Exploiting Multiprecision Hardware in Solving Linear Systems and Least Squares Problems

Erin C. Carson,
KNM, MFF, Charles University

Current Problems in Numerical Analysis Seminar
Institute of Mathematics, Czech Academy of Sciences
December 14, 2018
Use of low precision in machine learning has driven emergence of low-precision capabilities in hardware:

- Half precision (FP16) defined as storage format in 2008 IEEE standard
- **ARM NEON**: SIMD architecture, instructions for 8x16-bit, 4x32-bit, 2x64-bit
- **AMD Radeon Instinct MI25 GPU**, 2017:
  - single: 12.3 TFLOPS, half: 24.6 TFLOPS
- **NVIDIA Tesla P100**, 2016: native ISA support for 16-bit FP arithmetic
- **NVIDIA Tesla V100**, 2017: tensor cores for half precision;
  - 4x4 matrix multiply in one clock cycle
  - double: 7 TFLOPS, half+tensor: 112 TFLOPS \((16x!)\)
- **Google's Tensor processing unit** (TPU): quantizes 32-bit FP computations into 8-bit integer arithmetic
- **Future exascale supercomputers**: (~2021) Expected extensive support for reduced-precision arithmetic (32/16/8-bit)
Performance of LU factorization on an NVIDIA V100 GPU

[Haidar, Tomov, Dongarra, Higham, 2018]
Iterative Refinement for $Ax = b$

Iterative refinement: well-established method for improving an approximate solution to $Ax = b$

$A$ is $n \times n$ and nonsingular; $u$ is unit roundoff

Solve $Ax_0 = b$ by LU factorization

for $i = 0$: maxit

$$r_i = b - Ax_i$$

Solve $Ad_i = r_i$ via $d_i = U^{-1}(L^{-1}r_i)$

$$x_{i+1} = x_i + d_i$$
Iterative Refinement for $Ax = b$

Iterative refinement: well-established method for improving an approximate solution to $Ax = b$

$A$ is $n \times n$ and nonsingular; $u$ is unit roundoff

Solve $Ax_0 = b$ by LU factorization (in precision $u$)

for $i = 0$: maxit

$$r_i = b - Ax_i$$ (in precision $u^2$)

Solve $Ad_i = r_i$ via $d_i = U^{-1}(L^{-1}r_i)$ (in precision $u$)

$$x_{i+1} = x_i + d_i$$ (in precision $u$)

"Traditional" (high-precision residual computation)

[Wilkinson, 1948] (fixed point), [Moler, 1967] (floating point)
Iterative Refinement for $Ax = b$

As long as $\kappa_\infty(A) \leq u^{-1}$,
- relative forward error is $O(u)$
- relative normwise and componentwise backward errors are $O(u)$

$$\kappa_\infty(A) = \|A^{-1}\|_\infty \|A\|_\infty$$
$$\text{cond}(A, x) = \| |A^{-1}| \|A\| \|x\|_\infty / \|x\|_\infty$$

Solve $Ax_0 = b$ by LU factorization

for $i = 0$: maxit

$$r_i = b - Ax_i$$

Solve $Ad_i = r_i$ via $d_i = U^{-1}(L^{-1}r_i)$

$$x_{i+1} = x_i + d_i$$

"Traditional" (high-precision residual computation)

[Wilkinson, 1948] (fixed point), [Moler, 1967] (floating point)
Iterative Refinement for $Ax = b$

Solve $Ax_0 = b$ by LU factorization (in precision $u$)

for $i = 0$: maxit

$$r_i = b - Ax_i$$ (in precision $u$)

Solve $Ad_i = r_i$ via $d_i = U^{-1}(L^{-1}r_i)$ (in precision $u$)

$$x_{i+1} = x_i + d_i$$ (in precision $u$)

"Fixed-Precision"

[Jankowski and Woźniakowski, 1977], [Skeel, 1980], [Higham, 1991]
Iterative Refinement for $Ax = b$

As long as $\kappa_\infty(A) \leq u^{-1}$, 
• relative forward error is $O(u)\text{cond}(A, x)$
• relative normwise and componentwise backward errors are $O(u)$

Solve $Ax_0 = b$ by LU factorization (in precision $u$)

for $i = 0$: maxit

\[ r_i = b - Ax_i \quad \text{(in precision $u$)} \]

Solve $Ad_i = r_i$ via $d_i = U^{-1}(L^{-1}r_i)$ (in precision $u$)

\[ x_{i+1} = x_i + d_i \quad \text{(in precision $u$)} \]

"Fixed-Precision"

[Jankowski and Woźniakowski, 1977], [Skeel, 1980], [Higham, 1991]
Iterative Refinement for $Ax = b$

Solve $Ax_0 = b$ by LU factorization

for $i = 0$: maxit

$r_i = b - Ax_i$  \hspace{2cm} (in precision $u^{1/2}$)

Solve $Ad_i = r_i$ \hspace{1cm} via $d_i = U^{-1}(L^{-1}r_i)$ \hspace{1cm} (in precision $u$)

$x_{i+1} = x_i + d_i$ \hspace{1cm} (in precision $u$)

"Low-precision factorization"

[Langou et al., 2006], [Arioli and Duff, 2009], [Hogg and Scott, 2010], [Abdelfattah et al., 2016]
Iterative Refinement for $Ax = b$

As long as $\kappa_\infty(A) \leq u^{-1/2}$,
- relative forward error is $O(u) \text{cond}(A, x)$
- relative normwise and componentwise backward errors are $O(u)$

Solve $Ax_0 = b$ by LU factorization (in precision $u^{1/2}$)

for $i = 0$: maxit

\[ r_i = b - Ax_i \]  
\[ \text{Solve } Ad_i = r_i \quad \text{via } d_i = U^{-1}(L^{-1}r_i) \]  
\[ x_{i+1} = x_i + d_i \]  

"Low-precision factorization"

[Langou et al., 2006], [Arioli and Duff, 2009], [Hogg and Scott, 2010], [Abdelfattah et al., 2016]
Iterative Refinement in 3 Precisions

Existing analyses only support at most two precisions.

Can we combine the performance benefits of low-precision factorization IR with the accuracy of traditional IR?
Iterative Refinement in 3 Precisions

Existing analyses only support at most two precisions.

Can we combine the performance benefits of low-precision factorization IR with the accuracy of traditional IR?

\[ \Rightarrow 3\text{-precision iterative refinement} \]

\[ u_f = \text{factorization precision}, \quad u = \text{working precision}, \quad u_r = \text{residual precision} \]

\[ u_f \geq u \geq u_r \]
Iterative Refinement in 3 Precisions

Existing analyses only support at most two precisions

Can we combine the performance benefits of low-precision factorization IR with the accuracy of traditional IR?

⇒ 3-precision iterative refinement

\[ u_f = \text{factorization precision}, \quad u = \text{working precision}, \quad u_r = \text{residual precision} \]

\[ u_f \geq u \geq u_r \]

• New analysis generalizes existing types of IR:

<table>
<thead>
<tr>
<th>Traditional</th>
<th>( u_f = u, u_r = u^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed precision</td>
<td>( u_f = u = u_r )</td>
</tr>
<tr>
<td>Lower precision factorization</td>
<td>( u_f^2 = u = u_r )</td>
</tr>
</tbody>
</table>

(and improves upon existing analyses in some cases)

[C. and Higham, SIAM SISC 40(2), 2018]
Can we combine the performance benefits of low-precision factorization IR with the accuracy of traditional IR?

⇒ 3-precision iterative refinement

\[ u_f = \text{factorization precision}, \quad u = \text{working precision}, \quad u_r = \text{residual precision} \]

\[ u_f \geq u \geq u_r \]

• New analysis generalizes existing types of IR:

<table>
<thead>
<tr>
<th></th>
<th>Traditional</th>
<th>Fixed precision</th>
<th>Lower precision factorization</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ u_f = u, u_r = u^2 ]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(C. and Higham, SIAM SISC 40(2), 2018)

(and improves upon existing analyses in some cases)

• Enables new types of IR: (half, single, double), (half, single, quad), (half, double, quad), etc.
Obtain tighter upper bounds:

Typical bounds used in analysis: $\|A(x - \hat{x}_i)\|_\infty \leq \|A\|_\infty \|x - \hat{x}_i\|_\infty$
Obtain tighter upper bounds:

Typical bounds used in analysis: 

\[ \| A(x - \hat{x}_i) \|_\infty \leq \| A \|_\infty \| x - \hat{x}_i \|_\infty \]

Define \( \mu_i \):

\[ \| A(x - \hat{x}_i) \|_\infty = \mu_i \| A \|_\infty \| x - \hat{x}_i \|_\infty \]
Obtain tighter upper bounds:

Typical bounds used in analysis: \[ \|A(x - \hat{x}_i)\|_\infty \leq \|A\|_\infty \|x - \hat{x}_i\|_\infty \]

Define \( \mu_i \): \[ \|A(x - \hat{x}_i)\|_\infty = \mu_i \|A\|_\infty \|x - \hat{x}_i\|_\infty \]

For a stable refinement scheme, in early stages we expect

\[ \frac{\|r_i\|}{\|A\|\|\hat{x}_i\|} \approx u \ll \frac{\|x - \hat{x}_i\|}{\|x\|} \]

\( \mu_i \ll 1 \)
Obtain tighter upper bounds:

Typical bounds used in analysis: \( \|A(x - \hat{x}_i)\|_\infty \leq \|A\|_\infty \|x - \hat{x}_i\|_\infty \)

Define \( \mu_i \): \( \|A(x - \hat{x}_i)\|_\infty = \mu_i \|A\|_\infty \|x - \hat{x}_i\|_\infty \)

For a stable refinement scheme, in early stages we expect

\[
\frac{\|r_i\|}{\|A\| \|\hat{x}_i\|} \approx u \ll \frac{\|x - \hat{x}_i\|}{\|x\|}
\]

\( \mu_i \ll 1 \)

But close to convergence,

\( \|r_i\| \approx \|A\| \|x - \hat{x}_i\| \)

\( \mu_i \approx 1 \)
\[ \|r_i\|_2 = \mu_i^{(2)} \|A\|_2 \|x - \hat{x}_i\|_2 \]

\[ x - \hat{x}_i = V \Sigma^{-1} U^T r_i = \sum_{j=1}^{n} \frac{(u_j^T r_i)v_j}{\sigma_j} \quad (A = U \Sigma V^T) \]
$$\|r_i\|_2 = \mu_i^{(2)} \| A \|_2 \| x - \hat{x}_i \|_2$$

$$x - \hat{x}_i = V \Sigma^{-1} U^T r_i = \sum_{j=1}^{n} \frac{(u_j^T r_i)v_j}{\sigma_j}$$  \hspace{1cm} (A = U \Sigma V^T)

$$\|x - \hat{x}_i\|_2^2 \geq \sum_{j=n+1-k}^{n} \frac{(u_j^T r_i)^2}{\sigma_j^2} \geq \frac{1}{\sigma_{n+1-k}^2} \sum_{j=n+1-k}^{n} (u_j^T r_i)^2 = \frac{\|P_k r_i\|_2^2}{\sigma_{n+1-k}^2}$$

where $P_k = U_k U_k^T$, $U_k = [u_{n+1-k}, ..., u_n]$
\[ \|r_i\|_2 = \mu_i^{(2)} \|A\|_2 \|x - \hat{x}_i\|_2 \]

\[ x - \hat{x}_i = V \Sigma^{-1} U^T r_i = \sum_{j=1}^{n} \frac{(u_j^T r_i)v_j}{\sigma_j} \quad (A = U \Sigma V^T) \]

\[ \|x - \hat{x}_i\|^2_2 \geq \sum_{j=n+1-k}^{n} \frac{\left(u_j^T r_i\right)^2}{\sigma_j^2} \geq \frac{1}{\sigma_{n+1-k}^2} \sum_{j=n+1-k}^{n} \left(u_j^T r_i\right)^2 = \frac{\|P_k r_i\|^2_2}{\sigma_{n+1-k}^2} \]

where \( P_k = U_k U_k^T, U_k = [u_{n+1-k}, \ldots, u_n] \)

\[ \mu_i^{(2)} \leq \frac{\|r_i\|_2 \sigma_{n+1-k}}{\|P_k r_i\|_2 \sigma_1} \]
Key Analysis Innovations I

\[ \| r_i \|_2 = \mu_i^{(2)} \| A \|_2 \| x - \hat{x}_i \|_2 \]

\[ x - \hat{x}_i = V \Sigma^{-1} U^T r_i = \sum_{j=1}^{n} \left( \frac{u_j^T r_i}{\sigma_j} \right) v_j \quad (A = U \Sigma V^T) \]

\[ \| x - \hat{x}_i \|_2^2 \geq \sum_{j=n+1-k}^{n} \left( \frac{u_j^T r_i}{\sigma_j^2} \right)^2 \geq \frac{1}{\sigma_{n+1-k}^2} \sum_{j=n+1-k}^{n} (u_j^T r_i)^2 = \frac{\| P_k r_i \|_2^2}{\sigma_{n+1-k}^2} \]

where \( P_k = U_k U_k^T, U_k = [u_{n+1-k}, ..., u_n] \)

\[ \mu_i^{(2)} \leq \frac{\| r_i \|_2}{\| P_k r_i \|_2} \frac{\sigma_{n+1-k}}{\sigma_1} \]

- \( \mu_i^{(2)} \ll 1 \) if \( r_i \) contains significant component in \( \text{span}(U_k) \) for any \( k \) s.t. \( \sigma_{n+1-k} \approx \sigma_n \)
\[ \|r_i\|_2 = \mu^{(2)}_i \|A\|_2 \|x - \hat{x}_i\|_2 \]

\[ x - \hat{x}_i = V\Sigma^{-1}U^T r_i = \sum_{j=1}^{n} \frac{(u_j^T r_i)v_j}{\sigma_j} \quad (A = U\Sigma V^T) \]

\[ \|x - \hat{x}_i\|_2^2 \geq \sum_{j=n+1-k}^{n} \frac{(u_j^T r_i)^2}{\sigma_j^2} \geq \frac{1}{\sigma_{n+1-k}^2} \sum_{j=n+1-k}^{n} (u_j^T r_i)^2 = \frac{\|P_k r_i\|_2^2}{\sigma_{n+1-k}^2} \]

where \( P_k = U_k U_k^T \), \( U_k = [u_{n+1-k}, ..., u_n] \)

\[ \mu^{(2)}_i \leq \frac{\|r_i\|_2}{\|P_k r_i\|_2} \frac{\sigma_{n+1-k}}{\sigma_1} \]

- \( \mu^{(2)}_i \ll 1 \) if \( r_i \) contains significant component in \( \text{span}(U_k) \) for any \( k \) s.t. \( \sigma_{n+1-k} \approx \sigma_n \)
- Expect \( \mu^{(2)}_i \ll 1 \) when \( r_i \) is "typical", i.e., contains sizeable components in the direction of each left singular vector
- In that case, \( x - \hat{x}_i \) is not "typical", i.e., it contains large components in right singular vectors corresponding to small singular values of \( A \)
Key Analysis Innovations I

\[ \|r_i\|_2 = \mu_i^{(2)} \|A\|_2 \|x - \hat{x}_i\|_2 \]

\[ x - \hat{x}_i = V\Sigma^{-1}U^T r_i = \sum_{j=1}^{n} \frac{(u_j^T r_i)v_j}{\sigma_j} \quad (A = U\Sigma V^T) \]

\[ \|x - \hat{x}_i\|_2^2 \geq \sum_{j=n+1-k}^{n} \frac{(u_j^T r_i)^2}{\sigma_j^2} \geq \frac{1}{\sigma_{n+1-k}^2} \sum_{j=n+1-k}^{n} (u_j^T r_i)^2 = \frac{\|P_k r_i\|_2^2}{\sigma_{n+1-k}^2} \]

where \( P_k = U_k U_k^T, U_k = [u_{n+1-k}, ..., u_n] \)

\[ \mu_i^{(2)} \leq \frac{\|r_i\|_2}{\|P_k r_i\|_2} \frac{\sigma_{n+1-k}}{\sigma_1} \]

- \( \mu_i^{(2)} \ll 1 \) if \( r_i \) contains significant component in \( \text{span}(U_k) \) for any \( k \) s.t. \( \sigma_{n+1-k} \approx \sigma_n \)
- Expect \( \mu_i^{(2)} \ll 1 \) when \( r_i \) is "typical", i.e., contains sizeable components in the direction of each left singular vector
- In that case, \( x - \hat{x}_i \) is not "typical", i.e., it contains large components in right singular vectors corresponding to small singular values of \( A \)
- Wilkinson (1977), comment in unpublished manuscript: \( \mu_i^{(2)} \) increases with \( i \)
Allow for general solver:
Let $u_s$ be the *effective precision* of the solve, with $u \leq u_s \leq u_f$
Allow for general solver:
Let $u_s$ be the effective precision of the solve, with $u \leq u_s \leq u_f$

example: LU solve:

$$u_s = u_f$$
Allow for general solver:
Let $u_s$ be the effective precision of the solve, with $u \leq u_s \leq u_f$

Assume computed solution $\hat{d}_i$ to $Ad_i = \hat{r}_i$ satisfies:

1. $\hat{d}_i = (I + u_s E_i)d_i, \quad u_s \|E_i\|_\infty < 1$
   → normwise relative forward error is bounded by multiple of $u_s$ and is less than 1

example: LU solve: $u_s = u_f$
Allow for general solver:
Let $u_s$ be the effective precision of the solve, with $u \leq u_s \leq u_f$

Assume computed solution $\hat{d}_i$ to $Ad_i = \hat{r}_i$ satisfies:

1. $\hat{d}_i = (I + u_s E_i)d_i, \quad u_s \|E_i\|_\infty < 1$
   → normwise relative forward error is bounded by multiple of $u_s$ and is less than 1

\begin{align*}
\text{example: LU solve:} \\
&\quad u_s = u_f \\
\text{Assume computed solution } &\hat{d}_i \text{ to } Ad_i = \hat{r}_i \text{ satisfies:} \\
&\quad \hat{d}_i = (I + u_s E_i)d_i, \quad u_s \|E_i\|_\infty < 1 \\
&\quad \rightarrow \text{ normwise relative forward error is bounded} \\
&\quad \text{by multiple of } u_s \text{ and is less than 1} \\
&\quad u_s \|E_i\|_\infty \leq 3n u_f \|A^{-1}\|\|L\|\|\tilde{U}\|_\infty
\end{align*}
Allow for general solver:
Let $u_s$ be the effective precision of the solve, with $u \leq u_s \leq u_f$

Assume computed solution $\hat{d}_i$ to $Ad_i = \hat{r}_i$ satisfies:

1. $\hat{d}_i = (I + u_s E_i)d_i, \quad u_s \|E_i\|_\infty < 1$
   \[\rightarrow\text{normwise relative forward error is bounded by multiple of } u_s \text{ and is less than 1}\]

2. $\|\hat{r}_i - A\hat{d}_i\|_\infty \leq u_s (c_1 \|A\|_\infty \|\hat{d}_i\|_\infty + c_2 \|\hat{r}_i\|_\infty)$
   \[\rightarrow\text{normwise relative backward error is at most } \max(c_1, c_2) u_s\]

\[\|E_i\|_\infty \leq 3n u_f \||A^{-1}||\hat{L}||\hat{U}||_\infty\]
Allow for general solver:
Let \( u_s \) be the effective precision of the solve, with \( u \leq u_s \leq u_f \)

Assume computed solution \( \hat{d}_i \) to \( A\hat{d}_i = \hat{r}_i \) satisfies:

1. \( \hat{d}_i = (I + u_s E_i)d_i, \quad u_s\|E_i\|_\infty < 1 \)
   \( \rightarrow \) normwise relative forward error is bounded by multiple of \( u_s \) and is less than 1

2. \( \|\hat{r}_i - A\hat{d}_i\|_\infty \leq u_s(c_1\|A\|_\infty\|d_i\|_\infty + c_2\|\hat{r}_i\|_\infty) \)
   \( \rightarrow \) normwise relative backward error is at most \( \max(c_1, c_2)u_s \)

\[
\|E_i\|_\infty \leq 3nu_\infty A^{-1}\|L\|\|U\|_\infty
\]

\[
\max(c_1, c_2)u_s \leq \frac{3nu_\infty\|L\|\|U\|_\infty}{\|A\|_\infty}
\]
Key Analysis Innovations II

Allow for general solver:
Let $\mathbf{u}_s$ be the effective precision of the solve, with $\mathbf{u} \leq \mathbf{u}_s \leq \mathbf{u}_f$

Assume computed solution $\hat{\mathbf{d}}_i$ to $A\mathbf{d}_i = \hat{\mathbf{r}}_i$ satisfies:

1. $\hat{\mathbf{d}}_i = (I + \mathbf{u}_s \mathbf{E}_i)\mathbf{d}_i$, $\mathbf{u}_s \| \mathbf{E}_i \|_\infty < 1$
   \[ \rightarrow \text{normwise relative forward error is bounded by multiple of } \mathbf{u}_s \text{ and is less than 1} \]

2. $\| \hat{\mathbf{r}}_i - A\hat{\mathbf{d}}_i \|_\infty \leq \mathbf{u}_s (c_1 \| A \|_\infty \| \hat{\mathbf{d}}_i \|_\infty + c_2 \| \hat{\mathbf{r}}_i \|_\infty)$
   \[ \rightarrow \text{normwise relative backward error is at most } \max(c_1, c_2) \mathbf{u}_s \]

3. $| \hat{\mathbf{r}}_i - A\hat{\mathbf{d}}_i | \leq \mathbf{u}_s G_i | \hat{\mathbf{d}}_i |$
   \[ \rightarrow \text{componentwise relative backward error is bounded by a multiple of } \mathbf{u}_s \]

example: LU solve:

\[ \mathbf{u}_s = \mathbf{u}_f \]

\[ \mathbf{u}_s \| \mathbf{E}_i \|_\infty \leq 3n \mathbf{u}_f \| A^{-1} \| \| L \| \| U \| \infty \]

\[ \max(c_1, c_2) \mathbf{u}_s \leq \frac{3n \mathbf{u}_f \| L \| \| U \| \infty}{\| A \|_\infty} \]
Allow for general solver:
Let $u_s$ be the effective precision of the solve, with $u \leq u_s \leq u_f$

Assume computed solution $\hat{d}_i$ to $Ad_i = \hat{r}_i$ satisfies:

1. $\hat{d}_i = (I + u_s E_i)d_i, \quad u_s \|E_i\|_\infty < 1$
   \[\rightarrow \text{normwise relative forward error is bounded by multiple of } u_s \text{ and is less than 1}\]

2. $\|\hat{r}_i - A\hat{d}_i\|_\infty \leq u_s (c_1 \|A\|_\infty \|\hat{d}_i\|_\infty + c_2 \|\hat{r}_i\|_\infty)$
   \[\rightarrow \text{normwise relative backward error is at most} \max(c_1, c_2) u_s\]

3. $|\hat{r}_i - A\hat{d}_i| \leq u_s G_i |\hat{d}_i|$
   \[\rightarrow \text{componentwise relative backward error is bounded by a multiple of } u_s\]
Allow for general solver:
Let $u_s$ be the *effective precision* of the solve, with $u \leq u_s \leq u_f$

Assume computed solution $\hat{d}_i$ to $A\hat{d}_i = \hat{r}_i$ satisfies:

1. $\hat{d}_i = (I + u_s E_i)d_i, \quad u_s \|E_i\|_\infty < 1$
   \[ \rightarrow \text{normwise relative forward error is bounded by multiple of } u_s \text{ and is less than } 1 \]

2. $\|\hat{r}_i - A\hat{d}_i\|_\infty \leq u_s (c_1 \|A\|_\infty \|\hat{d}_i\|_\infty + c_2 \|\hat{r}_i\|_\infty)$
   \[ \rightarrow \text{normwise relative backward error is at most } \max(c_1, c_2) u_s \]

3. $|\hat{r}_i - A\hat{d}_i| \leq u_s G_i |\hat{d}_i|$
   \[ \rightarrow \text{componentwise relative backward error is bounded by a multiple of } u_s \]

$E_i, c_1, c_2, \text{ and } G_i \text{ depend on } A, \hat{r}_i, n, \text{ and } u_s$
Forward Error for IR3

• Three precisions:
  • $u_f$: factorization precision
  • $u$: working precision
  • $u_r$: residual computation precision

$$\kappa_\infty(A) = \|A^{-1}\|_{\infty} \|A\|_{\infty}$$

$$\text{cond}(A) = \| |A^{-1}| |A| \|_{\infty}$$

$$\text{cond}(A, x) = \| |A^{-1}| |A||x| \|_{\infty}/\|x\|_{\infty}$$
Forward Error for IR3

• Three precisions:
  • $u_f$: factorization precision
  • $u$: working precision
  • $u_r$: residual computation precision

For IR in precisions $u_f \geq u \geq u_r$ and effective solve precision $u_s$, if

$$\phi_i \equiv 2u_s \min(\text{cond}(A), \kappa_\infty(A)\mu_i) + u_s \|E_i\|_\infty$$

is sufficiently less than 1, then the forward error is reduced on the $i$th iteration by a factor $\approx \phi_i$ until an iterate $\hat{x}_i$ is produced for which

$$\frac{\|x - \hat{x}_i\|_\infty}{\|x\|_\infty} \leq 4N u_r \text{cond}(A, x) + u,$$

where $N$ is the maximum number of nonzeros per row in $A$. 

Theorem [C. and Higham, SISC 40(2), 2018]
Forward Error for IR3

• Three precisions:
  • $u_f$: factorization precision
  • $u$: working precision
  • $u_r$: residual computation precision

Analogous traditional bounds:
\[
\phi_i \equiv 3n u_f \kappa_{\infty}(A)
\]

Theorem [C. and Higham, SISC 40(2), 2018]

For IR in precisions $u_f \geq u \geq u_r$ and effective solve precision $u_s$, if
\[
\phi_i \equiv 2u_s \min(\text{cond}(A), \kappa_{\infty}(A)\mu_i) + u_s \|E_i\|_{\infty}
\]

is sufficiently less than 1, then the forward error is reduced on the $i$th iteration by a factor $\approx \phi_i$ until an iterate $\hat{x}_i$ is produced for which
\[
\frac{\|x - \hat{x}_i\|_{\infty}}{\|x\|_{\infty}} \leq 4Nu_r \text{cond}(A, x) + u,
\]

where $N$ is the maximum number of nonzeros per row in $A$. 

Analogous traditional bounds: $\phi_i \equiv 3n u_f \kappa_{\infty}(A)$
For IR in precisions $\mathbf{u}_f \geq \mathbf{u} \geq \mathbf{u}_r$ and effective solve precision $\mathbf{u}_s$, if

$$\phi_i \equiv (c_1 \kappa_\infty(A) + c_2) \mathbf{u}_s$$

is sufficiently less than 1, then the residual is reduced on the $i$th iteration by a factor $\approx \phi_i$ until an iterate $\hat{x}_i$ is produced for which

$$\|\mathbf{b} - A\hat{x}_i\|_\infty \lesssim N\mathbf{u}(\|\mathbf{b}\|_\infty + \|A\|_\infty \|\hat{x}_i\|_\infty),$$

where $N$ is the maximum number of nonzeros per row in $A$. 
IR3: Summary

Standard (LU-based) IR in three precisions \((u_s = u_f)\)

Half \(\approx 10^{-4}\), Single \(\approx 10^{-8}\), Double \(\approx 10^{-16}\), Quad \(\approx 10^{-34}\)

<table>
<thead>
<tr>
<th>(u_f)</th>
<th>(u)</th>
<th>(u_r)</th>
<th>(\text{max } \kappa_{\infty}(A))</th>
<th>Backward error</th>
<th>Forward error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(\text{norm})</td>
<td>(\text{comp})</td>
</tr>
<tr>
<td>H</td>
<td>S</td>
<td>S</td>
<td>10^4</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>H</td>
<td>S</td>
<td>D</td>
<td>10^4</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>H</td>
<td>D</td>
<td>D</td>
<td>10^4</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>10^4</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>10^8</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>D</td>
<td>10^8</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>S</td>
<td>D</td>
<td>D</td>
<td>10^8</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>10^8</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
</tbody>
</table>
Standard (LU-based) IR in three precisions ($u_s = u_f$)

Half $\approx 10^{-4}$, Single $\approx 10^{-8}$, Double $\approx 10^{-16}$, Quad $\approx 10^{-34}$

<table>
<thead>
<tr>
<th>$u_f$</th>
<th>$u$</th>
<th>$u_r$</th>
<th>$\max \kappa_\infty(A)$</th>
<th>Backward error</th>
<th>Forward error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>norm</td>
<td>comp</td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>S</td>
<td>S</td>
<td>10$^4$</td>
<td>10$^{-8}$</td>
<td>10$^{-8}$</td>
</tr>
<tr>
<td>H</td>
<td>S</td>
<td>D</td>
<td>10$^4$</td>
<td>10$^{-8}$</td>
<td>10$^{-8}$</td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>D</td>
<td>D</td>
<td>10$^4$</td>
<td>10$^{-16}$</td>
<td>10$^{-16}$</td>
</tr>
<tr>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>10$^4$</td>
<td>10$^{-16}$</td>
<td>10$^{-16}$</td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>10$^8$</td>
<td>10$^{-8}$</td>
<td>10$^{-8}$</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>D</td>
<td>10$^8$</td>
<td>10$^{-8}$</td>
<td>10$^{-8}$</td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>D</td>
<td>D</td>
<td>10$^8$</td>
<td>10$^{-16}$</td>
<td>10$^{-16}$</td>
</tr>
<tr>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>10$^8$</td>
<td>10$^{-16}$</td>
<td>10$^{-16}$</td>
</tr>
</tbody>
</table>
**IR3: Summary**

Standard (LU-based) IR in three precisions ($u_s = u_f$)

Half $\approx 10^{-4}$, Single $\approx 10^{-8}$, Double $\approx 10^{-16}$, Quad $\approx 10^{-34}$

<table>
<thead>
<tr>
<th>$u_f$</th>
<th>$u$</th>
<th>$u_r$</th>
<th>$\max \kappa_\infty (A)$</th>
<th>Backward error</th>
<th>Forward error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>norm</td>
<td>comp</td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>S</td>
<td>S</td>
<td>10^4</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>H</td>
<td>S</td>
<td>D</td>
<td>10^4</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>D</td>
<td>D</td>
<td>10^4</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>10^4</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>10^8</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>D</td>
<td>10^8</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>D</td>
<td>D</td>
<td>10^8</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>10^8</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
</tbody>
</table>
IR3: Summary

Standard (LU-based) IR in three precisions ($u_s = u_f$)

Half $\approx 10^{-4}$, Single $\approx 10^{-8}$, Double $\approx 10^{-16}$, Quad $\approx 10^{-34}$

<table>
<thead>
<tr>
<th>$u_f$</th>
<th>$u$</th>
<th>$u_r$</th>
<th>$\max \kappa_\infty (A)$</th>
<th>Backward error</th>
<th>Forward error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>norm</td>
<td>comp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>S</td>
<td>S</td>
<td>10^4</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>H</td>
<td>S</td>
<td>D</td>
<td>10^4</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>D</td>
<td>D</td>
<td>10^4</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>10^4</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>10^8</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>Trad.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>D</td>
<td>10^8</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>D</td>
<td>D</td>
<td>10^8</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>10^8</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
</tbody>
</table>
Standard (LU-based) IR in three precisions ($u_s = u_f$)

Half $\approx 10^{-4}$, Single $\approx 10^{-8}$, Double $\approx 10^{-16}$, Quad $\approx 10^{-34}$

<table>
<thead>
<tr>
<th></th>
<th>$u_f$</th>
<th>$u$</th>
<th>$u_r$</th>
<th>$\max \kappa_\infty(A)$</th>
<th>Backward error norm</th>
<th>Backward error comp</th>
<th>Forward error</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP fact.</td>
<td>H</td>
<td>S</td>
<td>S</td>
<td>$10^4$</td>
<td>$10^{-8}$</td>
<td>$10^{-8}$</td>
<td>$\text{cond}(A,x) \cdot 10^{-8}$</td>
</tr>
<tr>
<td>New</td>
<td>H</td>
<td>S</td>
<td>D</td>
<td>$10^4$</td>
<td>$10^{-8}$</td>
<td>$10^{-8}$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>LP fact.</td>
<td>H</td>
<td>D</td>
<td>D</td>
<td>$10^4$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
<td>$\text{cond}(A,x) \cdot 10^{-16}$</td>
</tr>
<tr>
<td>New</td>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>$10^4$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>Fixed</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>$10^8$</td>
<td>$10^{-8}$</td>
<td>$10^{-8}$</td>
<td>$\text{cond}(A,x) \cdot 10^{-8}$</td>
</tr>
<tr>
<td>Trad.</td>
<td>S</td>
<td>S</td>
<td>D</td>
<td>$10^8$</td>
<td>$10^{-8}$</td>
<td>$10^{-8}$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>LP fact.</td>
<td>S</td>
<td>D</td>
<td>D</td>
<td>$10^8$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
<td>$\text{cond}(A,x) \cdot 10^{-16}$</td>
</tr>
<tr>
<td>New</td>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>$10^8$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
</tr>
</tbody>
</table>
Standard (LU-based) IR in three precisions ($u_s = u_f$)

Half $\approx 10^{-4}$, Single $\approx 10^{-8}$, Double $\approx 10^{-16}$, Quad $\approx 10^{-34}$

<table>
<thead>
<tr>
<th>$u_f$</th>
<th>$u$</th>
<th>$u_r$</th>
<th>$\max \kappa_\infty(A)$</th>
<th>Backward error</th>
<th>Forward error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>norm</td>
<td>comp</td>
</tr>
<tr>
<td>LP fact.</td>
<td>H</td>
<td>S</td>
<td>S</td>
<td>10^4</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>New</td>
<td>H</td>
<td>S</td>
<td>D</td>
<td>10^4</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>LP fact.</td>
<td>H</td>
<td>D</td>
<td>D</td>
<td>10^4</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>New</td>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>10^4</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>Fixed</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>10^8</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>Trad.</td>
<td>S</td>
<td>S</td>
<td>D</td>
<td>10^8</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>LP fact.</td>
<td>S</td>
<td>D</td>
<td>D</td>
<td>10^8</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>New</td>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>10^8</td>
<td>10^{-16}</td>
</tr>
</tbody>
</table>

$\Rightarrow$ Benefit of IR3 vs. "LP fact.": no $\text{cond}(A, x)$ term in forward error
IR3: Summary

Standard (LU-based) IR in three precisions ($u_s = u_f$)

Half $\approx 10^{-4}$, Single $\approx 10^{-8}$, Double $\approx 10^{-16}$, Quad $\approx 10^{-34}$

<table>
<thead>
<tr>
<th>$u_f$</th>
<th>$u$</th>
<th>$u_r$</th>
<th>max $\kappa_\infty(A)$</th>
<th>Backward error</th>
<th>Forward error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>norm</td>
<td>comp</td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>S</td>
<td>S</td>
<td>10^4</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>New</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>S</td>
<td>D</td>
<td>10^4</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>D</td>
<td>D</td>
<td>10^4</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>New</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>10^4</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>10^8</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td>Trad.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>D</td>
<td>10^8</td>
<td>10^{-8}</td>
<td>10^{-8}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LP fact.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>D</td>
<td>D</td>
<td>10^8</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
<tr>
<td>New</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>10^8</td>
<td>10^{-16}</td>
<td>10^{-16}</td>
</tr>
</tbody>
</table>

$\Rightarrow$ Benefit of IR3 vs. traditional IR: As long as $\kappa_\infty(A) \leq 10^4$, can use lower precision factorization w/no loss of accuracy!
A = gallery('randsvd', 100, 1e9, 2)
b = randn(100,1)

\( \kappa_\infty(A) \approx 2e10, \ \text{cond}(A, x) \approx 5e9 \)
A = gallery('randsvd', 100, 1e9, 2)
b = randn(100, 1)

\( \kappa_{\infty}(A) \approx 2e10, \ \text{cond}(A, x) \approx 5e9 \)

Standard (LU-based) IR with \( u_f \): single, \( u \): double, \( u_r \): quad
A = gallery('randsvd', 100, 1e9, 2)
b = randn(100,1)

\( \kappa_\infty(A) \approx 2e10, \quad \text{cond}(A, x) \approx 5e9 \)

Standard (LU-based) IR with \( u_f \): single, \( u \): double, \( u_r \): quad
A = gallery('randsvd', 100, 1e9, 2)
b = randn(100,1)

\[ \kappa_\infty(A) \approx 2e10, \quad \text{cond}(A, x) \approx 5e9 \]

Standard (LU-based) IR with \( u_f \): double, \( u \): double, \( u_r \): quad
GMRES-Based Iterative Refinement

- Observation [Rump, 1990]: if \( \hat{L} \) and \( \hat{U} \) are computed LU factors of \( A \) in precision \( u_f \), then

\[
\kappa_\infty(\hat{U}^{-1}\hat{L}^{-1}A) \approx 1 + \kappa_\infty(A)u_f,
\]

even if \( \kappa_\infty(A) \gg u_f^{-1} \).
• Observation [Rump, 1990]: if $\hat{L}$ and $\hat{U}$ are computed LU factors of $A$ in precision $u_f$, then

$$\kappa_\infty(\hat{U}^{-1}\hat{L}^{-1}A) \approx 1 + \kappa_\infty(A)u_f,$$

even if $\kappa_\infty(A) \gg u_f^{-1}$.

**GMRES-IR** [C. and Higham, SISC 39(6), 2017]

• To compute the updates $d_i$, apply GMRES to

$$\hat{U}^{-1}\hat{L}^{-1}Ad_i = \hat{U}^{-1}\hat{L}^{-1}r_i$$
GMRES-Based Iterative Refinement

- Observation [Rump, 1990]: if $\hat{L}$ and $\hat{U}$ are computed LU factors of $A$ in precision $u_f$, then

$$κ_∞(\hat{U}^{-1}\hat{L}^{-1}A) ≈ 1 + κ_∞(A)u_f,$$

even if $κ_∞(A) \gg u_f^{-1}$.

GMRES-IR [C. and Higham, SISC 39(6), 2017]

- To compute the updates $d_i$, apply GMRES to

$$\hat{U}^{-1}\hat{L}^{-1}Ad_i = \hat{U}^{-1}\hat{L}^{-1}r_i$$

Solve $Ax_0 = b$ by LU factorization for $i = 0$: maxit

$$r_i = b - Ax_i$$

Solve $Ad_i = r_i$ via GMRES on $\tilde{A}d_i = \tilde{r}_i$

$$x_{i+1} = x_i + d_i$$
GMRES-Based Iterative Refinement

• Observation [Rump, 1990]: if \( \hat{L} \) and \( \hat{U} \) are computed LU factors of \( A \) in precision \( u_f \), then

\[
\kappa_\infty (\hat{U}^{-1}\hat{L}^{-1}A) \approx 1 + \kappa_\infty (A) u_f,
\]

even if \( \kappa_\infty (A) \gg u_f^{-1} \).

GMRES-IR [C. and Higham, SISC 39(6), 2017]

• To compute the updates \( d_i \), apply GMRES to

\[
\hat{U}^{-1}\hat{L}^{-1}Ad_i = \hat{U}^{-1}\hat{L}^{-1}r_i
\]

Solve \( Ax_0 = b \) by LU factorization for \( i = 0 \): maxit

\[
\begin{align*}
    r_i &= b - Ax_i \\
    x_{i+1} &= x_i + d_i
\end{align*}
\]

via GMRES on \( \tilde{A}d_i = \tilde{r}_i \)
A = gallery('randsvd', 100, 1e9, 2)
b = randn(100,1)

\[ \kappa_\infty(A) \approx 2e10, \quad \text{cond}(A, x) \approx 5e9 \]
$$A = \text{gallery('randsvd', 100, 1e9, 2)}$$
$$b = \text{randn(100,1)}$$

$$\kappa_\infty(A) \approx 2e10, \quad \text{cond}(A, x) \approx 5e9, \quad \kappa_\infty(\tilde{A}) \approx 2e4$$

**GMRES-IR** with $u_f$: single, $u$: double, $u_r$: quad
## GMRES-IR: Summary

### Benefits of GMRES-IR:

<table>
<thead>
<tr>
<th>$u_f$</th>
<th>$u$</th>
<th>$u_r$</th>
<th>$\max \kappa_\infty (A)$</th>
<th>Backward error</th>
<th>Forward error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>norm</td>
<td>comp</td>
</tr>
<tr>
<td><strong>LU-IR</strong></td>
<td>H</td>
<td>S</td>
<td>D</td>
<td>$10^4$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td><strong>GMRES-IR</strong></td>
<td>H</td>
<td>S</td>
<td>D</td>
<td>$10^8$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td><strong>LU-IR</strong></td>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>$10^8$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td><strong>GMRES-IR</strong></td>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>$10^{16}$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td><strong>LU-IR</strong></td>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>$10^4$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td><strong>GMRES-IR</strong></td>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>$10^{12}$</td>
<td>$10^{-16}$</td>
</tr>
</tbody>
</table>
## GMRES-IR: Summary

### Benefits of GMRES-IR:

<table>
<thead>
<tr>
<th></th>
<th>$u_f$</th>
<th>$u$</th>
<th>$u_r$</th>
<th>$\max \kappa_\infty(A)$</th>
<th>Backward error</th>
<th>Forward error</th>
</tr>
</thead>
<tbody>
<tr>
<td>LU-IR</td>
<td>H</td>
<td>S</td>
<td>D</td>
<td>$10^4$</td>
<td>$10^{-8}$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>GMRES-IR</td>
<td>H</td>
<td>S</td>
<td>D</td>
<td>$10^8$</td>
<td>$10^{-8}$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>LU-IR</td>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>$10^8$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>GMRES-IR</td>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>$10^{16}$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>LU-IR</td>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>$10^4$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>GMRES-IR</td>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>$10^{12}$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
</tr>
</tbody>
</table>

$\Rightarrow$ With GMRES-IR, lower precision factorization will work for higher $\kappa_\infty(A)$
### Benefits of GMRES-IR:

<table>
<thead>
<tr>
<th></th>
<th>$u_f$</th>
<th>$u$</th>
<th>$u_r$</th>
<th>max $\kappa_\infty(A)$</th>
<th>Backward error</th>
<th>Forward error</th>
</tr>
</thead>
<tbody>
<tr>
<td>LU-IR</td>
<td>H</td>
<td>S</td>
<td>D</td>
<td>$10^4$</td>
<td>$10^{-8}$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>GMRES-IR</td>
<td>H</td>
<td>S</td>
<td>D</td>
<td>$10^8$</td>
<td>$10^{-8}$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>LU-IR</td>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>$10^8$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>GMRES-IR</td>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>$10^{16}$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>LU-IR</td>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>$10^4$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>GMRES-IR</td>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>$10^{12}$</td>
<td>$10^{-16}$</td>
<td>$10^{-16}$</td>
</tr>
</tbody>
</table>

⇒ With GMRES-IR, lower precision factorization will work for higher $\kappa_\infty(A)$

$k_\infty(A) \leq u^{-1/2} u_f^{-1}$
**GMRES-IR: Summary**

### Benefits of GMRES-IR:

<table>
<thead>
<tr>
<th>$u_f$</th>
<th>$u$</th>
<th>$u_r$</th>
<th>$\max \kappa_{\infty}(A)$</th>
<th>Backward error</th>
<th>Forward error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\infty$ norm</td>
<td>comp</td>
<td></td>
</tr>
<tr>
<td>LU-IR</td>
<td>H</td>
<td>S</td>
<td>D</td>
<td>$10^4$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>GMRES-IR</td>
<td>H</td>
<td>S</td>
<td>D</td>
<td>$10^8$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>LU-IR</td>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>$10^8$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>GMRES-IR</td>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>$10^{16}$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>LU-IR</td>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>$10^4$</td>
<td>$10^{-16}$</td>
</tr>
<tr>
<td>GMRES-IR</td>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>$10^{12}$</td>
<td>$10^{-16}$</td>
</tr>
</tbody>
</table>

$\Rightarrow$ If $\kappa_{\infty}(A) \leq 10^{12}$, can use lower precision factorization w/no loss of accuracy!
## GMRES-IR: Summary

### Benefits of GMRES-IR:

<table>
<thead>
<tr>
<th></th>
<th>$u_f$</th>
<th>$u$</th>
<th>$u_r$</th>
<th>max $\kappa_{\infty}(A)$</th>
<th>Backward error</th>
<th>Forward error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>norm</td>
<td>comp</td>
<td></td>
</tr>
<tr>
<td>LU-IR</td>
<td>H</td>
<td>S</td>
<td>D</td>
<td>10$^4$</td>
<td>10$^{-8}$</td>
<td>10$^{-8}$</td>
</tr>
<tr>
<td>GMRES-IR</td>
<td>H</td>
<td>S</td>
<td>D</td>
<td>10$^8$</td>
<td>10$^{-8}$</td>
<td>10$^{-8}$</td>
</tr>
<tr>
<td>LU-IR</td>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>10$^8$</td>
<td>10$^{-16}$</td>
<td>10$^{-16}$</td>
</tr>
<tr>
<td>GMRES-IR</td>
<td>S</td>
<td>D</td>
<td>Q</td>
<td>10$^{16}$</td>
<td>10$^{-16}$</td>
<td>10$^{-16}$</td>
</tr>
<tr>
<td>LU-IR</td>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>10$^4$</td>
<td>10$^{-16}$</td>
<td>10$^{-16}$</td>
</tr>
<tr>
<td>GMRES-IR</td>
<td>H</td>
<td>D</td>
<td>Q</td>
<td>10$^{12}$</td>
<td>10$^{-16}$</td>
<td>10$^{-16}$</td>
</tr>
</tbody>
</table>

Try IR3! MATLAB codes available at: [https://github.com/eccarson/ir3](https://github.com/eccarson/ir3)
Comments and Caveats

• Convergence tolerance $\tau$ for GMRES?
  • Smaller $\tau \rightarrow$ more GMRES iterations, potentially fewer refinement steps
  • Larger $\tau \rightarrow$ fewer GMRES iterations, potentially more refinement steps
Comments and Caveats

- Convergence tolerance $\tau$ for GMRES?
  - Smaller $\tau \rightarrow$ more GMRES iterations, potentially fewer refinement steps
  - Larger $\tau \rightarrow$ fewer GMRES iterations, potentially more refinement steps

- Convergence rate of GMRES?
Comments and Caveats

• Convergence tolerance $\tau$ for GMRES?
  • Smaller $\tau \rightarrow$ more GMRES iterations, potentially fewer refinement steps
  • Larger $\tau \rightarrow$ fewer GMRES iterations, potentially more refinement steps

• Convergence rate of GMRES?
  • If $A$ is ill conditioned and LU factorization is performed in very low precision, it can be a poor preconditioner
    • e.g., if $\tilde{A}$ still has cluster of eigenvalues near origin, GMRES can stagnate until $n^{th}$ iteration, regardless of $\kappa_{\infty}(A)$ [Liesen and Tichý, 2004]
• Convergence tolerance $\tau$ for GMRES?
  • Smaller $\tau \rightarrow$ more GMRES iterations, potentially fewer refinement steps
  • Larger $\tau \rightarrow$ fewer GMRES iterations, potentially more refinement steps

• Convergence rate of GMRES?
  • If $A$ is ill conditioned and LU factorization is performed in very low precision, it can be a poor preconditioner
    • e.g., if $\tilde{A}$ still has cluster of eigenvalues near origin, GMRES can stagnate until $n^{th}$ iteration, regardless of $\kappa_\infty(A)$ [Liesen and Tichý, 2004]
  • Potential remedies: deflation, Krylov subspace recycling, using additional preconditioner
Comments and Caveats

• Convergence tolerance $\tau$ for GMRES?
  • Smaller $\tau \rightarrow$ more GMRES iterations, potentially fewer refinement steps
  • Larger $\tau \rightarrow$ fewer GMRES iterations, potentially more refinement steps

• Convergence rate of GMRES?
  • If $A$ is ill conditioned and LU factorization is performed in very low precision, it can be a poor preconditioner
    • e.g., if $\tilde{A}$ still has cluster of eigenvalues near origin, GMRES can stagnate until $n^{th}$ iteration, regardless of $\kappa_\infty(A)$ [Liesen and Tichý, 2004]
  • Potential remedies: deflation, Krylov subspace recycling, using additional preconditioner

• Depending on conditioning of $A$, applying $\tilde{A}$ to a vector must be done accurately (precision $u^2$) in each GMRES iteration
Comments and Caveats

• Convergence tolerance $\tau$ for GMRES?
  • Smaller $\tau \rightarrow$ more GMRES iterations, potentially fewer refinement steps
  • Larger $\tau \rightarrow$ fewer GMRES iterations, potentially more refinement steps

• Convergence rate of GMRES?
  • If $A$ is ill conditioned and LU factorization is performed in very low precision, it can be a poor preconditioner
    • e.g., if $\tilde{A}$ still has cluster of eigenvalues near origin, GMRES can stagnate until $n^{th}$ iteration, regardless of $\kappa_\infty(A)$ [Liesen and Tichý, 2004]
  • Potential remedies: deflation, Krylov subspace recycling, using additional preconditioner

• Depending on conditioning of $A$, applying $\tilde{A}$ to a vector must be done accurately (precision $u^2$) in each GMRES iteration

• Why GMRES?
  • Theoretical purposes: existing analysis and proof of backward stability [Paige, Rozložník, Strakoš, 2006]
  • In practice, use any solver you want!
Extension to Least Squares Problems

• Want to solve

\[ \min_x \| b - Ax \|_2 \]

where \( A \in \mathbb{R}^{m \times n} \ (m > n) \) has rank \( n \)
Extension to Least Squares Problems

• Want to solve

$$\min_x \|b - Ax\|_2$$

where $A \in \mathbb{R}^{m \times n}$ ($m > n$) has rank $n$

• Commonly solved using QR factorization:

$$A = QR = [Q_1, Q_2] \begin{bmatrix} U \\ 0 \end{bmatrix}$$

where $Q$ is an $m \times m$ orthogonal matrix and $U$ is upper triangular.

$$x = U^{-1}Q_1^Tb, \quad \|b - Ax\|_2 = \|Q_2^Tb\|_2$$
Extension to Least Squares Problems

• Want to solve

$$\min_x \| b - Ax \|_2$$

where $A \in \mathbb{R}^{m \times n}$ ($m > n$) has rank $n$

• Commonly solved using QR factorization:

$$A = QR = [Q_1, Q_2] \begin{bmatrix} U \\ 0 \end{bmatrix}$$

where $Q$ is an $m \times m$ orthogonal matrix and $U$ is upper triangular.

$$x = U^{-1}Q_1^T b, \quad \|b - Ax\|_2 = \|Q_2^T b\|_2$$

• As in linear system case, for ill-conditioned problems, iterative refinement often needed to improve accuracy and stability
Least Squares Iterative Refinement

• For inconsistent systems, must simultaneously refine both solution and residual

• (Björck, 1967): Least squares problem can be written as a linear system with square matrix of size \((m + n)\):

\[
\begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix}
\begin{bmatrix}
x \\
r
\end{bmatrix}
=
\begin{bmatrix}
b \\
0
\end{bmatrix}
\]
For inconsistent systems, must simultaneously refine both solution and residual.

(Björck, 1967): Least squares problem can be written as a linear system with square matrix of size \((m + n)\):

\[
\begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix}
\begin{bmatrix}
r \\
x
\end{bmatrix} =
\begin{bmatrix}
b \\
0
\end{bmatrix}
\]

Refinement proceeds as follows:

1. Compute "residuals"

\[
\begin{bmatrix}
f_i \\
g_i
\end{bmatrix} =
\begin{bmatrix}
b \\
0
\end{bmatrix} -
\begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix}
\begin{bmatrix}
r_i \\
x_i
\end{bmatrix} =
\begin{bmatrix}
b - r_i - Ax_i \\
a^T r_i
\end{bmatrix}
\]

2. Solve for corrections

\[
\begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix}
\begin{bmatrix}
\Delta r_i \\
\Delta x_i
\end{bmatrix} =
\begin{bmatrix}
f_i \\
g_i
\end{bmatrix}
\]

3. Update "solution":

\[
\begin{bmatrix}
r_{i+1} \\
x_{i+1}
\end{bmatrix} =
\begin{bmatrix}
r_i \\
x_i
\end{bmatrix} +
\begin{bmatrix}
\Delta r_i \\
\Delta x_i
\end{bmatrix}
\]

Least Squares Iterative Refinement
• For inconsistent systems, must simultaneously refine both solution and residual
• (Björck,1967): Least squares problem can be written as a linear system with square matrix of size \((m+n)\):

\[
\begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix}
\begin{bmatrix}
x \\
r
\end{bmatrix} =
\begin{bmatrix}
b \\
0
\end{bmatrix}
\Rightarrow \tilde{A}\tilde{x} = \tilde{b}
\]

• Refinement proceeds as follows:
1. Compute "residuals"

\[
\begin{bmatrix}
 f_i \\
g_i
\end{bmatrix} =
\begin{bmatrix}
b \\
0
\end{bmatrix} -
\begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix}
\begin{bmatrix}
x_i \\
r_i
\end{bmatrix} =
\begin{bmatrix}
b - r_i - Ax_i \\
-A^T r_i
\end{bmatrix}
\]

2. Solve for corrections

\[
\begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix}
\begin{bmatrix}
\Delta r_i \\
\Delta x_i
\end{bmatrix} =
\begin{bmatrix}
f_i \\
g_i
\end{bmatrix}
\]

3. Update "solution":

\[
\begin{bmatrix}
r_{i+1} \\
x_{i+1}
\end{bmatrix} =
\begin{bmatrix}
r_i \\
x_i
\end{bmatrix} +
\begin{bmatrix}
\Delta r_i \\
\Delta x_i
\end{bmatrix}
\]
Least Squares Iterative Refinement

• For inconsistent systems, must simultaneously refine both solution and residual
• (Björck, 1967): Least squares problem can be written as a linear system with square matrix of size \((m + n)\):
  \[
  \begin{bmatrix}
  I & A \\
  A^T & 0
  \end{bmatrix}
  \begin{bmatrix}
  r \\
  x
  \end{bmatrix}
  =
  \begin{bmatrix}
  b \\
  0
  \end{bmatrix}
  \quad \tilde{A}\tilde{x} = \tilde{b}
  \]

• Refinement proceeds as follows:
  1. Compute "residuals"

  \[
  \begin{bmatrix}
  f_i \\
  g_i
  \end{bmatrix}
  =
  \begin{bmatrix}
  b \\
  0
  \end{bmatrix}
  -
  \begin{bmatrix}
  I & A \\
  A^T & 0
  \end{bmatrix}
  \begin{bmatrix}
  r_i \\
  x_i
  \end{bmatrix}
  =
  \begin{bmatrix}
  b
  - r_i - Ax_i \\
  -A^T r_i
  \end{bmatrix}
  \quad \tilde{r}_i = \tilde{b} - \tilde{A}\tilde{x}_i
  \]

  2. Solve for corrections

  \[
  \begin{bmatrix}
  I & A \\
  A^T & 0
  \end{bmatrix}
  \begin{bmatrix}
  \Delta r_i \\
  \Delta x_i
  \end{bmatrix}
  =
  \begin{bmatrix}
  f_i \\
  g_i
  \end{bmatrix}
  \quad \tilde{A}\tilde{d}_i = \tilde{r}_i
  \]

  3. Update "solution":

  \[
  \begin{bmatrix}
  r_{i+1} \\
  x_{i+1}
  \end{bmatrix}
  =
  \begin{bmatrix}
  r_i \\
  x_i
  \end{bmatrix}
  +
  \begin{bmatrix}
  \Delta r_i \\
  \Delta x_i
  \end{bmatrix}
  \quad \tilde{x}_{i+1} = \tilde{x}_i + \tilde{d}_i
  \]
Least Squares Iterative Refinement

• For inconsistent systems, must simultaneously refine both solution and residual

• (Björck,1967): Least squares problem can be written as a linear system with square matrix of size \((m+n)\):

\[
\begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix}
\begin{bmatrix}
x \\
r
\end{bmatrix} =
\begin{bmatrix}
b \\
0
\end{bmatrix}
\]

\[\tilde{A}\tilde{x} = \tilde{b}\]

• Refinement proceeds as follows:

1. Compute "residuals"

\[
\begin{bmatrix}
f_i \\
g_i
\end{bmatrix} = \begin{bmatrix} b \\
0
\end{bmatrix} - \begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix}
\begin{bmatrix}
r_i \\
x_i
\end{bmatrix} = \begin{bmatrix}
b - r_i - Ax_i \\
-A^T r_i
\end{bmatrix}
\]

2. Solve for corrections

\[
\begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix}
\begin{bmatrix}
\Delta r_i \\
\Delta x_i
\end{bmatrix} =
\begin{bmatrix}
f_i \\
g_i
\end{bmatrix}
\]

3. Update "solution":

\[
\begin{bmatrix}
r_{i+1} \\
x_{i+1}
\end{bmatrix} =
\begin{bmatrix}
r_i \\
x_i
\end{bmatrix} +
\begin{bmatrix}
\Delta r_i \\
\Delta x_i
\end{bmatrix}
\]

\[\tilde{r}_i = \tilde{b} - \tilde{A}\tilde{x}_i\]

\[\tilde{A}d_i = \tilde{r}_i\]

\[\tilde{x}_{i+1} = \tilde{x}_i + d_i\]

Results for 3-precision IR for linear systems also applies to least squares problems
Least Squares Iterative Refinement

• To apply the existing analysis, we must consider:
  1. How is the condition number of $\tilde{A}$ related to the condition number of $A$?
  2. What are bounds on the forward and backward error in solving the correction equation $\tilde{A}d_i = \tilde{r}_i$?
• We now have a QR factorization rather than an LU factorization, and the augmented system has structure which can be exploited...
Augmented System Condition Number

- Result of Björck (1967):

The matrix

$$\tilde{A}_\alpha = \begin{bmatrix} \alpha I & A \\ A^T & 0 \end{bmatrix}$$

has condition number bounded by

$$\sqrt{2} \kappa_2(A) \leq \min_{\alpha} \kappa_2(\tilde{A}_\alpha) \leq 2 \kappa_2(A), \quad \max_{\alpha} \kappa_2(\tilde{A}_\alpha) > \kappa_2(A)^2$$

and $$\min_{\alpha} \kappa_2(\tilde{A}_\alpha)$$ is attained for $$\alpha = 2^{-\frac{1}{2}} \sigma_{\text{min}}(A)$$.
The matrix
\[ \tilde{A}_\alpha = \begin{bmatrix} \alpha I & A \\ A^T & 0 \end{bmatrix} \]
has condition number bounded by
\[ \sqrt{2} \kappa_2(A) \leq \min_\alpha \kappa_2(\tilde{A}_\alpha) \leq 2 \kappa_2(A), \quad \max_\alpha \kappa_2(\tilde{A}_\alpha) > \kappa_2(A)^2 \]
and \( \min_\alpha \kappa_2(\tilde{A}_\alpha) \) is attained for \( \alpha = 2^{-\frac{1}{2}} \sigma_{min}(A) \).

Scaling does not change the solution to least squares problem; further, if \( \alpha \) is a power of the machine base, it doesn't affect rounding errors.
\[ \Rightarrow \text{Safe to assume that } \kappa_2(\tilde{A}) \text{ is the same order of magnitude as } \kappa_2(A) \]
LS-IR in 3 precisions

Compute QR factorization $A = QR = [Q_1, Q_2] \begin{bmatrix} U \\ 0 \end{bmatrix}$ → precision $u_f$
LS-IR in 3 precisions

Compute QR factorization \( A = QR = [Q_1, Q_2] \begin{bmatrix} U \\ 0 \end{bmatrix} \) → precision \( u_f \)

Compute \( x_0 = U^{-1}Q_1^Tb, r_0 = b - Ax_0 \) → precision \( u \)
LS-IR in 3 precisions

Compute QR factorization $A = QR = [Q_1, Q_2] \begin{bmatrix} U \\ 0 \end{bmatrix}$ → precision $u_f$

Compute $x_0 = U^{-1}Q_1^T b$, $r_0 = b - Ax_0$ → precision $u$

For $i = 0, ...$

Compute residuals $\begin{bmatrix} f_i \\ g_i \end{bmatrix} = \begin{bmatrix} b - r_i - Ax_i \\ -A^T r_i \end{bmatrix}$ → precision $u_r$
LS-IR in 3 precisions

Compute QR factorization \( A = QR = [Q_1, Q_2] \begin{bmatrix} U \end{bmatrix} \)  
\( \Rightarrow \) precision \( u_f \)

Compute \( x_0 = U^{-1}Q_1^Tb, r_0 = b - Ax_0 \)  
\( \Rightarrow \) precision \( u \)

For \( i = 0, \ldots \)

Compute residuals \( \begin{bmatrix} f_i \\ g_i \end{bmatrix} = \begin{bmatrix} b - r_i - Ax_i \\ -A^T r_i \end{bmatrix} \)  
\( \Rightarrow \) precision \( u_r \)

Solve \( \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \Delta r_i \\ \Delta x_i \end{bmatrix} = \begin{bmatrix} f_i \\ g_i \end{bmatrix}, \) via

\( h = U^{-T} g_i \)

\( \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = [Q_1, Q_2]^T f_i \)

\( \Delta r_i = Q \begin{bmatrix} h \\ d_2 \end{bmatrix} \)

\( \Delta x_i = U^{-1}(d_1 - h) \)  
\( \Rightarrow \) precision \( u \)
LS-IR in 3 precisions

Compute QR factorization $A = QR = [Q_1, Q_2][U]_0$ → precision $u_f$

Compute $x_0 = U^{-1}Q_1^Tb, r_0 = b - Ax_0$ → precision $u$

For $i = 0, ...$

Compute residuals $\begin{bmatrix} f_i \\ g_i \end{bmatrix} = \begin{bmatrix} b - r_i - Ax_i \\ -A^Tr_i \end{bmatrix}$ → precision $u_r$

Solve $\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix}\begin{bmatrix} \Delta r_i \\ \Delta x_i \end{bmatrix} = \begin{bmatrix} f_i \\ g_i \end{bmatrix}$, via

$h = U^{-T}g_i$

$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = [Q_1, Q_2]^Tf_i$

$\Delta r_i = Q\begin{bmatrix} h \\ d_2 \end{bmatrix}$

$\Delta x_i = U^{-1}(d_1 - h)$

Update $x_{i+1} = x_i + \Delta x_i, r_{i+1} = r_i + \Delta r_i$ → precision $u$
Returning to IR3 Analysis...

The backward error for the correction solve:

\[(\tilde{A} + \Delta\tilde{A}) \hat{d}_i = \tilde{r}_i, \quad \|\Delta\tilde{A}\|_\infty \leq c_{m,n} u_f \|\tilde{A}\|_\infty\]
Returning to IR3 Analysis...

The backward error for the correction solve:

\[(\tilde{A} + \Delta \tilde{A}) \hat{d}_i = \tilde{r}_i, \quad \|\Delta \tilde{A}\|_\infty \leq c_{m,n} u_f \|\tilde{A}\|_\infty\]

\[u_s = u_f\]
Returning to IR3 Analysis...

The backward error for the correction solve:

\[
(A + \Delta A) \hat{d}_i = \hat{r}_i, \quad \|\Delta A\|_\infty \leq c_{m,n} u_f \|A\|_\infty
\]

1. \( \hat{d}_i = (I + u_s E_i) d_i, \quad u_s \|E_i\|_\infty < 1 \)

\[
u_s = u_f
\]

\[
u_s \|E_i\|_\infty \leq c_{m,n} u_f \|A\|_\infty
\]
Returning to IR3 Analysis...

The backward error for the correction solve:

$$(\tilde{A} + \Delta \tilde{A}) \hat{d}_i = \tilde{r}_i, \quad \|\Delta \tilde{A}\|_\infty \leq c_{m,n} u_f \|\tilde{A}\|_\infty$$

\[ u_s = u_f \]

1. \( \hat{d}_i = (I + u_s E_i) d_i, \quad u_s \|E_i\|_\infty < 1 \)

\[ u_s \|E_i\|_\infty \leq c_{m,n} u_f \|\tilde{A}\|_\infty \]

2. \( \|\hat{r}_i - A \hat{d}_i\|_\infty \leq u_s (c_1 \|A\|_\infty \|\hat{d}_i\|_\infty + c_2 \|\hat{r}_i\|_\infty) \)

\[ \max(c_1, c_2) \ u_s = O(u_f) \]
Returning to IR3 Analysis...

The backward error for the correction solve:

$$(\tilde{A} + \Delta\tilde{A}) \hat{d}_i = \hat{r}_i, \quad \|\Delta\tilde{A}\|_\infty \leq c_{m,n} u_f \|\tilde{A}\|_\infty$$

1. $\hat{d}_i = (I + u_s E_i) d_i, \quad u_s \|E_i\|_\infty < 1$

2. $\|\hat{r}_i - A\hat{d}_i\|_\infty \leq u_s (c_1 \|A\|_\infty \|\hat{d}_i\|_\infty + c_2 \|\hat{r}_i\|_\infty)$

3. $|\hat{r}_i - A\hat{d}_i| \leq u_s G_i |\hat{d}_i|$

4. $u_s = u_f$

$u_s \|E_i\|_\infty \leq c_{m,n} u_f \|\tilde{A}\|_\infty$

$\max(c_1, c_2) u_s = O(u_f)$

$u_s \|G_i\|_\infty = O(u_f \|\tilde{A}\|_\infty)$
Returning to IR3 Analysis...

The backward error for the correction solve:

\[(\tilde{A} + \Delta \tilde{A}) \hat{d}_i = \tilde{r}_i, \quad \| \Delta \tilde{A} \|_\infty \leq c_{m,n} u_f \| \tilde{A} \|_\infty\]

\[u_s = u_f\]

1. \[\hat{d}_i = (I + u_s E_i) d_i, \quad u_s \| E_i \|_\infty < 1\]

As long as \(\kappa_\infty(\tilde{A}) \leq u_f^{-1}\), expect convergence to limiting relative forward error

\[\frac{\| \tilde{x} - \hat{x} \|_\infty}{\| \tilde{x} \|_\infty} \approx u_r \text{cond}(\tilde{A}, \tilde{x}) + u\]

2. \[\| \hat{r}_i - A \hat{d}_i \|_\infty \leq u_s (c_1 \| A \|_\infty \| \hat{d}_i \|_\infty + c_2 \| \hat{r}_i \|_\infty)\]

3. \[| \hat{r}_i - A \hat{d}_i | \leq u_s G_i | \hat{d}_i |\]

\[\max(c_1, c_2) u_s = O(u_f)\]

\[u_s \| G_i \|_\infty = O(u_f \| \tilde{A} \|_\infty)\]
The backward error for the correction solve:

\[(\tilde{A} + \Delta \tilde{A}) \hat{d}_i = \tilde{r}_i, \quad \|\Delta \tilde{A}\|_\infty \leq c_{m,n} u_f \|\tilde{A}\|_\infty\]

1. \( \hat{d}_i = (I + u_s E_i) d_i, \quad u_s \|E_i\|_\infty < 1 \)

As long as \( \kappa_\infty(\tilde{A}) \leq u_f^{-1} \), expect convergence to limiting relative forward error

\[
\frac{\|\tilde{x} - \hat{x}\|_\infty}{\|\tilde{x}\|_\infty} \approx u_r \ \text{cond}(\tilde{A}, \tilde{x}) + u
\]

2. \( \|\hat{r}_i - A \hat{d}_i\|_\infty \leq u_s (c_1 \|A\|_\infty \|\hat{d}_i\|_\infty + c_2 \|\hat{r}_i\|_\infty) \)

3. \( |\hat{r}_i - A \hat{d}_i| \leq u_s G_i |\hat{d}_i| \)

As long as \( \kappa_\infty(\tilde{A}) \leq u_f^{-1} \), expect normwise and componentwise backward errors to be \( O(u) \)
Standard (QR-based) least squares IR with
\( u_f \): half, \( u \): single, \( u_r \): double

\[
A = \text{gallery}('\text{randsvd}', 100, 10, \text{kappa})
\]
\[
b = \text{randn}(100,1); \ b = b./\text{norm}(b)
\]
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)

Standard (QR-based) least squares IR with
  \( u_f \): half,  \( u \): single,  \( u_r \): double

\( \kappa = 1e+02 \)
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)

Standard (QR-based) least squares IR with
\( u_f \): half, \( u \): single, \( u_r \): double

\( \kappa = 1e+03 \)
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)

Standard (QR-based) least squares IR with

$\mathbf{u}_f$: half, $\mathbf{u}$: single, $\mathbf{u}_r$: double

$\kappa = 10^{4}$
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)

Standard (QR-based) least squares IR with
\( u_f \): half, \( u \): single, \( u_r \): double

\( \kappa = 1e+05 \)
GMRES-IR for Least Squares

• Similar to the linear system case, we can use a lower precision factorization for even more ill-conditioned problems if we improve the effective precision of the solver
• Similar to the linear system case, we can use a lower precision factorization for even more ill-conditioned problems if we improve the effective precision of the solver
• Again, don't want to compute an LU factorization of the augmented system
• How can we use existing QR factors to construct an effective and inexpensive preconditioner for the augmented system?
• Similar to the linear system case, we can use a lower precision factorization for even more ill-conditioned problems if we improve the effective precision of the solver

• Again, don't want to compute an LU factorization of the augmented system

• How can we use existing QR factors to construct an effective and inexpensive preconditioner for the augmented system?

• A couple possibilities:

1. Construct triangular factors using $R (= [U^T 0]^T)$ factor; use as split-preconditioner:

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \approx \begin{bmatrix} I & 0 \\ R^T & U^T \end{bmatrix} \begin{bmatrix} I & R \\ 0 & -U \end{bmatrix}$$
GMRES-IR for Least Squares

- Similar to the linear system case, we can use a lower precision factorization for even more ill-conditioned problems if we improve the effective precision of the solver.
- Again, don't want to compute an LU factorization of the augmented system.
- How can we use existing QR factors to construct an effective and inexpensive preconditioner for the augmented system?
- A couple possibilities:

1. **Construct triangular factors using** $R (= [U^T \ 0]^T)$ **factor; use as split-preconditioner:**

\[
\begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix} \approx \begin{bmatrix}
I & 0 \\
R^T & U^T
\end{bmatrix} \begin{bmatrix}
I & R \\
0 & -U
\end{bmatrix}
\]
GMRES-IR for Least Squares

• Similar to the linear system case, we can use a lower precision factorization for even more ill-conditioned problems if we improve the effective precision of the solver.

• Again, don't want to compute an LU factorization of the augmented system.

• How can we use existing QR factors to construct an effective and inexpensive preconditioner for the augmented system?

• A couple possibilities:

1. Construct triangular factors using $R (= [U^T 0]^T)$ factor; use as split-preconditioner:

\[
\begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix} \approx \begin{bmatrix}
I & 0 \\
R^T & U^T
\end{bmatrix} \begin{bmatrix}
I & R \\
0 & -U
\end{bmatrix}
\]

2. Use Hermitian/skew Hermitian splitting (HSS) preconditioning for saddlepoint systems; use left-preconditioned system matrix $M^{-1} \tilde{A}$ where

\[
M = (H + \alpha I)(S + \alpha I)
\]

\[
= \begin{bmatrix}
(\alpha + 1)I & 0 \\
0 & \alpha I
\end{bmatrix} \begin{bmatrix}
\alpha I & R \\
R^T & \alpha I
\end{bmatrix}
\]
GMRES-IR for Least Squares

• Similar to the linear system case, we can use a lower precision factorization for even more ill-conditioned problems if we improve the effective precision of the solver.

• Again, don't want to compute an LU factorization of the augmented system.

• How can we use existing QR factors to construct an effective and inexpensive preconditioner for the augmented system?

• A couple possibilities:

1. Construct triangular factors using \( R (= [U^T \ 0]^T) \) factor; use as split-preconditioner:

\[
\begin{bmatrix}
I & A \\
A^T & 0
\end{bmatrix} \approx \begin{bmatrix}
I & 0 \\
R^T & U^T
\end{bmatrix} \begin{bmatrix}
I & R \\
0 & -U
\end{bmatrix}
\]

2. Use Hermitian/skew Hermitian splitting (HSS) preconditioning for saddlepoint systems; use left-preconditioned system matrix \( M^{-1} \bar{A} \) where

\[
M = (H + \alpha I)(S + \alpha I)
= \begin{bmatrix}
(\alpha + 1)I & 0 \\
0 & \alpha I
\end{bmatrix} \begin{bmatrix}
\alpha I & R \\
R^T & \alpha I
\end{bmatrix}
\]
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)

GMRES-LSIR and "Standard" LSIR with
\( \mathbf{u}_f \): half, \( \mathbf{u} \): single, \( \mathbf{u}_r \): double

\( \kappa = 10^4 \)
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)

GMRES-LSIR and "Standard" LSIR with
\( \mathbf{u}_f \): half, \( \mathbf{u} \): single, \( \mathbf{u}_r \): double

\( \kappa = 1 \times 10^7 \)
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)

GMRES-LSIR and "Standard" LSIR with
\( u_f \): half, \( u \): single, \( u_r \): double

\[ \kappa = 1e+08 \]
The rise of multiprecision hardware

• Future machines will support a range of precisions: quarter, half, single, double, quad
The rise of multiprecision hardware

• Future machines will support a range of precisions: quarter, half, single, double, quad

• New, non-IEEE compliant floating point formats will appear in commercially-available hardware
  • e.g., bfloat16 (truncated 16-bit version of single precision) in upcoming Intel AI processors, Google Cloud TPUs, etc.
The rise of multiprecision hardware

• Future machines will support a range of precisions: quarter, half, single, double, quad

• New, non-IEEE compliant floating point formats will appear in commercially-available hardware
  • e.g., bfloat16 (truncated 16-bit version of single precision) in upcoming Intel AI processors, Google Cloud TPUs, etc.

• Lower-precision arithmetic is faster and more energy efficient, but the potential for its use depends heavily on the particular problem and algorithm
The rise of multiprecision hardware

- Future machines will support a range of precisions: quarter, half, single, double, quad

- New, non-IEEE compliant floating point formats will appear in commercially-available hardware
  - e.g., bfloat16 (truncated 16-bit version of single precision) in upcoming Intel AI processors, Google Cloud TPUs, etc.

- Lower-precision arithmetic is faster and more energy efficient, but the potential for its use depends heavily on the particular problem and algorithm

- As numerical analysts, we must determine when and where we can exploit lower-precision arithmetic to improve performance
Thank You!

carson@karlin.mff.cuni.cz
www.karlin.mff.cuni.cz/~carson/