Consider the generalized large scale linear differential equation that is steady state

$$-AX = F,$$

or time dependent

$$\frac{d}{dt}X = AX + F.$$

The differential operator $A$ can represent complex dynamics of multiple coupled phenomena (e.g., multi-physics).

Finding the solution $X$ requires a powerful supercomputer.
Supercomputing Speed vs Reliability

Hardware Failure
- Operation at near critical voltage, ever shrinking fabrication size and increasing number of components
- Increased computational throughput leads to decrease in reliability
- Soft faults are already causing problems in largest scale computations (e.g., climate)
- In near future we expect the mean failure time to be of the order of an application runtime

Hardware Error Control
- DRAM has build-in error detection and correction
- Network communication has CRC checksum
- CPU cache and logical circuits have virtually no protection
Results of Faults

- The code crashes and no result is given
- The code completes execution, but returns an invalid result (e.g., \( \text{inf} \) or \( \text{nan} \))
- The code completes execution and returns a result that “appears” valid

\[
2 + 3 = 7
\]

Software Error Control

- Check-pointing: how to detect silent faults?
- Check-summing: in heterogeneous computing environment, we cannot check-sum floating point operations.
- Post-processing: detection is too late.
- Redundant computations: doable on at limited scale.
The Cutting Edge Approach


- Code is split into two modes with different reliability
  - **Fast Mode**: The bulk of the computations should be performed in this mode and any “fast” computation is susceptible to hardware faults.
  - **Safe Mode**: Only the bare minimum of work will be done in this mode, computations are assumed to be completely reliable.
- A sanity-check is performed in **Safe Mode** to guard against introduction of large hardware error
- Natural self-correcting properties are exploited to correct hardware error
Challenges of Selective Reliability

- Rigorously quantify the impact of soft faults
- Analyze the convergence rate
- Devise optimal accept/reject criteria
- What type and rate of faults can be corrected
Analytic Fault Model

- Algorithms are comprised of steps
- Steps compute intermediate answer $\rightarrow$ final solution
- Each step is performed in either fast or safe mode
- **Safe** mode always returns correct answer
- **Fast** mode is susceptible to faults and there is probability $p$ of computing something that is not correct (i.e., we should get $x$, but instead we get $x + \tilde{x}$)

- *If the intermediate result is corrupted, then we make no assumption on the distribution of the perturbation*
\[ \| \mathbf{x}_{\text{exact}} - \mathbf{x}_{\text{computed}} \| \leq E_{\text{rounding}} + E_{\text{discrete}} + E_{\text{iterative}} + E_{\text{hardware}} \]

- \( E_{\text{rounding}} \to 0 \) with more precision, i.e. single, double, quad.
- \( E_{\text{discrete}} \to 0 \) with more the nodes, i.e. denser mesh
- \( E_{\text{iterative}} \to 0 \) with more solver iterations
- In all cases, more work means better approximation
- \( E_{\text{hardware}} \to 0 \), as we do more work

**Definition (Convergence with Respect to Hardware Error)**

A method is convergent on a specific hardware, if the statistical moments of the Hardware Error converge to zero as the computational cost increases, regardless of the distribution of the perturbation.
Consider the steady state problem

\[ A \mathbf{x} = \mathcal{F}, \]

which we discretize into a system of linear equations

\[ A|_N \mathbf{x}|_N = b|_N, \]

where

\[ A|_N \rightarrow \mathcal{A}, \quad b|_N \rightarrow \mathcal{F}, \quad \mathbf{x}|_N \rightarrow \mathbf{x}. \]

We need a resilient linear solver!
Given the linear system of equations

\[ Ax = b, \]

iterative solvers general iterative solvers generate a sequence

\[ x_k \rightarrow x. \]

We perform \( K \) iterations so that tolerance \( \epsilon \) is reached

\[ \| b - Ax_K \| < \epsilon. \]

Krylov basis is expensive to store and manipulate, use restated scheme

\[ \{ \{ x_k^m \}_{k=0}^{K} \}_{m=1}^{M} \]
Due to M. Hoemmen and M. Heroux:

Given $A$, $b$, $x_0$ and $\epsilon > 0$

$e_0 = \|b - Ax_0\|$

$m = 0$

while $e^m > \epsilon$ do

Compute $x^{m+1}_K$

$e^{m+1} = \|b - Ax^{m+1}_K\|$

if $e^m > e^{m-1}$ then

discard $x^{m+1}_K$ and $h_{m+1}$ and redo the last inner loop

else

accept $x^{m+1}_K$ and advance to the next $m = m + 1$

end if

end while
Fault Free Convergence

The residual is \( e^m_k = \| b - Ax^m_k \| \) and let

\[
\Lambda^+ = \{ \lambda : Av = \lambda v, \Re(\lambda) > 0 \}
\]
\[
\Lambda^+ \subset \text{disk}(D, d/2)
\]
\[
\Lambda^- = \{ \lambda : Av = \lambda v, \Re(\lambda) \leq 0 \}
\]
\[
\nu = |\Lambda^-|
\]

Define magnification and reduction factors

\[
R = \text{cond}(A) \left( \frac{\max_{\lambda^+, \lambda^-} |\lambda^+ - \lambda^-|}{\min_{\lambda^-} |\lambda^-|} \right)^\nu \left( \frac{D}{d} \right)^\nu, \quad r = \frac{d}{D}.
\]

Then,

\[
e^K_M \leq (Rr^K)^M e_0 = R^M r^{M \times K} e_0.
\]
Let $M^*$ be so that

$$e^{M^*} \leq \epsilon, \quad M^* = \left\lceil \frac{\log \epsilon - \log e_0}{\log R + K \log r} \right\rceil$$

- Hardware faults do not increase the residual of FT-GMRES.
- The method will reach desired tolerance $\epsilon$ after $M^*$ fault free outer iterations.
- FT-GMRES convergence is equivalent to analyzing the probability of computing $M^*$ fault free iterations in $M$ tries.
Let $p$ indicate the probability of encountering failure in the supercomputer over a unit of time equal to a single inner loop iteration.

Assume fault rate is uncorrelated in time.

The probability that $K$ inner loop iterations encounter a failure

$$s(K) \approx (1 - p)^K$$

Then the probability of encountering $M^*$ fault free iterations in $M \geq M^*$ attempts is

$$P(M) = s(K) \left( \frac{M - 1}{M^* - 1} \right) s(K)^{M^* - 1} (1 - s(K))^{M - M^*}.$$
The expected number of iterations needed for convergence is

\[ \mathbb{E}[M] = \sum_{M=M^*}^{\infty} M \left( \frac{M - 1}{M^* - 1} \right) s(K)^{M^*} (1 - s(K))^{M-M^*} = \frac{M^*}{s(K)} \]

The variance is bounded by

\[ \mathbb{V}[M] = \sum_{M=M^*}^{\infty} (M - \mathbb{E}[M])^2 P(M) \leq \frac{M^*}{s(K)^2} \]

The probability of FT-GMRES converging, approaches 1, as \( M \to \infty \).
Let $W$ be the cost associated with computing the residual in safe mode (e.g., for triple redundancy, $W = 3$).

Then, the total cost is

$$ C = M(K + W) $$

and the expected cost is

$$ \mathbb{E}[C] = \frac{M^*}{s(K)}(K + W) = \left[ \frac{\log \epsilon - \log e_0}{\log R + K \log r} \right] \frac{K + W}{(1 - p)^K}. $$

There is an optimal restart rate that is the global minimizer of $\mathbb{E}[C]$ that is independent from $\epsilon$ and $e_0$. 
Consider boundary value problem

\[0.2x_{\xi\xi} + x_\xi = 0, \quad x(0) = 0, \quad x(1) = 1.\]

We discretize the problem using finite difference scheme resulting in the linear system of equations

\[
\begin{pmatrix}
(0.4 + \frac{1}{N}) & -0.2 - \frac{1}{N} & 0 & \cdots & 0 \\
-0.2 & (0.4 + \frac{1}{N}) & -0.2 - \frac{1}{N} & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & \cdots & -0.2 & (0.4 + \frac{1}{N}) & -0.2 - \frac{1}{N} \\
0 & 0 & \cdots & -0.2 & (0.4 + \frac{1}{N})
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_{N-2} \\
x_{N-1}
\end{pmatrix}
= \begin{pmatrix}
0 \\
0 \\
\vdots \\
0 \\
0.2
\end{pmatrix}
\]
Figure: **Left**: Fault free GMRES convergence. **Right**: Faulty GMRES convergence (or the lack thereof). $N = 101$, $p = 0.05$, $K = 20$, $s(K) \approx 0.36$

$$\tilde{x} = \frac{g}{\|g\|} 10^u, \quad g \sim \mathcal{N}(0, 1)^N, \quad u \sim \mathcal{U}(-9, 19).$$
Figure: Two realization of FT-GMRES, despite the high fault rate, the algorithm converges reliably.
Figure: Expected cost of the FT-GMRES algorithm for different values of $K$. The expectation is computed using Monte Carlo sampling.
We have analytic proof of convergence for the FT-GMRES method.

The method converges for virtually any fault rate $p$.

The computational cost and convergence rate are strongly influenced by the fault rate $p$ and restart rate $K$.

When making a decision on the restart rate $K$, we need to consider the rate of faults $p$ in addition to eigenstructure of the matrix $A$ and storage limitations.
Time dependent PDE

\[ \frac{d}{dt} \mathcal{X} = A \mathcal{X} + \mathcal{F}. \]

which we discretize into a system of ordinary differential equations

\[ \frac{d}{dt} x|_N = A|_N x|_N + b|_N, \]

where

\[ A|_N \rightarrow A, \quad b|_N \rightarrow \mathcal{F}, \quad x|_N \rightarrow \mathcal{X}. \]

Given \( \mathcal{X}(0) \) we are looking for \( \mathcal{X}(T) \).

We need a resilient time integrator!
Consider first order integration scheme for the system of ODEs

\[ \frac{d}{dt} x = Ax. \]

Select time step \( \Delta t \) and approximate \( x^n \approx x(n\Delta t) \)

First order Euler time stepping schemes

\[ x^{n+1} = x^n + \Delta tAx^n, \quad x^{n+1} = x^n + \Delta tAx^{n+1}. \]

Asymptotic convergence rate for \( n_f = \frac{T}{\Delta t} \)

\[ \| x^{n_f} - x(n_f\Delta t) \| \leq C_T \Delta t. \]
Single Fault Scenario

Assume that computations encounter a silent fault and at step \( j \), instead of \( x^j \) we compute a corrupted

\[
\tilde{x}^j = x^j + \tilde{x}.
\]

For \( n > j \) all time steps will be corrupted \( \tilde{x}^n \).

Then at the final time step we have that

\[
\|\tilde{x}^{n_f} - x(n_f \Delta t)\| \leq \|\tilde{x}^{n_f} - x^{n_f}\| + \|x^{n_f} - x(n_f \Delta t)\|
\leq \|(I + \Delta tA)^{n_f-j} \tilde{x}\| + C_T \Delta t
\leq \|(I + \Delta tA)^{n_f-j}\|\|\tilde{x}\| + C_T \Delta t
\leq B_T \|\tilde{x}\| + C_T \Delta t,
\]

where

\[
B_T = \max(1, (I + \Delta tA)^{n_f}).
\]
According to the Mean Value Theorem

\[ x^{n+1} - 2x^n + x^{n-1} = \Delta t^2 \frac{1}{2} \left( \frac{d^2}{dt^2} x(\xi_l) + \frac{d^2}{dt^2} x(\xi_r) \right), \]

where \( \xi_l \in [(n - 1)\Delta t, n\Delta t] \) and \( \xi_r \in [n\Delta t, (n + 1)\Delta t] \).

Therefore,

\[ \| x^{n+1} - 2x^n + x^{n-1} \| \leq \Delta t^2 \alpha, \]

for any \( \alpha \geq \| \frac{d^2}{dt^2} x \|_\infty. \)
Resilience Enhancement

At each step of the integration we accept $x^{n+1}$ only if

$$\|x^{n+1} - 2x^n + x^{n-1}\| \leq \Delta t^2 \alpha.$$ 

Thus, if we encounter a fault $\tilde{x}$ it will be rejected unless

$$\|x^{n+1} + \tilde{x} - 2x^n + x^{n-1}\| \leq \Delta t^2 \alpha \quad \Rightarrow \quad \|\tilde{x}\| \leq 2\alpha \Delta t^2.$$

If we encounter a single fault

$$\|\tilde{x}^{n_f} - x(n_f \Delta t)\| \leq 2\alpha B_T \Delta t^2 + C_T \Delta t.$$ 

We expect to encounter $pn_f$ faults in $n_f = \frac{T}{\Delta t}$ iterations, then

$$\|\tilde{x}^{n_f} - x(n_f \Delta t)\| \leq 2\alpha p B_T T \Delta t + C_T \Delta t = (2\alpha p B_T T + C_T) \Delta t.$$
Resilient Time Stepping

Given $A$, $x^0$, $\Delta t$, $T$

Given $\alpha$

Set $n = 0$

while $n \Delta t < T$ do

Compute $x^{n+1}$, e.g., $x^{n+1} = x^n + \Delta t Ax^n$

if $\|x^{n+1} - 2x^n + x^{n-1}\| > \alpha \Delta t^2$ then

discard $x^{n+1}$ and redo the last step

else

accept $x^{n+1}$ and advance to the next $n = n + 1$

end if

end while
Consider the heat transfer equation

\[ \frac{d}{dt} \mathcal{X} = 10^{-3} \mathcal{X}_{\xi\xi}, \quad \mathcal{X}(t, 0) = \mathcal{X}(t, 1) = 0, \quad \mathcal{X}(0, \xi) = \xi(1 - \xi). \]

We discretize the problem using finite difference scheme resulting in the system of linear ODEs

\[ \frac{d}{dt} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-2} \\ x_{N-1} \end{pmatrix} = 10^{-3} (N + 1)^2 \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & \cdots & 1 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-2} \\ x_{N-1} \end{pmatrix}. \]

For the numerical experiments we use \( N = 1024 \).
Figure: Lack of convergence for the classical backward Euler scheme for one and multiple faults when $p = 0.1$. 
Figure: Convergence for the resilient time stepping method for different values of $p$. 
Time Stepping Conclusions

- We have a resilient first order time-stepping algorithm.
- The method extends to non-linear problems as well as variable time step $\Delta t$ (so long as we have continuity w.r.t. initial data)
- The resilient method preserves the linear convergence rate even in the presence of faults.
Questions?