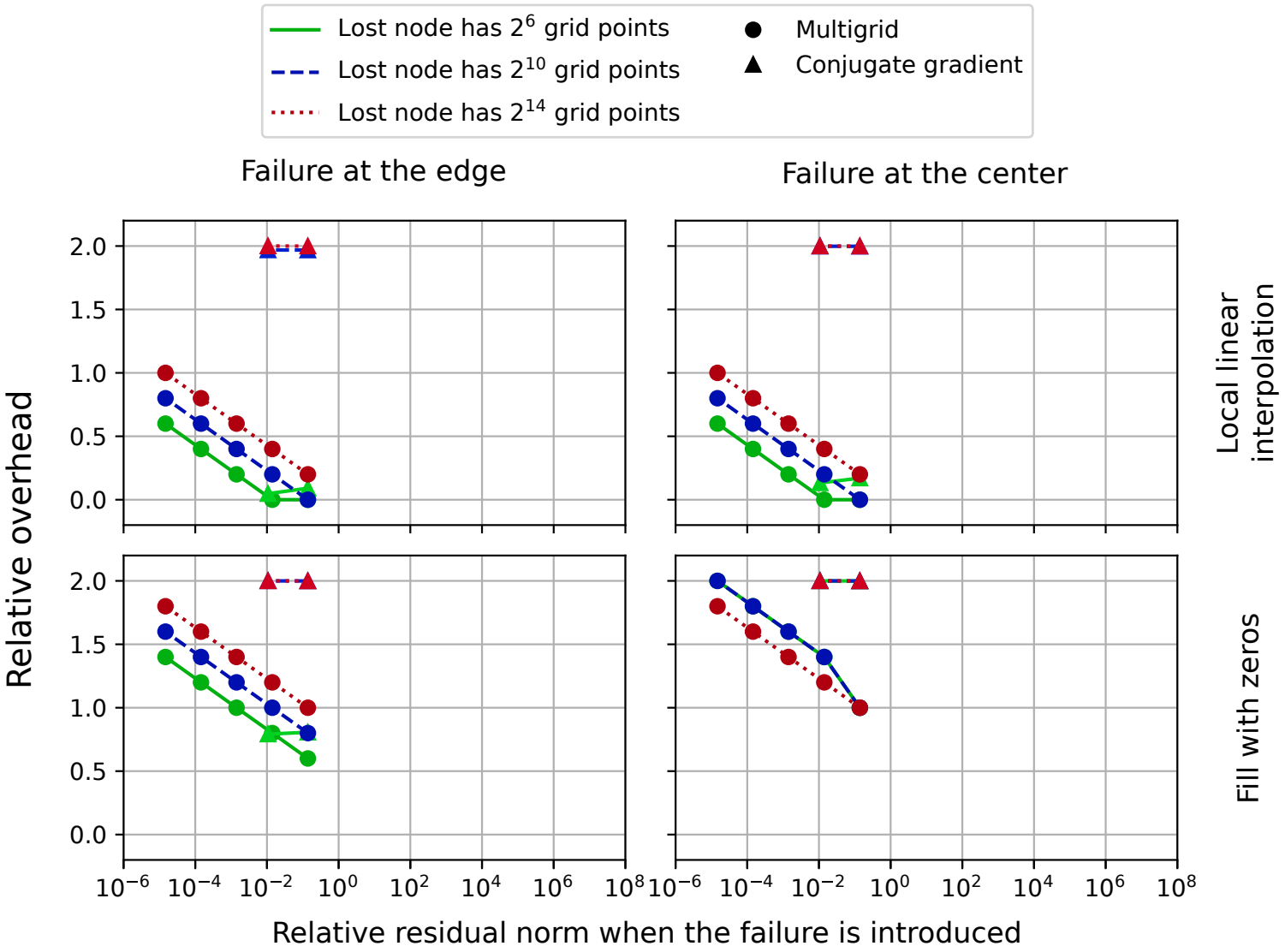
1. Introduction
2. Model setup
3. Algorithms
4. Experimental setup
5. Results
6. Conclusions

Example: Results for the Poisson problem

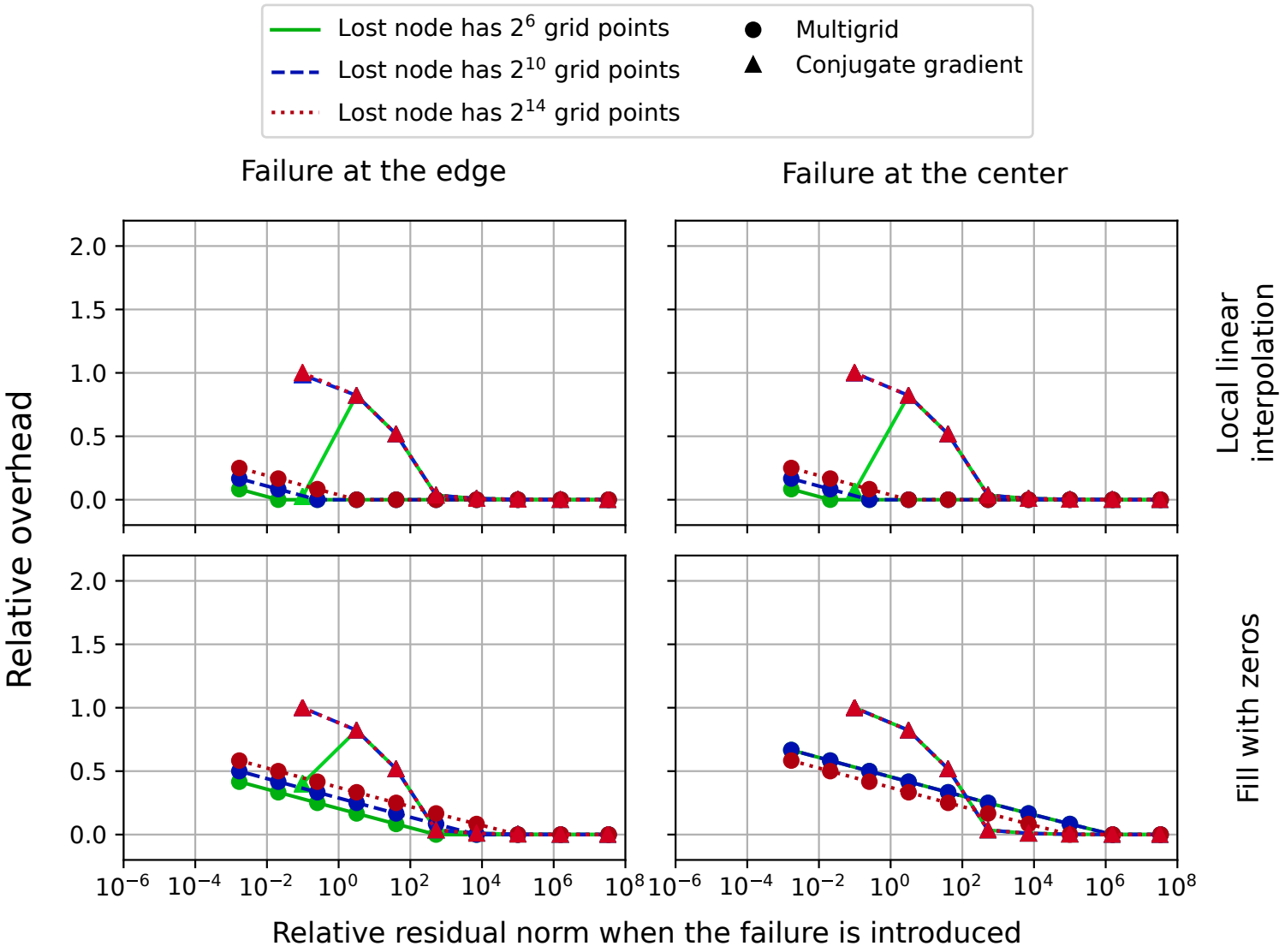


- In MG, node failures are introduced at fixed times
- In CG, node failures are introduced when the relative residual reaches the value that MG had at the time of a failure

Results for the Poisson problem: Initialization with zeros



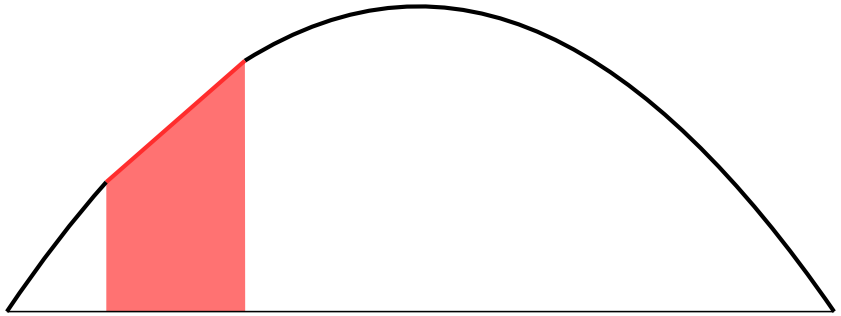
Results for the Poisson problem: Initialization with random values



Independence of the location of the node failure in some circumstances

- Under the following conditions:
  - Solving the one-dimensional Poisson problem with a constant right-hand side
  - A single node failure occurs
  - Reconstructing information of a lost node using linear interpolation
  - The solver is close to the solution

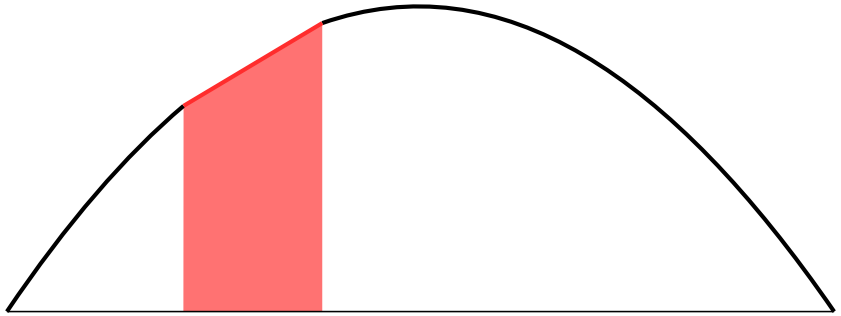
  one can show that the increase in the residual due to a node failure is independent of the location of the node failure
- Because of the almost constant rate of convergence of MG, the location of a node failure consequently has very little influence on the overhead



Independence of the location of the node failure in some circumstances

- Under the following conditions:
  - Solving the one-dimensional Poisson problem with a constant right-hand side
  - A single node failure occurs
  - Reconstructing information of a lost node using linear interpolation
  - The solver is close to the solution

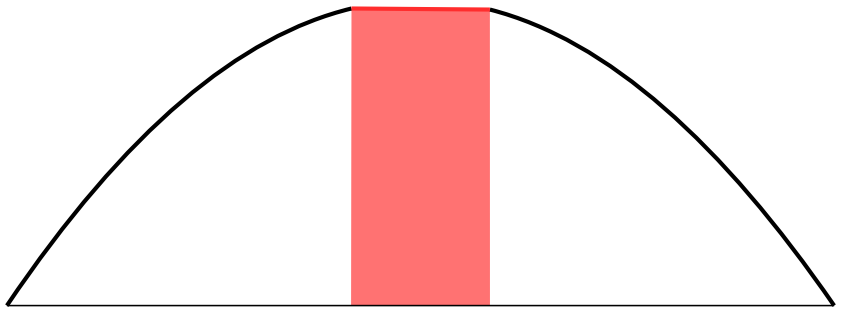
  one can show that the increase in the residual due to a node failure is independent of the location of the node failure
- Because of the almost constant rate of convergence of MG, the location of a node failure consequently has very little influence on the overhead



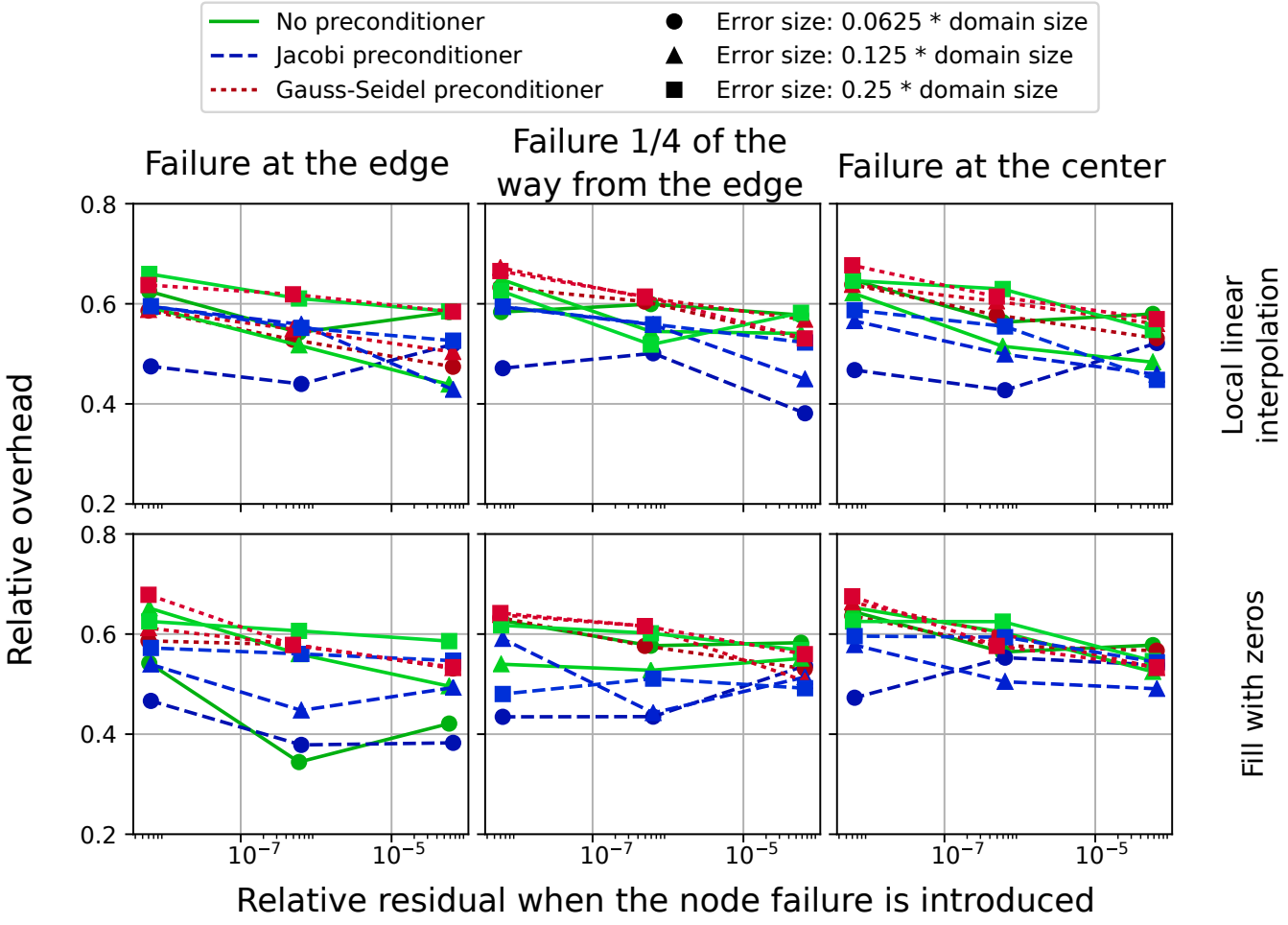
Independence of the location of the node failure in some circumstances

- Under the following conditions:
  - Solving the one-dimensional Poisson problem with a constant right-hand side
  - A single node failure occurs
  - Reconstructing information of a lost node using linear interpolation
  - The solver is close to the solution

  one can show that the increase in the residual due to a node failure is independent of the location of the node failure
- Because of the almost constant rate of convergence of MG, the location of a node failure consequently has very little influence on the overhead

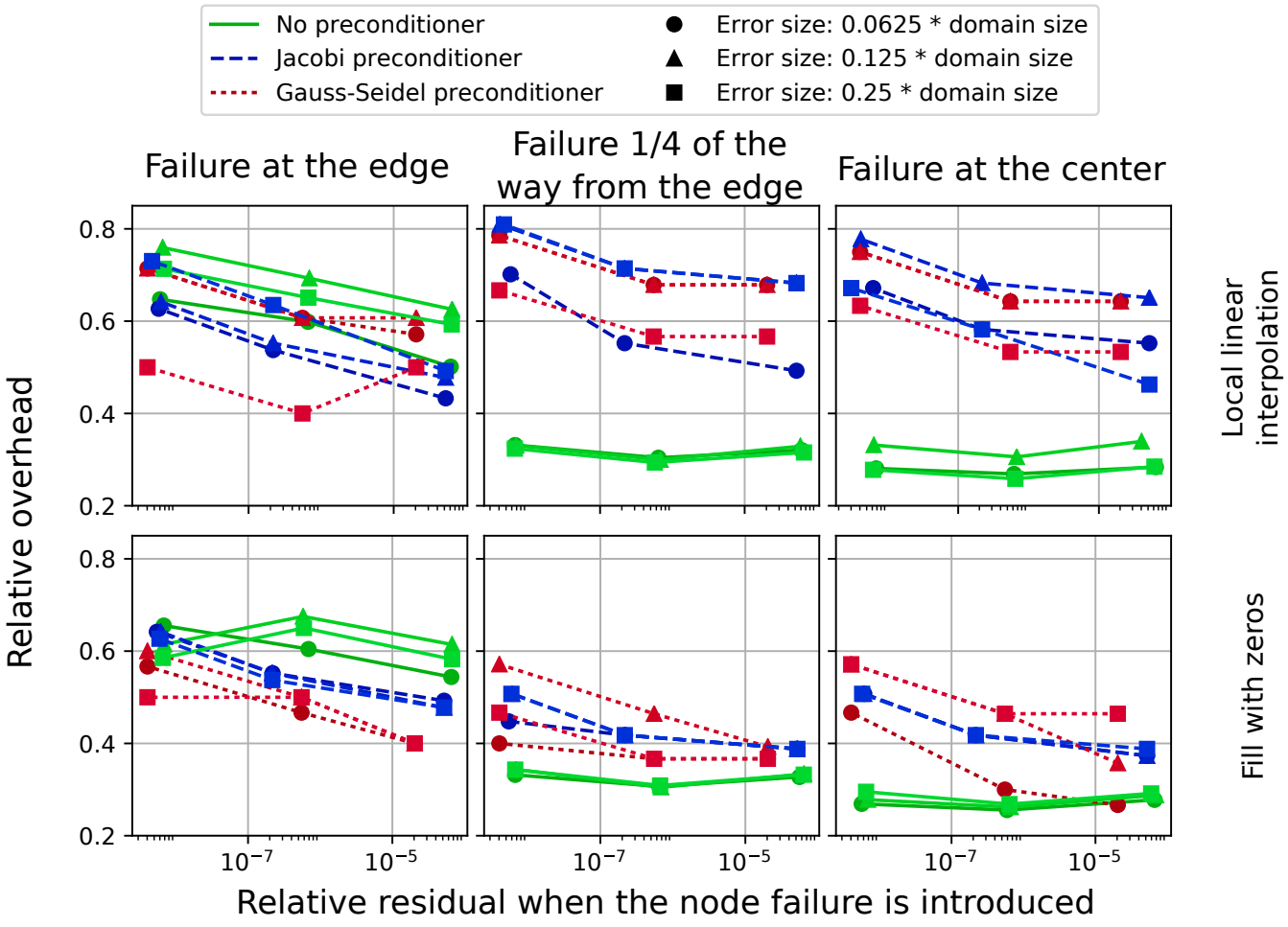


Results for the sparse matrices: Wide-band matrix



These plots correspond to a randomized initial guess. The results are roughly the same for the zero-initialized vector

Results for the sparse matrices: Narrow-band matrix



These plots correspond to a randomized initial guess. The results are roughly the same for the zero-initialized vector

Table of Contents

1. Introduction
2. Model setup
3. Algorithms
4. Experimental setup
5. Results
6. Conclusions

Conclusions for the Poisson problem

- Proximity to the solution strongly influences MG
- Relative overhead is smaller for random initial guesses
- For the reconstruction with linear interpolation and MG, the location of the node does not influence the overhead

Conclusions for the SuitSparse matrices

- Smaller overheads than for Poisson. The relative overhead is below one for all cases

Future work

- Examination of algebraic multigrid for SPD matrices
- Reconstruction of the state of the CG solver after a node failure

References I

[1] Emmanuel Agullo et al. "Numerical recovery strategies for parallel resilient Krylov linear solvers". In: *Numer Lin Algebra Appl* 23.5 (2016), pp. 888–905.

[2] Emmanuel Agullo et al. *Towards resilient parallel linear Krylov solvers: recover-restart strategies*. Research Report RR-8324. INRIA, 2013, p. 36.

[3] Mark Ainsworth and Christian Glusa. "Is the Multigrid Method Fault Tolerant? The Two-Grid Case". In: *SIAM J. Sci. Comput.* 39.2 (2017), pp. C116–C143. eprint: https://doi.org/10.1137/16M1100691.

[4] Mirco Altenbernd and Dominik Göddeke. "Soft fault detection and correction for multigrid". In: *The International Journal of High Performance Computing Applications* 0.0 (2017), p. 1094342016684006. eprint: http://dx.doi.org/10.1177/1094342016684006.

[5] Marc Casas et al. "Fault Resilience of the Algebraic Multi-grid Solver". In: *Proceedings of the 26th ACM International Conference on Supercomputing*. 2012, pp. 91–100.

[6] Timothy A. Davis and Yifan Hu. "The University of Florida Sparse Matrix Collection". In: *ACM Trans. Math. Softw.* 38.1 (2011), 1:1–1:25.

[7] Kuang-Hua Huang and J. A. Abraham. "Algorithm-Based Fault Tolerance for Matrix Operations". In: *IEEE Trans. Comput.* 33.6 (1984), pp. 518–528.

[8] J. Langou et al. "Recovery Patterns for Iterative Methods in a Parallel Unstable Environment". In: *SIAM J. Sci. Comput.* 30.1 (2007), pp. 102–116.

References II

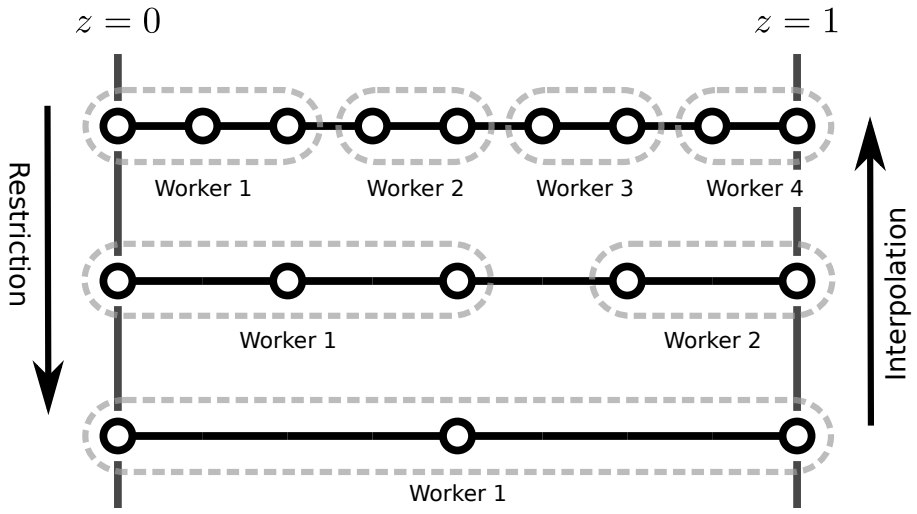
[9] A. Mishra and P. Banerjee. "An algorithm-based error detection scheme for the multigrid method". In: *IEEE Trans. Comput.* 52.9 (2003), pp. 1089–1099.

[10] U. Trottenberg, C. W. Oosterlee, and Anton Schüller. *Multigrid.* 2001.

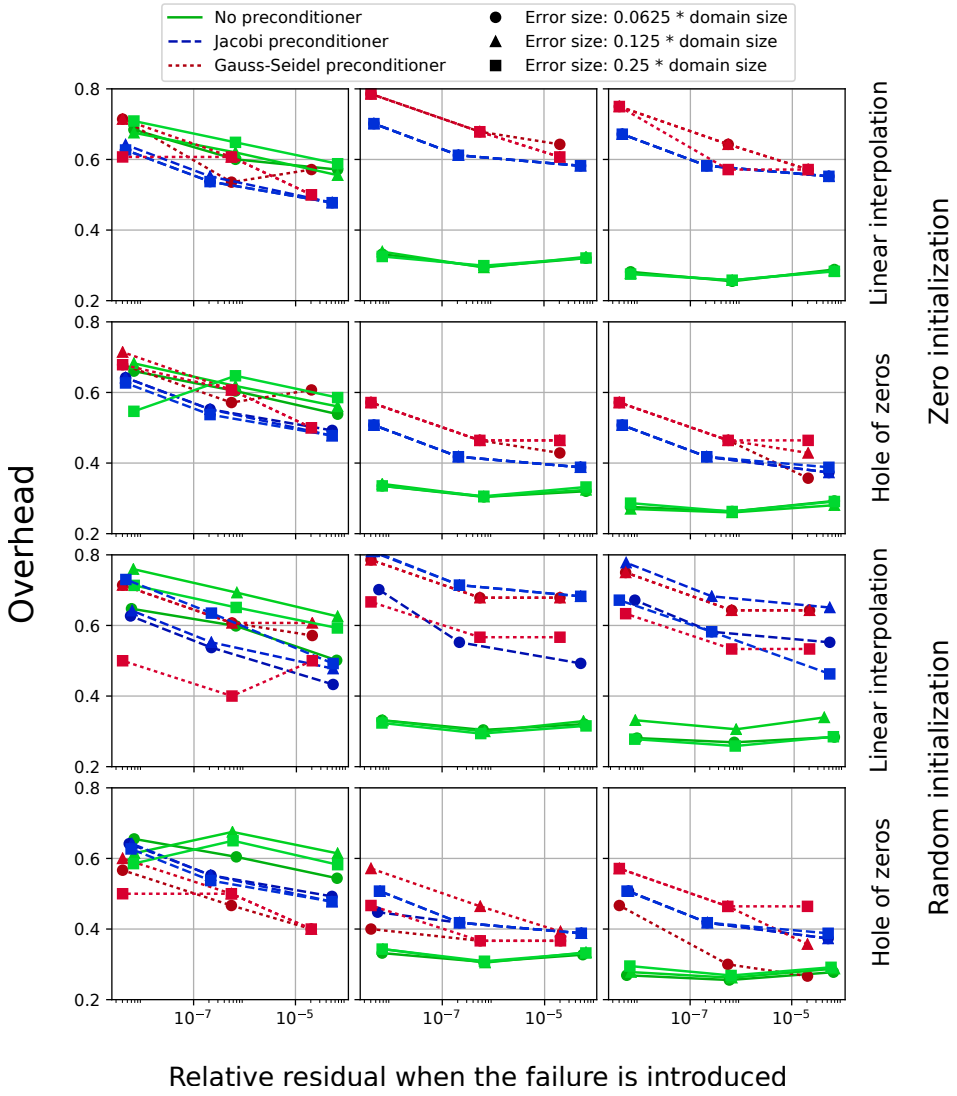
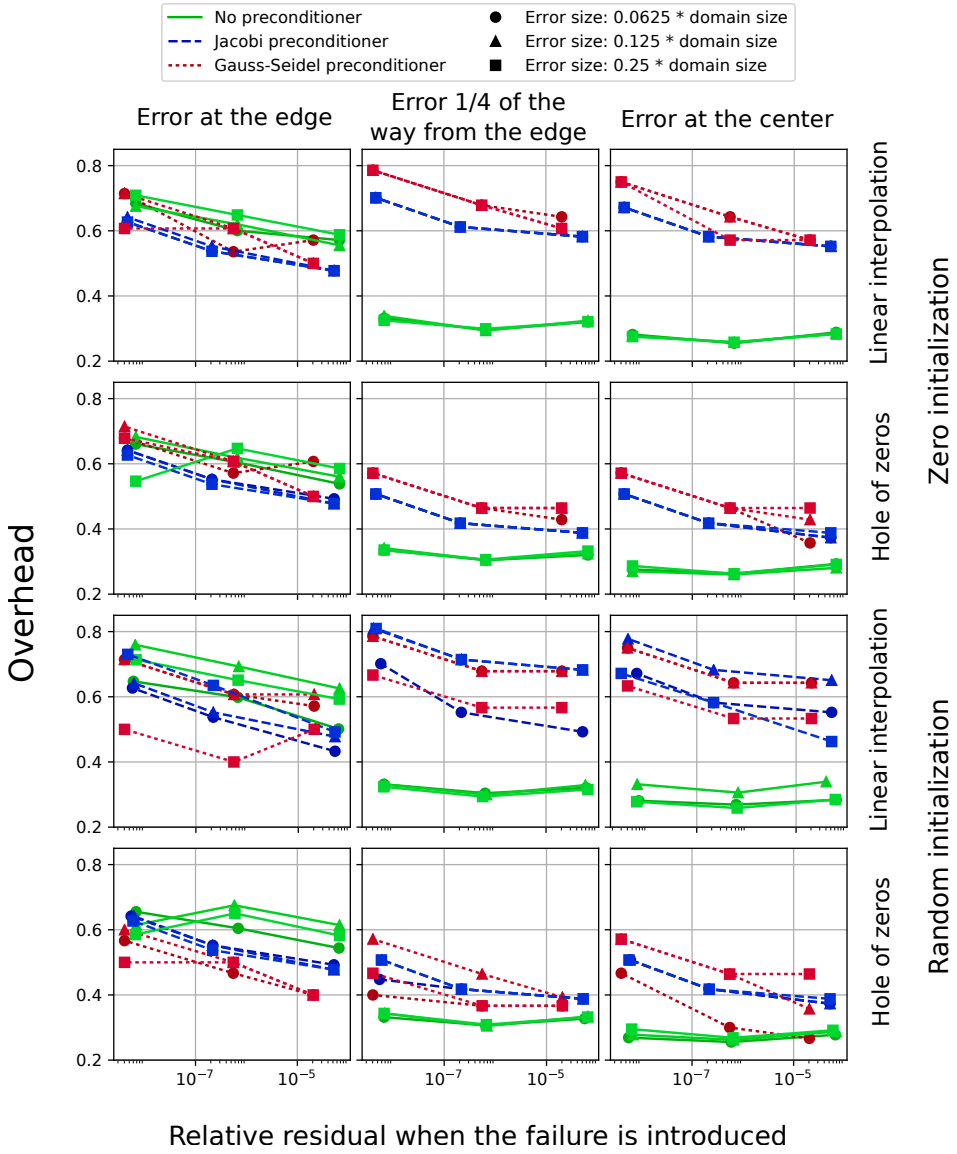
Backup slides: Geometric multigrid

Using Jacobi as a solver

- High-frequency error components can be quickly reduced
- Low-frequency error components take a long time to be reduced
- Low-frequency components can be transfered to a coarser grid, where they have a relatively higher frequency



Backup slides: Results for wide-band matrix



Backup slides: Results for narrow-band matrix