Exceptional service in the national interest

NC STATE UNIVERSITY

Making GMRES Resilient to Single Bit Flips

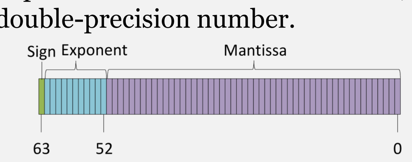
James Elliott — jjellio3@ncsu.edu
Mark Hoemmen — mhoemme@sandia.gov
Frank Mueller — mueller@cs.ncsu.edu

Project Overview

- Energy, power, and performance may make future hardware less reliable
- Data may experience *silent*, *transient* bit flips
- Floating-point numbers have a particular binary representation
- We can exploit this to detect some errors and bound almost all the rest
- Numerical algorithms can tolerate errors in certain code and data
- With some system cooperation, we can make numerical algorithms reliable despite data corruption and incorrect arithmetic

Floating Point Numbers

Figure 1. Binary representation for IEEE standard (Binary64) for storing a 64-bit double-precision number.



Features

(1) 51 bits for mantissa, 11 bits for the exponent stored using a bias of 1023, 1 bit for the sign, which is expressed in Eq. (1).

Examples (*mantissa and sign bits excluded*)

Base10	Exponent Bits b_{62} … b_{52}	Bias Relation	Effective Exp.
5	10000000001	$2^{1025-1023}$	2^2
2	10000000000	$2^{1024-1023}$	2^1
1	01111111111	$2^{1023-1023}$	2^0
.2	01111111100	$2^{1020-1023}$	2^{-3}
.5	01111111110	$2^{1022-1023}$	2^{-1}

Methodology

Perturbation Lookup Table for Dot Products

(1) Analysis tool used to evaluate processes based on dot products
(2) Allows determination of largest and smallest errors that *could* be injected into a dot product
(3) Prior to execution, a lookup table is created:
  (1) Dimension 2046 x 2046 x 11 x 3, accounting for all possible exponent biases (0-2046), 11 exponent bits, three operands to fault on – operands to multiply or the result (operand to summation)
  (2) Each entry in the table contains the absolute error injected should a bit flip happen in either of the two magnitudes, or should the bit flip occur in the result of the.
  (3) Mantissa impact is accounted for by assuming a value larger than the largest possible mantissa value, equivalent to incrementing the magnitude biases by 1.

Possible Arnoldi Process Perturbations

(1) Min and Max of each basis vector is computed after they are built
(2) Using min and max, an interval of potential exponent biases is determined.
(3) Possible perturbations are queried from lookup table

Analytical Bounds

(1) Norm bounds on the dot product (Cauchy-Schwarz)
(2) Arnoldi normalization allows simplification to recurrence form based on largest singular value of the matrix

Perturbation Counting

(1) Consider all possible magnitude combinations for dot product
(2) Query lookup table comparing possible absolute error injected into dot product against interesting bounds (norm bound and 1)

Potential Fault Detection

(1) Norm bound calculated *a priori*
(2) Compare upper Hessenberg entries against bound to prevent theoretically impossible values (including non-numeric)

Error Minimization

(1) Previous work[1] shows benefit of scaling data such that values < 1
(2) Equilibration algorithms well known, we use equilibration routines from the Linear Algebra PACKage (LAPACK) to scale matrices

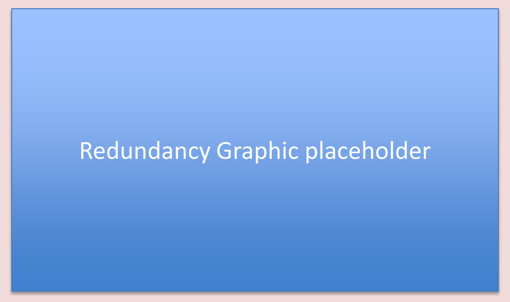
Acknowledgments

This research was supported by the Consortium for Advanced Simulation of Light Water Reactors (http://www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725.

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.
SAND No. 2013-6136P

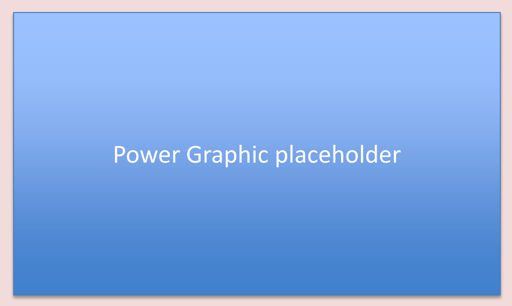
Motivation

- Coping with faulty hardware
  - Redundancy – allows detection and correction
  - Robustness – absorb faults



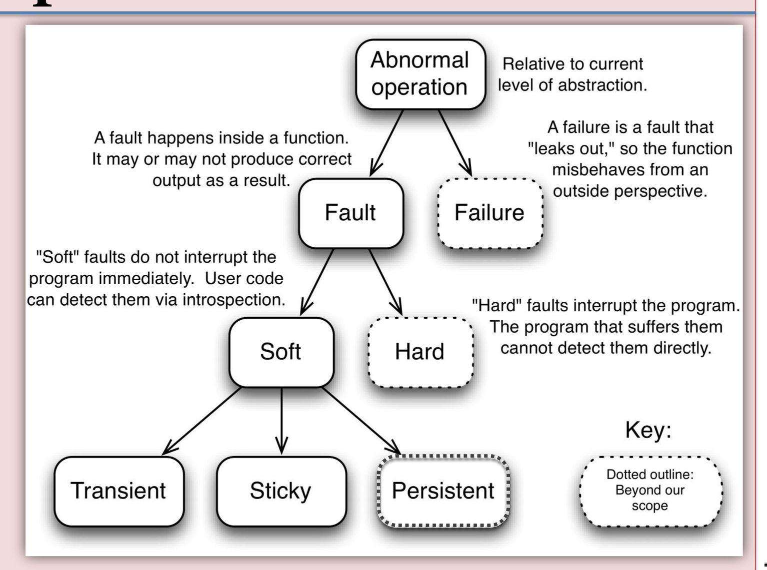
Real hardware combines both, with one of two goals:
- Lower performance, more robust
  - Exotic manufacturing (behind industry performance)
  - Hazardous environments (like outer space)
- Higher performance, higher power
  - Individual components may be less reliable
  - Choice of high-performance computing

Redundancy costs power
- Redundant storage & computation (checksums)
  - ECC (error-correcting) memory
  - RAID (Redundant Arrays of Inexpensive Disks)
  - Software checksums (Algorithm-Based Fault Tolerance)
- Redundant hardware & communication (voting)
  - MPI process replication (in software)
  - Redundant arithmetic units (in hardware) Run it several times ("time replication")



Scope

Redundancy costs power
- "Fault," "error," "failure," etc. don't have standard definitions
- We define "fault" as "something went wrong inside," and "failure" as "a fault that leaks outside"
- "Inside" / "outside" is relative to your perspective; for us, it means "the linear solver"



Fault Model

- ***Bit flips only impact the solution if the perturbed quantity is used***
  - Model a bit flip as a perturbed input to primitive arithmetic operations, compose higher-order operations
- ***What bits to flip?***
  - Arbitrary bit flips provide little insight → specific bits produce unique error characteristics.
  - Analyze specific bit ranges based on error characteristic
    - Exponent flips can introduce error orders of magnitude larger than the input
    - Mantissa flips introduce error with same magnitude
    - Sign flips introduce error equivalent to flipping the least significant exponent bit

Bit Location	Absolute Error: $\lvert\lambda - \hat{\lambda}\rvert$	
Mantissa	$(1 + 2^{j-52})\lambda_{mag.}$,	for $j = 0, \ldots, 51$
$\text{Exponent}_{1\to0}$	$2^{-2^j} \times \lambda$,	for $j = 0, \ldots, 10$; and $bit_{j+52} = 1$
$\text{Exponent}_{0\to1}$	$2^{2^j} \times \lambda$,	for $j = 0, \ldots, 10$; and $bit_{j+52} = 0$

Table 1: Bit flip absolute error for a perturbed scalar λ where $\lambda = \lambda_{magnitude} \times \lambda_{fraction} < \lambda_{magnitude} \times 2$

- ***Possible Numerical Perturbations***
  - *All* real-valued vectors translate to a finite set of IEEE-754 biases

$$\mathbf{a} \in \mathbb{R}^n.$$
$$\mathbf{a} = \begin{Bmatrix} 0.5 \\ 8.1 \\ 4 \end{Bmatrix} \Rightarrow \begin{Bmatrix} 2^{-1} \times 1.0 \\ 2^{+3} \times 1.m \\ 2^{+2} \times 1.m \end{Bmatrix} < \begin{Bmatrix} 1022+1 \\ 1026+1 \\ 1025+1 \end{Bmatrix} \Rightarrow \begin{Bmatrix} 01111111111 \\ 10000000011 \\ 10000000010 \end{Bmatrix}$$

A bit flip in floating point data creates a predictable numerical perturbation

Algorithm 1: GMRES algorithm

```
for l = 1 to  do
    r := b − Ax^(j−1)
    q_1 := r/||r||_2
    for j = 1 to restart do
        w_0 := Aq_j
        for i = 1 to j do
            h_{i,j} := (q_i, w_{i−1})
            w_i := w_{i−1} − h_{i,j} q_i
        end
        h_{j+1,j} := ||w||_2
        q_{j+1} := w/h_{j+1,j}
        Find y = min||H_j y − ||b|| e_1||_2
        Evaluate convergence criteria
        Optionally, compute x_j = Q_j y
    end
end
```

GMRES

Theoretical Bounds on the Arnoldi Process

$$\|\mathbf{w}_0\| = \|\mathbf{A}\mathbf{q}_j\| \le \|\mathbf{A}\|\,\|\mathbf{q}_j\|$$
$$\|\mathbf{w}_i\| = \|\mathbf{w}_{i-1} - h_{i,j}\mathbf{q}_i\| \le \|\mathbf{w}_{i-1}\| + |h_{i,j}|\,\|\mathbf{q}_i\|$$

With respect to the ℓ_2 norm,

$$\|\mathbf{w}_0\|_2 \le \|\mathbf{A}\|_2\,\|\mathbf{q}_j\|_2 \le \max|\sigma_k|$$
$$|h_{i,j}| \le \|\mathbf{w}_{i-1}\|_2$$
$$\|\mathbf{w}_i\|_2 \le 2\,\|\mathbf{w}_{i-1}\|_2 \le 2^i\,\|\mathbf{w}_0\|_2$$

Two norm

Requires determining the largest singular value of A, but allows detection of large perturbations.

Combining Magnitude Scaling with Norm Bounds

Count of possible bit flip perturbations in exponents for all Arnoldi iterations

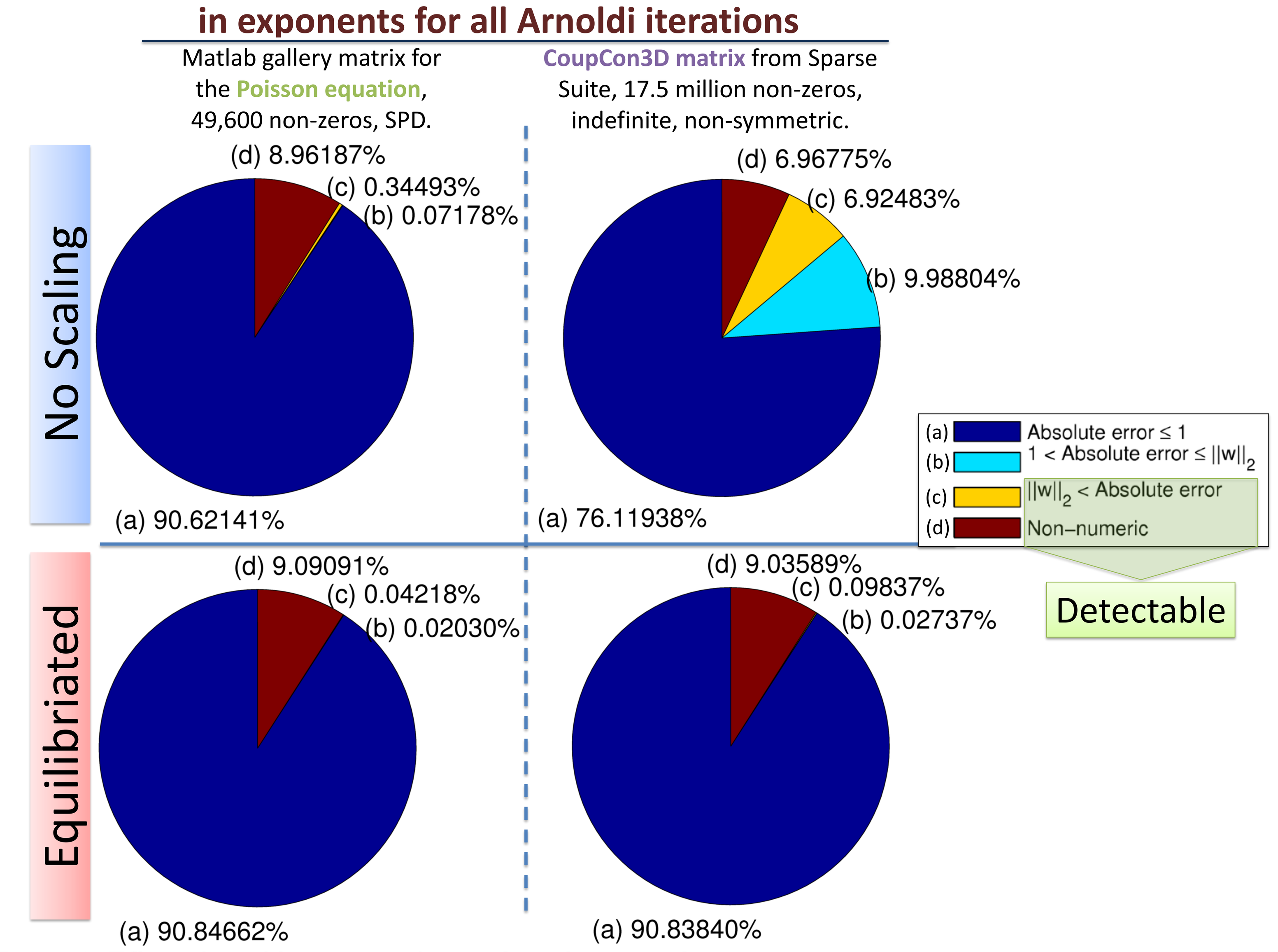


Figure 4. Effect of equilibration on the number of magnitude changing bit flips that may occur inside the Arnoldi process.

- Basis vectors (q_i) in GMRES are unit vectors (2-norm is 1).
- Bounds based solely on spectral norm of input matrix.
- If matrix is equilibrated, dot products consist primarily of values less than 1.
- Mantissa bit flips in numbers less than 1, produce small perturbations
- Majority of exponent flips produce error less than 1.

Interpretation: Chart sorts *possible* errors, does not convey frequency of error.
Intent: Analysis tool, understand relationship between inputs and what is theoretically possible in algorithm.
Fault Resilience: *Design algorithms to tolerate what is possible and detect what is impossible.*
Current Status: Can detect impossible values by checking Hessenberg entries against norm bound.
Inexact Krylov can allow convergence given bounded error
Sandbox model ensures algorithm does not produce a silently incorrect answer

Previous Work[1]

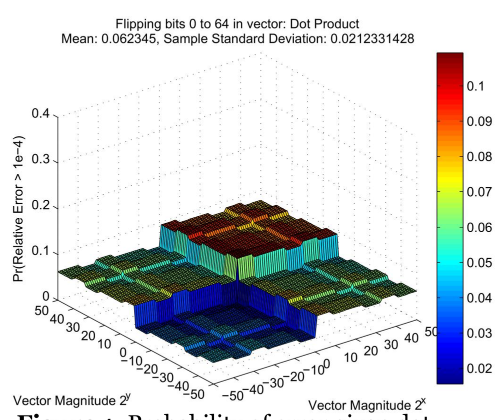


Figure 1. Probability of error given dot products of specific magnitudes

Scaling mitigates faults in dot products

(1) Exploit binary representation of bias to minimize impact of exponent bit flips
(2) Optimal bias is 1023 (2^0), less than 2^0 better than 2^1
(3) To manipulate biases, scale (equilibrate) the inputs

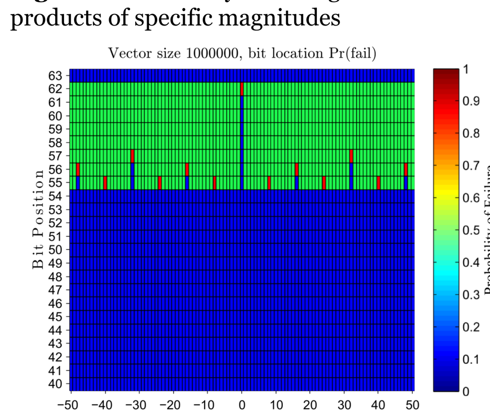


Figure 2. Slice showing per-bit probability of large error from Figure 1, when large numbers multiply small numbers.

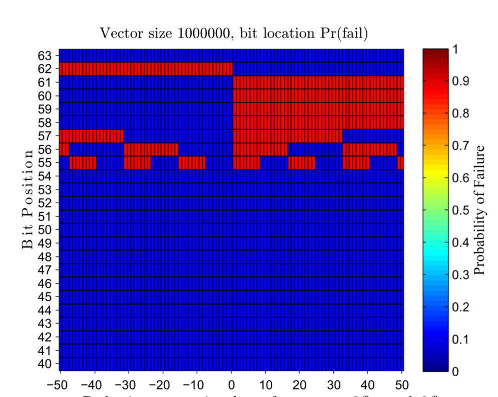
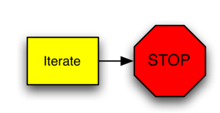


Figure 3. Slice showing per-bit probability of large error from Figure 1 , when numbers have similar magnitudes.

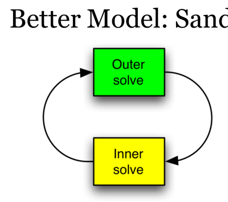
Reliability Models

Current Model: Fail-stop



- System tries to detect all soft faults
- Turn all detected soft faults into hard faults
- Detected local faults become global
- Checkpoint / restart is the
- only recovery model

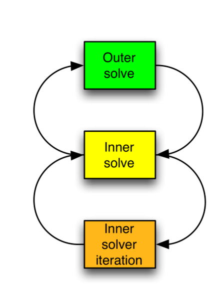
Better Model: Sandbox



- Isolate unreliable data & computation in a box
- Reliable code invokes box
- Local faults stay local
- App gets flexibility to define recovery model

Potential Model: Nested Sandbox

- Hessenberg monitoring takes place inside inner solve
- Use theoretical bounds to allow early rejection of faulty inner solves
- Research additional bounds to allow more efficient detection



Inexact Krylov

Application of Inexact Krylov to Inner Solves

- Errors in matrix or preconditioner
  - Bounded errors have bounded effects on (residual) error
  - Effect is inversely proportional to last iteration's residual error
- Errors in Gram-Schmidt process
  - Worst case when perturbation occurs before scaling
  - Have the following form

$$\mathbf{w}_j = (\mathbf{I} - \mathbf{Q}_j\mathbf{Q}_j^*)(\mathbf{A} + \hat{\mathbf{E}}_j)\mathbf{q}_j + \mathbf{E}_j\mathbf{q}_j$$

  - Where $\hat{\mathbf{E}}_j = \mathbf{Q}_j\mathbf{Q}_j^*\mathbf{E}_j$, and $\|\hat{\mathbf{E}}_j\| \le \|\mathbf{E}_j\|$.
  - Since the error is bounded, the convergence criteria from inexact Krlyov are applicable.

Fault Tolerant GMRES[2]

- Use Sandbox model
  - GMRES as inner solve
  - Reliability compute residual
- Faults injected as perturbations to matrix

Future Work

- Integrate inner solver checks
- Evaluate suitability of Inexact Krylov bounds checking in the inner solve

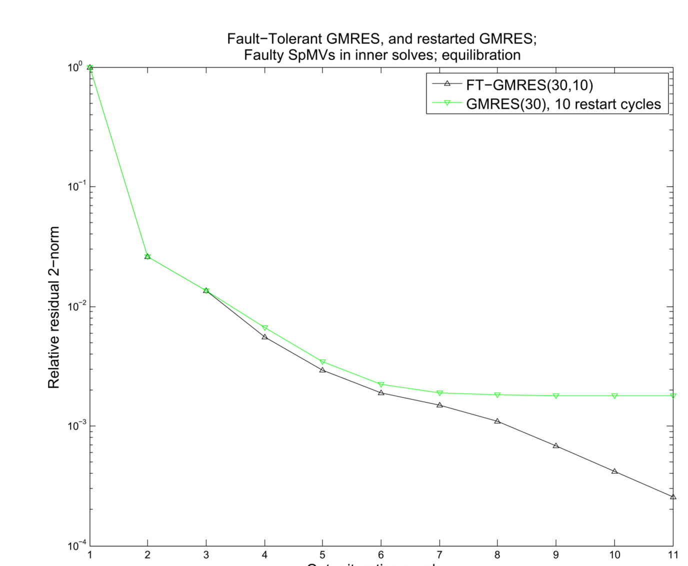


Figure 5. FT-GMRES and restarted GMRES solving CoupCons3D matrix after equilibration, using basic sandbox reliability model.

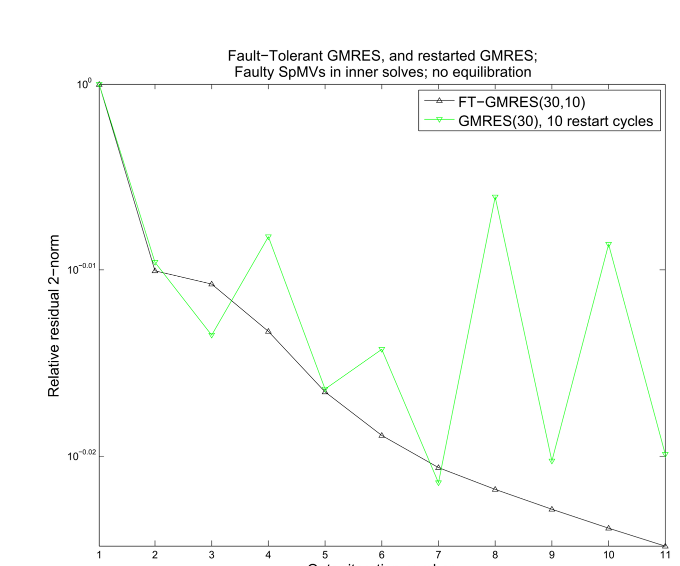


Figure 6. FT-GMRES and restarted GMRES solving CoupCons3D matrix with no equilibration, using basic sandbox reliability model.

References

1. "Quantifying the Impact of Single Bit Flips on Floating Point Arithmetic" by J. Elliott, F. Mueller, M. Stoyanov, C. Webster", invited talk at SIAM Conference on Computational Science and Engineering, Feb 2013, see TR 2013-2, Dept. of Computer Science, North Carolina State University, Mar 2013.
2. P. G. Bridges, K. B. Ferreira, M. A. Heroux, and M. Hoemmen. Fault-tolerant linear solvers via selective reliability. ArXiv e-prints, June 2012.