On Numerical Resiliency in Numerical Linear Algebra Solvers

Luc Giraud

joint work with
E. Agullo (Inria), P. Salas (Sherbrooke Univ.),
F. Yetkin (Inria) and M. Zounon (Inria)

HiePACS - Inria Project
Inria Bordeaux Sud-Ouest
Outline

1. Hard fault: Interpolation-Restart strategies
   - In Krylov subspace linear solvers
   - In revisited eigensolvers

2. Soft errors in Conjugate Gradient (preliminary)
Outline

1. Hard fault: Interpolation-Restart strategies
   - In Krylov subspace linear solvers
   - In revisited eigensolvers

2. Soft errors in Conjugate Gradient (preliminary)
Objective: Design resilient algorithms for solving sparse linear systems

Two classes of iterative methods

- Stationary methods (Jacobi, Gauss-Seidel, . . .)
- Krylov subspace methods (CG, GMRES, Bi-CGStab, . . .)

Krylov methods are widely used but efforts are required to go for extreme-scale
We distinguish two categories of data:

- **Static data**
- **Dynamic data**
We distinguish two categories of data:

- **Static data**
- **Dynamic data**

What will happen when $P_1$ crashes?
We distinguish two categories of data:

- **Static data**
- **Dynamic data**

What will happen when $P_1$ crashes?

Failed process is replaced

Static data are restored
We distinguish two categories of data:

- **Static data**
- **Dynamic data**

What will happen when $P_1$ crashes?

Failed process is replaced
Static data are restored

**Reset:** set $x_1$ to the initial value and restart
**IR:** interpolate $x_1$ and restart
We distinguish two categories of data:

- **Static data**
- **Dynamic data**

What will happen when $P_1$ crashes?

Failed process is replaced
Static data are restored

Reset: set $x_1$ to the initial value and restart
IR: interpolate $x_1$ and restart
General assumptions

1. Large dimensional vectors and matrices are distributed

2. Scalars and low dimensional vectors and matrices are replicated

3. There is a system mechanism to report faults

4. Faulty process is replaced

5. Static data are restored
Interpolation methods

Fault in linear system

\[
\begin{pmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2
\end{pmatrix}
=
\begin{pmatrix}
b_1 \\
b_2
\end{pmatrix}
\]

Linear Interpolation (LI) [J. Langou et al., 2007]
Solve \( A_{11}x_1 = b_1 - A_{12}x_2 \) \( A_{11} \) must be non singular

Least Squares Interpolation (LSI)

(To be added)
In Krylov subspace linear solvers

Interpolation methods

Fault in linear system

\[
\begin{pmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{pmatrix}
\begin{pmatrix}
? \\
x_2
\end{pmatrix}
= 
\begin{pmatrix}
b_1 \\
b_2
\end{pmatrix}
\]

How to interpolate \( x_1 \)?

Linear Interpolation (LI) [J. Langou et al., 2007]

Solve

\[
A_{11}x_1 = b_1 - A_{12}x_2
\]

\( A_{11} \) must be non singular

Least Squares Interpolation (LSI)

\[
\begin{pmatrix}
A_{11} \\
A_{21}
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2
\end{pmatrix}
+ 
\begin{pmatrix}
A_{21} \\
A_{22}
\end{pmatrix}
\begin{pmatrix}
x_2 \\
x_1
\end{pmatrix}
= 
\begin{pmatrix}
b_1 \\
b_2
\end{pmatrix}
\]

\( x_1 = \arg \min_x \left\| \begin{pmatrix}
b_1 \\
b_2
\end{pmatrix} - \begin{pmatrix}
A_{12} \\
A_{22}
\end{pmatrix} x_2 - \begin{pmatrix}
A_{11} \\
A_{21}
\end{pmatrix} x_1 \right\|_2 \)
Interpolation methods

Fault in linear system

\[
\begin{pmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{pmatrix}
\begin{pmatrix}
? \\
x_2
\end{pmatrix}
=
\begin{pmatrix}
b_1 \\
b_2
\end{pmatrix}
\]

How to interpolate \(x_1\)?

Linear Interpolation (LI) [J. Langou et al., 2007]

Solve \(A_{11}x_1 = b_1 - A_{12}x_2\)

\(A_{11}\) must be non singular

Least Squares Interpolation (LSI)

\[
\begin{pmatrix}
A_{11} \\
A_{21}
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2
\end{pmatrix}
=
\begin{pmatrix}
b_1 \\
b_2
\end{pmatrix}
\]

\[x_1 = \arg\min_x \left\| \begin{pmatrix}
b_1 \\
b_2
\end{pmatrix} - \begin{pmatrix}
A_{12} \\
A_{22}
\end{pmatrix} x_2 - \begin{pmatrix}
A_{11} \\
A_{21}
\end{pmatrix} x_1 \right\|_2\]
**Interpolation methods**

Fault in linear system

\[
\begin{pmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{pmatrix}
\begin{pmatrix}
? \\
x_2
\end{pmatrix} =
\begin{pmatrix}
b_1 \\
b_2
\end{pmatrix}
\]

How to interpolate \(x_1\)?

**Linear Interpolation (LI)** [J. Langou et al., 2007]

Solve \(A_{11}x_1 = b_1 - A_{12}x_2\) \(A_{11}\) must be non singular

**Least Squares Interpolation (LSI)**

\[
\begin{pmatrix}
A_{11} \\
A_{21}
\end{pmatrix}x_1 +
\begin{pmatrix}
A_{21} \\
A_{22}
\end{pmatrix}x_2 =
\begin{pmatrix}
b_1 \\
b_2
\end{pmatrix}
\]

\[
x_1 = \arg\min_x \left\| \begin{pmatrix}
b_1 \\
b_2
\end{pmatrix} - \begin{pmatrix}
A_{12} \\
A_{22}
\end{pmatrix}x_2 - \begin{pmatrix}
A_{11} \\
A_{21}
\end{pmatrix}x \right\|_2
\]
Main properties of Interpolation-Restart strategies

**LI property**
The LI strategy preserves the monotone decrease of the A-norm of the forward error of CG or PCG.

**LSI property**
The LSI strategy preserves the monotone decrease of the residual norm of minimal residual Krylov subspace methods such as GMRES and MinRES.
Main properties of Interpolation-Restart strategies

**LI property**
The LI strategy preserves the monotone decrease of the A-norm of the forward error of CG or PCG.

**LSI property**
The LSI strategy preserves the monotone decrease of the residual norm of minimal residual Krylov subspace methods such as GMRES and MinRES.
Impact of lost data volume
Impact of lost data volume

\[ \frac{\|b-Ax\|}{\|b\|} \]

\[ \text{Iteration} \]

\[ \text{NF} \]

GMRES(100) - Averous/epb3
Hard fault: Interpolation-Restart strategies in Krylov subspace linear solvers

Impact of lost data volume

\[ \frac{\|b-Ax\|}{\|b\|} \] vs. iteration for GMRES(100) - Averous/epb3 - 10 faults - **0.001% data loss**
Impact of lost data volume

GMRES(100) - Averous/epb3 - 10 faults - 0.001% data loss
Impact of lost data volume

GMRES(100) - Averous/epb3 - 10 faults - 0.001% data loss
Impact of lost data volume

GMRES(100) - Averous/epb3 - 10 faults - 0.001% data loss
Impact of lost data volume

\[ \frac{\|b-Ax\|}{\|b\|} \]

Iteration

Reset
LI
LSI
ER

GMRES(100) - Averous/epb3 - 10 faults - 0.2% data loss
Impact of lost data volume

\[
\frac{\|b-Ax\|}{\|b\|} \quad \text{Iteration}
\]

- Reset
- LI
- LSI
- ER

GMRES(100) - Averous/epb3 - 10 faults - 0.8% data loss
Impact of lost data volume

GMRES(100) - Averous/epb3 - 10 faults - 3% data loss
Experimental results

Fault rate impact
Impact of fault rate

GMRES(100) - Kim - **4 faults** - 0.2% data loss
Impact of fault rate

GMRES(100) - Kim - 8 faults - 0.2% data loss
Impact of fault rate

GMRES(100) - Kim - 17 faults - 0.2% data loss
Impact of fault rate

GMRES(100) - Kim - **40 faults** - 0.2% data loss
Penalty of restart strategy on PCG

PCG (Cunningham/qa8fm - 9 faults)
Penalty of restart strategy on PCG

PCG (Cunningham/qa8fm - 9 faults)
Penalty of restart strategy on GMRES

Preconditioned GMRES(100) (Kim1 - 8 faults)
Penalty of restart strategy on GMRES

Preconditioned GMRES(100) (Kim1 - 8 faults)
Summary on resilient Krylov linear solvers

Interpolation-Restart strategies key features:

- Make sense: key properties of CG and GMRES are preserved
- The restarting effect remains reasonable within the restarted GMRES context
- Robust even with a high fault rate and a high volume of lost data
- No fault implies no overhead
- Extended to multiple faults
Hard fault: Interpolation-Restart strategies in revisited eigensolvers

$L. \text{Giraud - On Numerical Resilience in Linear Algebra}$

\[ Au = \lambda u \text{ with } u \neq 0, \text{ where } A \in \mathbb{C}^{n \times n}, u \in \mathbb{C}^n, \text{ and } \lambda \in \mathbb{C} \]

- $\lambda$ : eigenvalue
- $u$ : eigenvector
- $(\lambda, u)$ : eigenpair

Two classes of methods

- **Fixed Point Methods** (Power Method, Subspace iteration)
- **Subspace Methods** (Jacobi-Davidson, Arnoldi, IRA/Krylov Schur)
Interpolation strategies

Fault in eigenproblem

\[
\begin{pmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{pmatrix}
\begin{pmatrix}
\mathbf{u}_1 \\
\mathbf{u}_2
\end{pmatrix} = \lambda
\begin{pmatrix}
\mathbf{u}_1 \\
\mathbf{u}_2
\end{pmatrix}
\]

Linear Interpolation (LI)

\[(A_{11} - \lambda I_1) \mathbf{u}_1 = -A_{12} \mathbf{u}_2\]

Least Squares Interpolation (LSI)

\[
\begin{pmatrix}
A_{11} \\
A_{21}
\end{pmatrix} \mathbf{x}_1 + \begin{pmatrix}
A_{21} \\
A_{22}
\end{pmatrix} \mathbf{u}_2 = \lambda
\begin{pmatrix}
\mathbf{u}_1 \\
\mathbf{u}_2
\end{pmatrix}
\]

\[
\mathbf{u}_1 = \arg\min_u \left\| \begin{pmatrix}
A_{11} - \lambda I_1 \\
A_{21}
\end{pmatrix} \mathbf{u} + \begin{pmatrix}
A_{12} \\
A_{22} - \lambda I_2
\end{pmatrix} \mathbf{u}_2 \right\|_2
\]
Interpolation strategies

Fault in eigenproblem

\[
\begin{pmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{pmatrix}
\begin{pmatrix}
? \\
u_2
\end{pmatrix}
= \lambda
\begin{pmatrix}
? \\
u_2
\end{pmatrix}
\]

\(u_1\) entries are lost
\(\lambda\) is replicated

Linear Interpolation (LI)

\[
(A_{11} - \lambda I_1) u_1 = -A_{12} u_2
\]

Least Squares Interpolation (LSI)

\[
\begin{pmatrix}
A_{11} \\
A_{21}
\end{pmatrix} x_1 + \begin{pmatrix}
A_{21} \\
A_{22}
\end{pmatrix} u_2 = \lambda \begin{pmatrix}
u_1 \\
u_2
\end{pmatrix}
\]

\[
u_1 = \arg \min_u \left\| \begin{pmatrix}
A_{11} - \lambda I_1 \\
A_{21}
\end{pmatrix} u + \begin{pmatrix}
A_{12} \\
A_{22} - \lambda I_2
\end{pmatrix} u_2 \right\|_2
\]
Eigensolvers revisited

- Basic subspace iteration method
- Subspace iteration with polynomial acceleration
- Arnoldi method
- Implicitly Restarted Arnoldi method
- Jacobi-Davidson method (JDQR)
Eigensolvers revisited

- Basic subspace iteration method
- Subspace iteration with polynomial acceleration
- Arnoldi method
- Implicitly Restarted Arnoldi method
- Jacobi-Davidson method (JDQR)
Jacobi-Davidson at a glance

**JDQR algorithm**

- Compute a few **nev** eigenpairs associated with eigenvalues closed to a target $\tau$
- Perform a partial Schur decomposition $AZ_{nconv} = Z_{nconv} T_{nconv}$
- The next Schur vectors are searched in the $V_k$ search space
In revisited eigensolvers

**Resilient Jacobi-Davidson algorithm**

**Converged Schur vectors**
- \( AZ_{n\text{conv}} = Z_{n\text{conv}} T_{n\text{conv}} \)
- Interpolate the \( n\text{conv} \) converged vectors

**Search space**
- \( C_k = V_k^H A V_k \)
- Interpolate a few \( s \) best candidates

JD is restarted with a search space of \( n\text{conv} + s \) vectors

Flexibility to select the dimension of the search space for restarting
Experimental framework

Thermoacoustic instabilities in combustion chambers [Image courtesy of CERFACS]

Damaged rocket chamber from combustion instabilities
Experimental framework

Thermoacoustic instabilities in combustion chambers

Simulation of combustion instabilities in annular chambers
Experimental framework

Thermoacoustic instabilities in combustion chambers

Compute eigenpairs associated with smallest magnitude eigenvalues laying in the periphery of the spectrum
Experimental results

Convergence of 5 eigenpairs associated with smallest magnitude eigenvalues
Experimental results

Restart with \textbf{nconv vectors} from converged Schur vectors combined with \textbf{5 best candidates} from the search space (6 faults)
Experimental results

Restart with **nconv vectors** from converged Schur vectors combined with **5 best candidates** from the search space (6 faults).
Impact of **keeping the best Schur vector candidate** in the search space after a fault (6 faults)
Summary on resilient eigensolvers

- Specific Interpolation-Restart strategies for each eigensolver
  - Basic subspace iteration method
  - Subspace iteration with polynomial acceleration
  - Arnoldi method
  - IRAM
  - Jacobi-Davidson

- More flexibility in eigensolver than in Krylov linear solvers
Outline

1. Hard fault: Interpolation-Restart strategies
   - In Krylov subspace linear solvers
   - In revisited eigensolvers

2. Soft errors in Conjugate Gradient (preliminary)
Aim of this study

Context

Soft Errors in Conjugate Gradient (CG)

Question 1
Impact of soft errors on convergence of CG

Question 2
Reliability of numerical detection mechanisms?

Question 3
Robust numerical recovery schemes?

Method to address question 1
Experimental study with basic statistics
Aim of this study

Context
Soft Errors in Conjugate Gradient (CG)

Question 1
Impact of soft errors on convergence of CG

Question 2
Reliability of numerical detection mechanisms?

Question 3
Robust numerical recovery schemes?

Method to address question 1
Experimental study with basic statistics
Aim of this study

**Context**
Soft Errors in Conjugate Gradient (CG)

**Question 1**
Impact of soft errors on convergence of CG

**Question 2**
Reliability of numerical detection mechanisms?

**Question 3**
Robust numerical recovery schemes?

**Method to address question 1**
Experimental study with basic statistics
Aim of this study

Context
Soft Errors in Conjugate Gradient (CG)

Question 1
Impact of soft errors on convergence of CG

Question 2
Reliability of numerical detection mechanisms?

Question 3
Robust numerical recovery schemes?

Method to address question 1
Experimental study with basic statistics
Aim of this study

Context
Soft Errors in Conjugate Gradient (CG)

Question 1
Impact of soft errors on convergence of CG

Question 2
Reliability of numerical detection mechanisms?

Question 3
Robust numerical recovery schemes?

Method to address question 1
Experimental study with basic statistics
Algorithm 1 Preconditioned CG

1: $r_0 := b - Ax_0$; $u_0 := M^{-1}r_0$; $p_0 := u_0$
2: for $i = 0, \ldots$ do
3: $s := Ap_i$
4: $\alpha := (r_i, u_i)/(s, p_i)$
5: $x_{i+1} := x_i + \alpha p_i$
6: $r_{i+1} := r_i - \alpha s$
7: $u_{i+1} := M^{-1}r_{i+1}$
8: $\beta := (r_{i+1}, u_{i+1})/(r_i, u_i)$
9: $p_{i+1} := u_{i+1} + \beta p_i$
10: end for
Algorithm 1 Preconditioned CG

1: \( r_0 := b - Ax_0; u_0 := M^{-1}r_0; p_0 := u_0 \)
2: for \( i = 0, \ldots \) do
3: \( s := Ap_i \)
4: \( \alpha := (r_i, u_i)/(s, p_i) \)
5: \( x_{i+1} := x_i + \alpha p_i \)
6: \( r_{i+1} := r_i - \alpha s \)
7: \( u_{i+1} := M^{-1}r_{i+1} \)
8: \( \beta := (r_{i+1}, u_{i+1})/(r_i, u_i) \)
9: \( p_{i+1} := u_{i+1} + \beta p_i \)
10: end for
Fault injection

![Graph showing relative error over iterations for SpMV operation](image)
Fault injection
Fault injection

![Graph showing relative error versus iterations for SpMV operation]

- Relative Error on the y-axis
- Iterations on the x-axis
- Graph indicates decreasing error with increasing iterations
- SpMV operation highlighted with a box marked 'A'
- Fault injection indicated by arrow from 's' to 'p'
Fault injection

![Graph showing relative error over iterations]

- 10% and 90% indicated on the graph.
Fault injection

![Graph showing relative error vs. iterations with labels 10%, 90%, and SpMV]

L. Giraud - On Numerical Resilience in Linear Algebra

26 / 31
Fault injection

AT ANY TIME

Relative Error

0 200 400 600 800 1000

10^{-15} 10^{-10} 10^{-5} 10^{0}

10% 90%

SpMV
Transient fault injection

AT ANY TIME

AT ANY LOCATION

SpMV

Relative Error

0 200 400 600 800 1000

Iterations

10^{-15} 10^{-10} 10^{-5} 10^{0}

10% 90%
Transient fault injection

WITH ANY MAGNITUDE

\[ p_F = P_{NF}(1 + 2^k) \]

AT ANY TIME

AT ANY LOCATION

Relative Error

\[ 10^{-15} \]
\[ 10^{-10} \]
\[ 10^{-5} \]
\[ 10^{0} \]

Iterations

\[ 0 \]
\[ 200 \]
\[ 400 \]
\[ 600 \]
\[ 800 \]
\[ 1000 \]

10% 90%
## Parameters for Fault Injection

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type ID</td>
<td>Type of the Conjugate Gradient Method</td>
<td>Classical, Chrono, Pipelined</td>
</tr>
<tr>
<td>Matrix ID</td>
<td>Matrix ID</td>
<td>[1 : 30]</td>
</tr>
<tr>
<td>Time k</td>
<td>Fault injection iteration as percentage</td>
<td>[10%,...,90%]</td>
</tr>
<tr>
<td></td>
<td>Magnitude of fault injection</td>
<td>[-16:2:16]</td>
</tr>
<tr>
<td>Location</td>
<td>Location of fault injection</td>
<td>50 random</td>
</tr>
</tbody>
</table>

[A. T. Chronopoulos, C. W. Gear - JCAM, 1989]

[P. Ghysels, W. Vanroose - ParCo, 2014]
Protocol: correct convergence

<table>
<thead>
<tr>
<th>Effect on the convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converged</td>
</tr>
<tr>
<td>Not Converged</td>
</tr>
</tbody>
</table>

![Graph](image)

Relative Error vs. Iterations
Protocol: correct convergence

<table>
<thead>
<tr>
<th>Effect on the convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converged</td>
</tr>
<tr>
<td>Not Converged</td>
</tr>
</tbody>
</table>

![Graph showing relative error vs iterations with Acc marker]
Protocol: correct convergence

<table>
<thead>
<tr>
<th>Effect on the convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converged</td>
</tr>
<tr>
<td>Not Converged</td>
</tr>
</tbody>
</table>

![Graph showing relative error versus iterations with two curves: one representing converged and the other not converged. The y-axis is labeled as "Relative Error" ranging from $10^{-15}$ to $10^0$, and the x-axis is labeled as "Iterations" ranging from 0 to 1000. The graph includes an annotation "Acc".](image-url)
Protocol: correct convergence

Effect on the convergence

<table>
<thead>
<tr>
<th>Converged</th>
<th>Not Converged</th>
</tr>
</thead>
</table>

![Graph showing convergence](image)

Acc
Protocol: correct convergence

<table>
<thead>
<tr>
<th></th>
<th>Effect on the convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converged</td>
<td></td>
</tr>
<tr>
<td>Not Converged</td>
<td></td>
</tr>
</tbody>
</table>
Protocol: correct convergence

<table>
<thead>
<tr>
<th>Effect on the convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converged</td>
</tr>
<tr>
<td>Not Converged</td>
</tr>
</tbody>
</table>

![Graph showing iterations vs. relative error](image)

Acc
Protocol: correct convergence

<table>
<thead>
<tr>
<th>Effect on the convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converged</td>
</tr>
<tr>
<td>Not Converged</td>
</tr>
</tbody>
</table>

![Diagram showing relative error over iterations with a table indicating the effect on convergence.]
Protocol: correct convergence

<table>
<thead>
<tr>
<th>Effect on the convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converged</td>
</tr>
<tr>
<td>Not Converged</td>
</tr>
</tbody>
</table>

![Graph showing relative error against iterations with a legend for converged and not converged cases.](chart.png)
Protocol: correct convergence

<table>
<thead>
<tr>
<th>Effect on the convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converged</td>
</tr>
<tr>
<td>Not Converged</td>
</tr>
</tbody>
</table>

![Graph showing convergence with annotations](image)
Characterize effect of magnitude \((k)\)

- Magnitude \((k)\)
- Acceptable Cases
- Variant of CG
  - Classical
  - Chrono/Gear
  - Pipeline
Characterize effect of fault injection time

![Graph showing the effect of fault injection time on acceptable cases for different variants of CG algorithms.](image)

- **Acceptable Cases**
- **Fault Injection Time**
- **Variant of CG**
  - Classical
  - Chrono/Gear
  - Pipeline

Legend:
- **OUTLIER** Less than 3/2 times of lower quartile
- **OUTLIER** More than 3/2 times of upper quartile
- **MAXIMUM** Greatest value, excluding outliers
- **UPPER QUARTILE** 25% of data greater than this value
- **MEDIAN** 50% of data is greater than this value; middle of dataset
- **LOWER QUARTILE** 25% of data less than this value
- **MINIMUM** Least value, excluding outliers
Qualitative observations

- Classical CG and Chronopolus/Gear CG are less sensitive for soft errors than Pipelined CG.

Future Work

- Extend the study to GMRES
- Detection based on checksum mechanisms to protect the sparse matrix-vector product
- Detection mechanisms based on round-off error analysis (additional benefit: insight on mixed precision calculation)
Acknowledgement for financial support:

- French ANR: RESCUE project
- European FP7: Exa2CT project
- G8: ECS project

http://hiepacs.bordeaux.inria.fr/
Merci for your attention

Questions ?

Acknowledgement for financial support:

- French ANR: RESCUE project
- European FP7: Exa2CT project
- G8: ECS project

http://hiepacs.bordeaux.inria.fr/