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

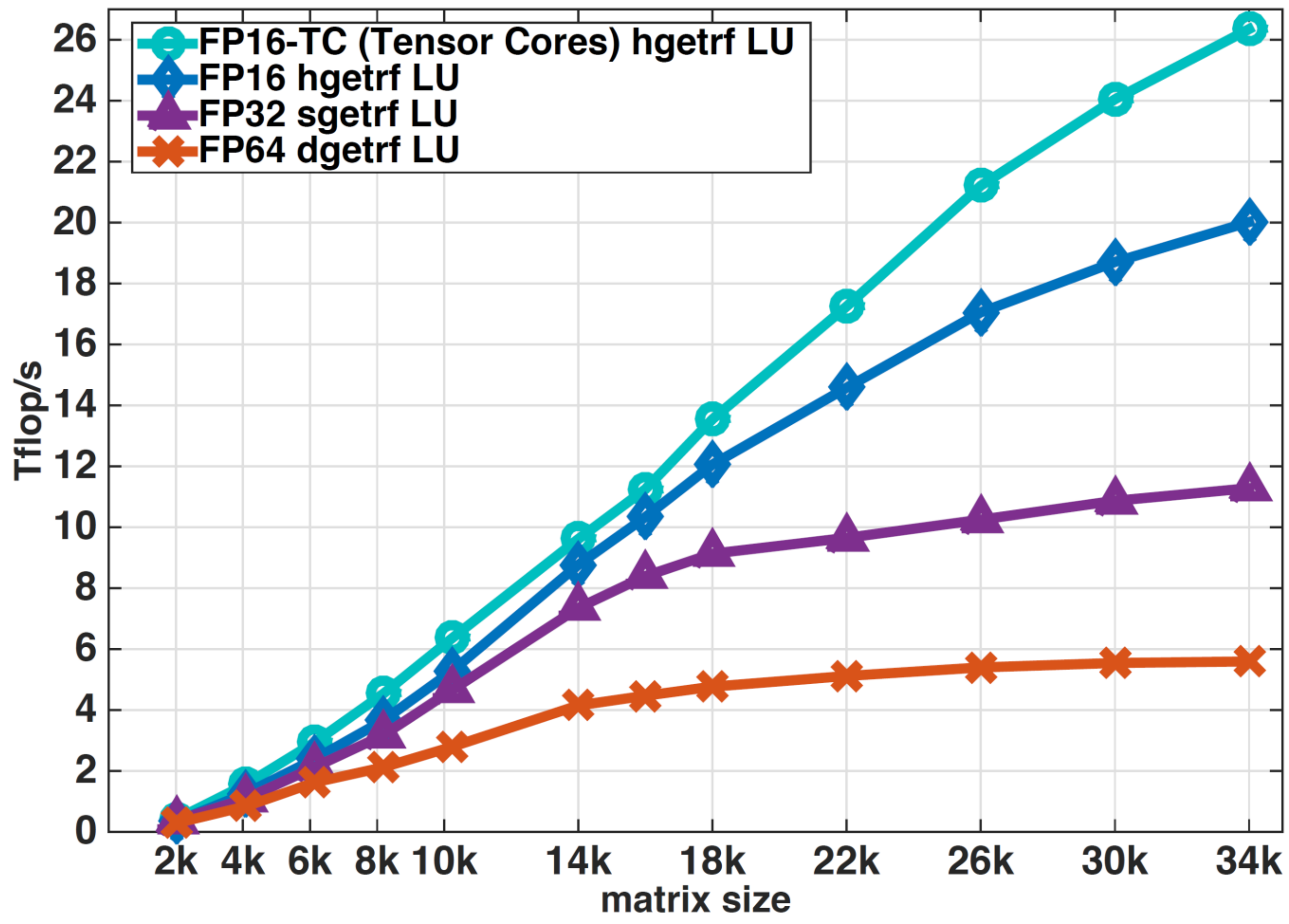
Hardware Support for Multiprecision Computation

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 $i = 0\colon \mathrm{maxit}$

$r_i = b - Ax_i$

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 $i = 0 : \mathrm{maxit}$

$r_i = b - Ax_i$ (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 $Ax = b$

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

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

As long as $\kappa_\infty(A) \le u^{-1}$,

- relative forward error is $O(u)$
- relative normwise and componentwise backward errors are $O(u)$

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

for $i = 0:\mathrm{maxit}$

$r_i = b - Ax_i$ (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 $Ax = b$

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

for $i = 0: \mathrm{maxit}$

$r_i = b - Ax_i$ (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 $O(u)\mathrm{cond}(A, x)$
- relative normwise and componentwise backward errors are $O(u)$

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

for $i = 0 : \mathrm{maxit}$

$r_i = b - Ax_i$ (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 $Ax = b$

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

for $i = 0: \mathrm{maxit}$

$r_i = b - Ax_i$ (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)

"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 $\kappa_\infty(A) \le u^{-1/2}$,

- relative forward error is $O(u)\mathrm{cond}(A, x)$
- relative normwise and componentwise backward errors are $O(u)$

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

for $i = 0{:}\,\mathrm{maxit}$

$r_i = b - Ax_i$ (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)

"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?

⇒ **3-precision iterative refinement**

u_f = factorization precision, u = working precision, u_r = 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?

$\Rightarrow$ **3-precision iterative refinement**

u_f = factorization precision, u = working precision, u_r = residual precision

$$u_f \geq u \geq u_r$$

- New analysis **generalizes** existing types of IR:

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

Traditional	$u_f = u, u_r = u^2$
Fixed precision	$u_f = u = u_r$
Lower precision factorization	$u_f^2 = u = u_r$

(and **improves** upon existing analyses in some cases)

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 = factorization precision, u = working precision, u_r = residual precision

$$u_f \geq u \geq u_r$$

- New analysis **generalizes** existing types of IR:

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

Traditional	$u_f = u, u_r = u^2$
Fixed precision	$u_f = u = u_r$
Lower precision factorization	$u_f^2 = u = u_r$

(and **improves** upon existing analyses in some cases)

- Enables **new** types of IR: (half, single, double), (half, single, quad), (half, double, quad), etc.

Key Analysis Innovations I

Obtain tighter upper bounds:

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

Key Analysis Innovations I

Obtain tighter upper bounds:

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

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

Key Analysis Innovations I

Obtain tighter upper bounds:

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

Define μ_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

$$\frac{\|r_i\|}{\|A\|\|\hat{x}_i\|} \approx u \ll \frac{\|x-\hat{x}_i\|}{\|x\|} \longrightarrow \boxed{\mu_i \ll 1}$$

Key Analysis Innovations I

Obtain tighter upper bounds:

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

Define μ_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

$$\frac{\|r_i\|}{\|A\|\|\hat{x}_i\|} \approx u \ll \frac{\|x-\hat{x}_i\|}{\|x\|} \longrightarrow \boxed{\mu_i \ll 1}$$

But close to convergence,

$$\|r_i\| \approx \|A\|\|x-\hat{x}_i\| \longrightarrow \boxed{\mu_i \approx 1}$$

Key Analysis Innovations I

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

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

Key Analysis Innovations I

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

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

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

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

Key Analysis Innovations I

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

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

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

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

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

Key Analysis Innovations I

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

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

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

where $P_k = U_k U_k^T, U_k = [u_{n+1-k}, \dots, u_n]$

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

- $\mu_i^{(2)} \ll 1$ if r_i contains significant component in $\mathrm{span}(U_k)$ for any k s.t. $\sigma_{n+1-k} \approx \sigma_n$

Key Analysis Innovations I

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

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

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

where $P_k = U_k U_k^T, U_k = [u_{n+1-k}, \dots, u_n]$

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

- $\mu_i^{(2)} \ll 1$ 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

Key Analysis Innovations I

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

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

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

where $P_k = U_k U_k^T, U_k = [u_{n+1-k}, \dots, u_n]$

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

- $\mu_i^{(2)} \ll 1$ if r_i contains significant component in $\mathrm{span}(U_k)$ 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

Key Analysis Innovations II

Allow for general solver:

Let $\boldsymbol{u_s}$ be the *effective precision* of the solve, with $\boldsymbol{u} \leq \boldsymbol{u_s} \leq \boldsymbol{u_f}$

Key Analysis Innovations II

Allow for general solver:

Let u_s be the *effective precision* of the solve, with $u \le u_s \le u_f$

example: LU solve:
$u_s = u_f$

Key Analysis Innovations II

Allow for general solver:

Let $\boldsymbol{u_s}$ be the *effective precision* of the solve, with $\boldsymbol{u} \leq \boldsymbol{u_s} \leq \boldsymbol{u_f}$

example: LU solve:

$\boldsymbol{u_s} = \boldsymbol{u_f}$

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

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

   → normwise relative forward error is bounded by multiple of u_s and is less than 1

Key Analysis Innovations II

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

example: LU solve:
$\boldsymbol{u_s} = \boldsymbol{u_f}$

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

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

   $\rightarrow$ normwise relative forward error is bounded by multiple of u_s and is less than 1

$$\boldsymbol{u_s} \|E_i\|_\infty \leq 3n\boldsymbol{u_f} \left\| |A^{-1}||\hat{L}||\hat{U}| \right\|_\infty$$

Key Analysis Innovations II

Allow for general solver:

Let $\boldsymbol{u_s}$ be the *effective precision* of the solve, with $\boldsymbol{u} \leq \boldsymbol{u_s} \leq \boldsymbol{u_f}$

example: LU solve:

$\boldsymbol{u_s} = \boldsymbol{u_f}$

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

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

   → normwise relative forward error is bounded by multiple of u_s and is less than 1

$$\boldsymbol{u_s} \|E_i\|_\infty \leq 3n\boldsymbol{u_f} \left\| |A^{-1}||\hat{L}||\hat{U}| \right\|_\infty$$

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

   → normwise relative backward error is at most $\max(c_1, c_2)\, u_s$

Key Analysis Innovations II

Allow for general solver:

Let $\boldsymbol{u_s}$ be the *effective precision* of the solve, with $\boldsymbol{u} \leq \boldsymbol{u_s} \leq \boldsymbol{u_f}$

example: LU solve:
$\boldsymbol{u_s} = \boldsymbol{u_f}$

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

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

   → normwise relative forward error is bounded by multiple of u_s and is less than 1

$$\boldsymbol{u_s} \|E_i\|_\infty \leq 3n\boldsymbol{u_f} \left\| |A^{-1}| |\hat{L}| |\hat{U}| \right\|_\infty$$

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

   → normwise relative backward error is at most $\max(c_1, c_2)\, u_s$

$$\max(c_1, c_2)\, \boldsymbol{u_s} \leq \frac{3n\boldsymbol{u_f} \left\| |\hat{L}| |\hat{U}| \right\|_\infty}{\|A\|_\infty}$$

Key Analysis Innovations II

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

example: LU solve:
$\boldsymbol{u_s} = \boldsymbol{u_f}$

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

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

   → normwise relative forward error is bounded by multiple of u_s and is less than 1

$\boldsymbol{u_s} \|E_i\|_\infty \leq 3n\boldsymbol{u_f} \left\| |A^{-1}| |\hat{L}| |\hat{U}| \right\|_\infty$

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

   → normwise relative backward error is at most $\max(c_1, c_2)\, u_s$

$\max(c_1, c_2)\, \boldsymbol{u_s} \leq \dfrac{3n\boldsymbol{u_f} \left\| |\hat{L}| |\hat{U}| \right\|_\infty}{\|A\|_\infty}$

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

   → componentwise relative backward error is bounded by a multiple of u_s

Key Analysis Innovations II

Allow for general solver:

Let u_s be the *effective precision* of the solve, with $u \le u_s \le u_f$

example: LU solve:
$u_s = u_f$

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

$$u_s \|E_i\|_\infty \le 3n u_f \left\| |A^{-1}| |\hat{L}| |\hat{U}| \right\|_\infty$$

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

   → normwise relative backward error is at most $\max(c_1, c_2)\, u_s$

$$\max(c_1, c_2)\, u_s \le \frac{3n u_f \left\| |\hat{L}| |\hat{U}| \right\|_\infty}{\|A\|_\infty}$$

3. $\left|\hat{r}_i - A\hat{d}_i\right| \le u_s G_i |\hat{d}_i|$

   → componentwise relative backward error is bounded by a multiple of u_s

$$u_s \|G_i\|_\infty \le 3n u_f \left\| |\hat{L}| |\hat{U}| \right\|_\infty$$

Key Analysis Innovations II

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

example: LU solve:
$u_s = u_f$

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

$$u_s \|E_i\|_\infty \le 3n u_f \left\| |A^{-1}||\hat{L}||\hat{U}| \right\|_\infty$$

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

   → normwise relative backward error is at most $\max(c_1, c_2)\, u_s$

$$\max(c_1, c_2)\, u_s \le \frac{3n u_f \left\| |\hat{L}||\hat{U}| \right\|_\infty}{\|A\|_\infty}$$

3. $\left| \hat{r}_i - A\hat{d}_i \right| \le u_s G_i |\hat{d}_i|$

   → componentwise relative backward error is bounded by a multiple of u_s

$$u_s \|G_i\|_\infty \le 3n u_f \left\| |\hat{L}||\hat{U}| \right\|_\infty$$

E_i, c_1, c_2, and G_i depend on A, $\hat{r}_i$, n, and u_s

Forward Error for IR3

- Three precisions:
  - u_f: factorization precision
  - u: working precision
  - u_r: residual computation precision

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

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

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

Forward Error for IR3

- Three precisions:
  - u_f: factorization precision
  - u: working precision
  - u_r: residual computation precision

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

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

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

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

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

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

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

Forward Error for IR3

- Three precisions:
  - u_f: factorization precision
  - u: working precision
  - u_r: residual computation precision

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

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

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

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

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

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

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

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

Normwise Backward Error for IR3

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

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

$$\phi_i \equiv (c_1 \kappa_\infty(A) + c_2)\boldsymbol{u_s}$$

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

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

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 $\approx 10^{-34}$

				Backward error		
u_f	u	u_r	max $\kappa_\infty(A)$	norm	comp	Forward error
H	S	S	10^4	10^{-8}	10^{-8}	$\text{cond}(A,x) \cdot 10^{-8}$
H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
H	D	D	10^4	10^{-16}	10^{-16}	$\text{cond}(A,x) \cdot 10^{-16}$
H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
S	S	S	10^8	10^{-8}	10^{-8}	$\text{cond}(A,x) \cdot 10^{-8}$
S	S	D	10^8	10^{-8}	10^{-8}	10^{-8}
S	D	D	10^8	10^{-16}	10^{-16}	$\text{cond}(A,x) \cdot 10^{-16}$
S	D	Q	10^8	10^{-16}	10^{-16}	10^{-16}

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 $\approx 10^{-34}$

					Backward error		
	u_f	u	u_r	max $\kappa_\infty(A)$	norm	comp	Forward error
LP fact.	H	S	S	10^4	10^{-8}	10^{-8}	$\text{cond}(A,x) \cdot 10^{-8}$
	H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
LP fact.	H	D	D	10^4	10^{-16}	10^{-16}	$\text{cond}(A,x) \cdot 10^{-16}$
	H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
	S	S	S	10^8	10^{-8}	10^{-8}	$\text{cond}(A,x) \cdot 10^{-8}$
	S	S	D	10^8	10^{-8}	10^{-8}	10^{-8}
LP fact.	S	D	D	10^8	10^{-16}	10^{-16}	$\text{cond}(A,x) \cdot 10^{-16}$
	S	D	Q	10^8	10^{-16}	10^{-16}	10^{-16}

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 $\approx 10^{-34}$

	u_f	u	u_r	max $\kappa_\infty(A)$	Backward error norm	Backward error comp	Forward error
LP fact.	H	S	S	10^4	10^{-8}	10^{-8}	$\mathrm{cond}(A,x) \cdot 10^{-8}$
	H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
LP fact.	H	D	D	10^4	10^{-16}	10^{-16}	$\mathrm{cond}(A,x) \cdot 10^{-16}$
	H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
Fixed	S	S	S	10^8	10^{-8}	10^{-8}	$\mathrm{cond}(A,x) \cdot 10^{-8}$
	S	S	D	10^8	10^{-8}	10^{-8}	10^{-8}
LP fact.	S	D	D	10^8	10^{-16}	10^{-16}	$\mathrm{cond}(A,x) \cdot 10^{-16}$
	S	D	Q	10^8	10^{-16}	10^{-16}	10^{-16}

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 $\approx 10^{-34}$

	u_f	u	u_r	$\max \kappa_\infty(A)$	Backward error norm	Backward error comp	Forward error
LP fact.	H	S	S	10^4	10^{-8}	10^{-8}	$\text{cond}(A,x)\cdot 10^{-8}$
	H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
LP fact.	H	D	D	10^4	10^{-16}	10^{-16}	$\text{cond}(A,x)\cdot 10^{-16}$
	H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
Fixed	S	S	S	10^8	10^{-8}	10^{-8}	$\text{cond}(A,x)\cdot 10^{-8}$
Trad.	S	S	D	10^8	10^{-8}	10^{-8}	10^{-8}
LP fact.	S	D	D	10^8	10^{-16}	10^{-16}	$\text{cond}(A,x)\cdot 10^{-16}$
	S	D	Q	10^8	10^{-16}	10^{-16}	10^{-16}

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 $\approx 10^{-34}$

	u_f	u	u_r	max $\kappa_\infty(A)$	Backward error norm	Backward error comp	Forward error
LP fact.	H	S	S	10^4	10^{-8}	10^{-8}	$\mathrm{cond}(A,x)\cdot 10^{-8}$
New	H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
LP fact.	H	D	D	10^4	10^{-16}	10^{-16}	$\mathrm{cond}(A,x)\cdot 10^{-16}$
New	H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
Fixed	S	S	S	10^8	10^{-8}	10^{-8}	$\mathrm{cond}(A,x)\cdot 10^{-8}$
Trad.	S	S	D	10^8	10^{-8}	10^{-8}	10^{-8}
LP fact.	S	D	D	10^8	10^{-16}	10^{-16}	$\mathrm{cond}(A,x)\cdot 10^{-16}$
New	S	D	Q	10^8	10^{-16}	10^{-16}	10^{-16}

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 $\approx 10^{-34}$

	u_f	u	u_r	max $\kappa_\infty(A)$	Backward error norm	Backward error comp	Forward error
LP fact.	H	S	S	10^4	10^{-8}	10^{-8}	$\mathrm{cond}(A,x)\cdot 10^{-8}$
New	H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
LP fact.	H	D	D	10^4	10^{-16}	10^{-16}	$\mathrm{cond}(A,x)\cdot 10^{-16}$
New	H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
Fixed	S	S	S	10^8	10^{-8}	10^{-8}	$\mathrm{cond}(A,x)\cdot 10^{-8}$
Trad.	S	S	D	10^8	10^{-8}	10^{-8}	10^{-8}
LP fact.	S	D	D	10^8	10^{-16}	10^{-16}	$\mathrm{cond}(A,x)\cdot 10^{-16}$
New	S	D	Q	10^8	10^{-16}	10^{-16}	10^{-16}

$\Rightarrow$ Benefit of IR3 vs. "LP fact.": no $\mathrm{cond}(A,x)$ 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 $\approx 10^{-34}$

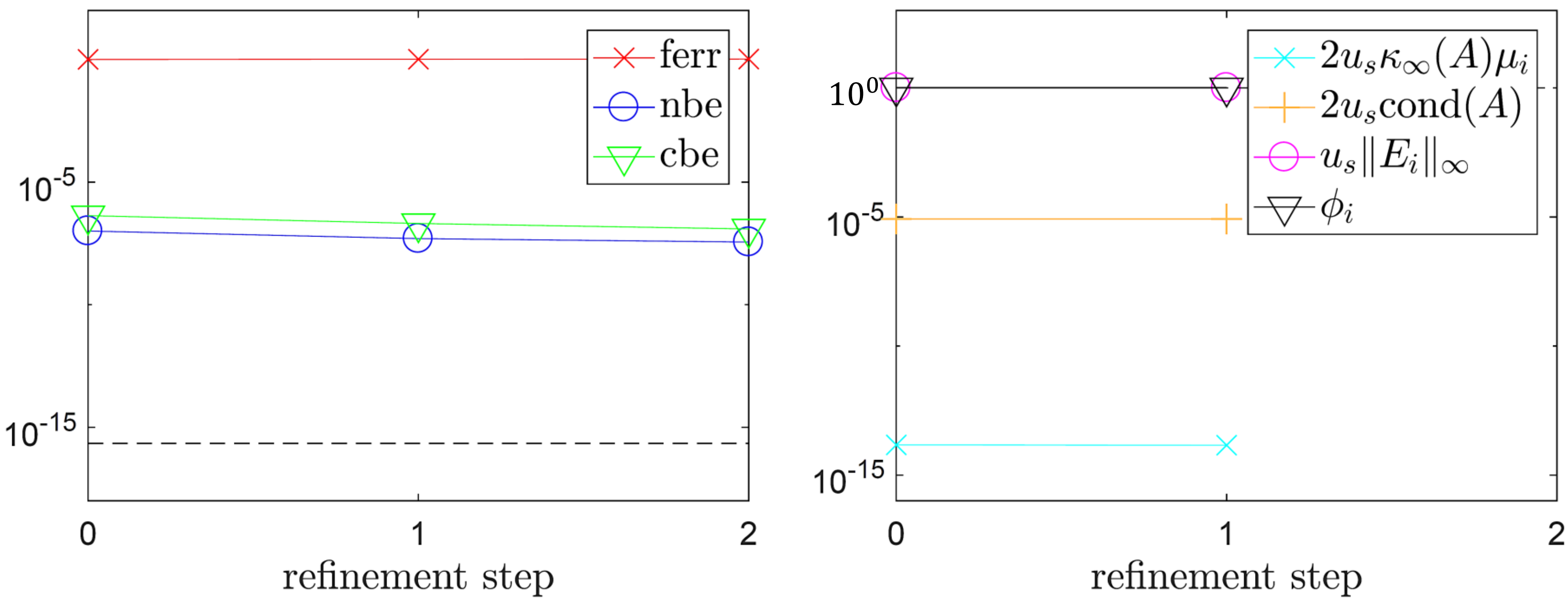
	u_f	u	u_r	max $\kappa_\infty(A)$	Backward error norm	Backward error comp	Forward error
LP fact.	H	S	S	10^4	10^{-8}	10^{-8}	$\text{cond}(A,x)\cdot 10^{-8}$
New	H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
LP fact.	H	D	D	10^4	10^{-16}	10^{-16}	$\text{cond}(A,x)\cdot 10^{-16}$
New	H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
Fixed	S	S	S	10^8	10^{-8}	10^{-8}	$\text{cond}(A,x)\cdot 10^{-8}$
Trad.	S	S	D	10^8	10^{-8}	10^{-8}	10^{-8}
LP fact.	S	D	D	10^8	10^{-16}	10^{-16}	$\text{cond}(A,x)\cdot 10^{-16}$
New	S	D	Q	10^8	10^{-16}	10^{-16}	10^{-16}

$\Rightarrow$ 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, $\mathrm{cond}(A,x) \approx$ 5e9

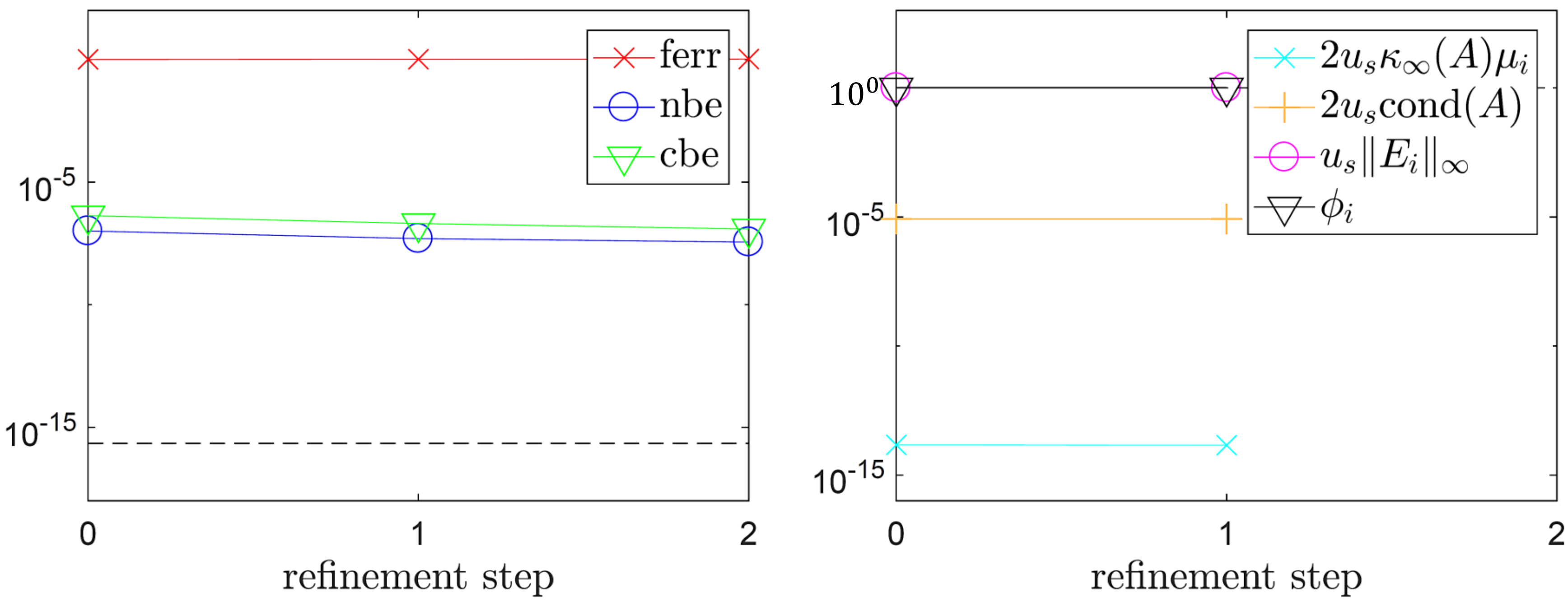
Standard (LU-based) IR with u_f: single, u: double, u_r: double



```
A = gallery('randsvd', 100, 1e9, 2)
b = randn(100,1)
```

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

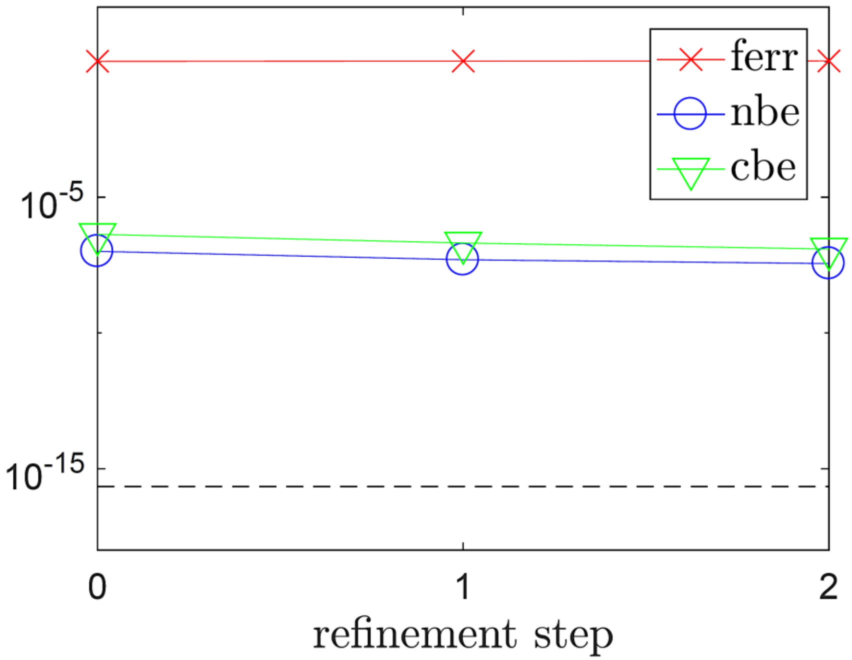
Standard (LU-based) IR with u_f: single, u: double, u_r: quad

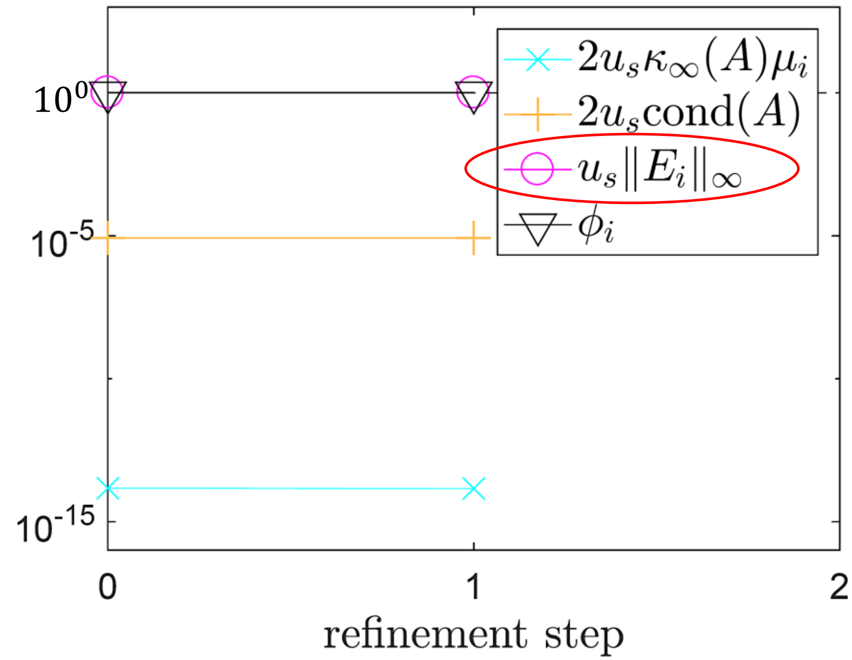


```
A = gallery('randsvd', 100, 1e9, 2)
b = randn(100,1)
```

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

Standard (LU-based) IR with u_f: single, u: double, u_r: quad

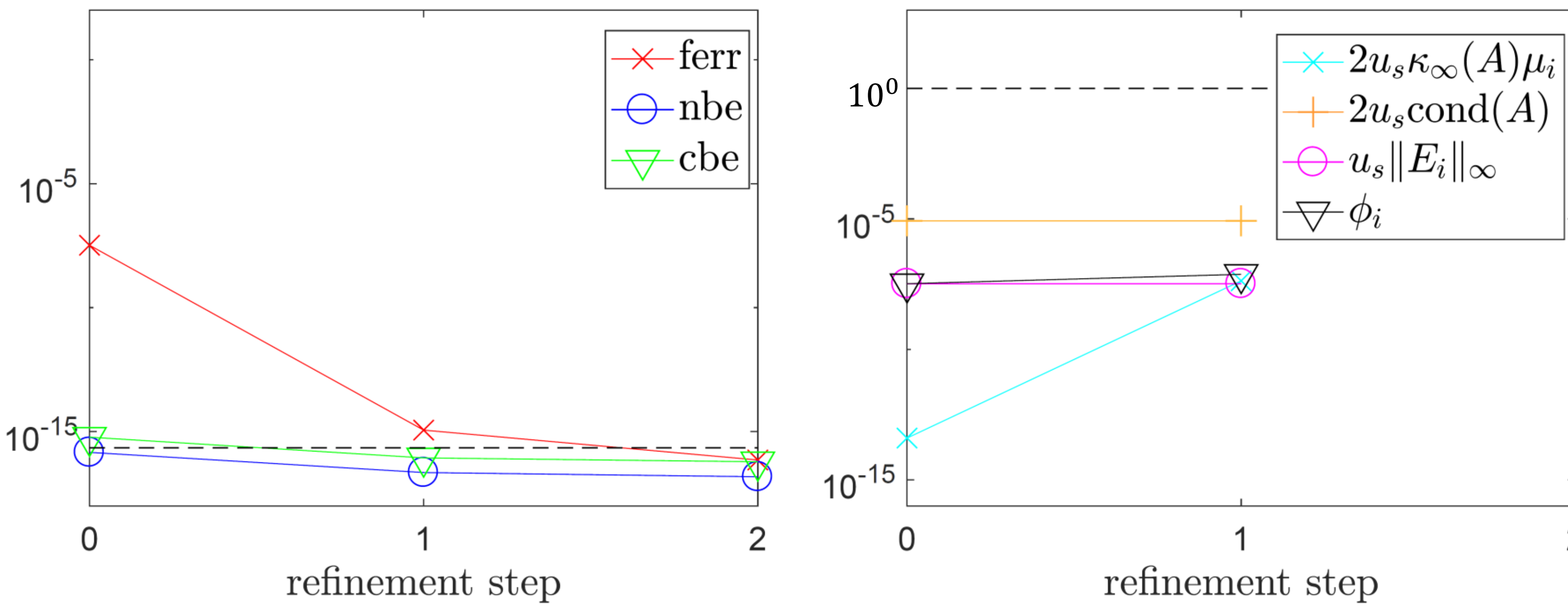




```
A = gallery('randsvd', 100, 1e9, 2)
b = randn(100,1)
```

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

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

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

GMRES-Based Iterative Refinement

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

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

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

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

- To compute the updates d_i, apply GMRES to $\overbrace{\hat{U}^{-1}\hat{L}^{-1}A}^{\tilde{A}}d_i = \overbrace{\hat{U}^{-1}\hat{L}^{-1}r_i}^{\tilde{r}_i}$

GMRES-Based Iterative Refinement

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

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

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 $\overbrace{\hat{U}^{-1}\hat{L}^{-1}A}^{\tilde{A}}d_i = \overbrace{\hat{U}^{-1}\hat{L}^{-1}r_i}^{\tilde{r}_i}$

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

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

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

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

- To compute the updates d_i, apply GMRES to $\overbrace{\hat{U}^{-1}\hat{L}^{-1}A}^{\tilde{A}}d_i = \overbrace{\hat{U}^{-1}\hat{L}^{-1}r_i}^{\tilde{r}_i}$

Solve $Ax_0 = b$ by LU factorization

for $i = 0\colon \text{maxit}$

$r_i = b - Ax_i$

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

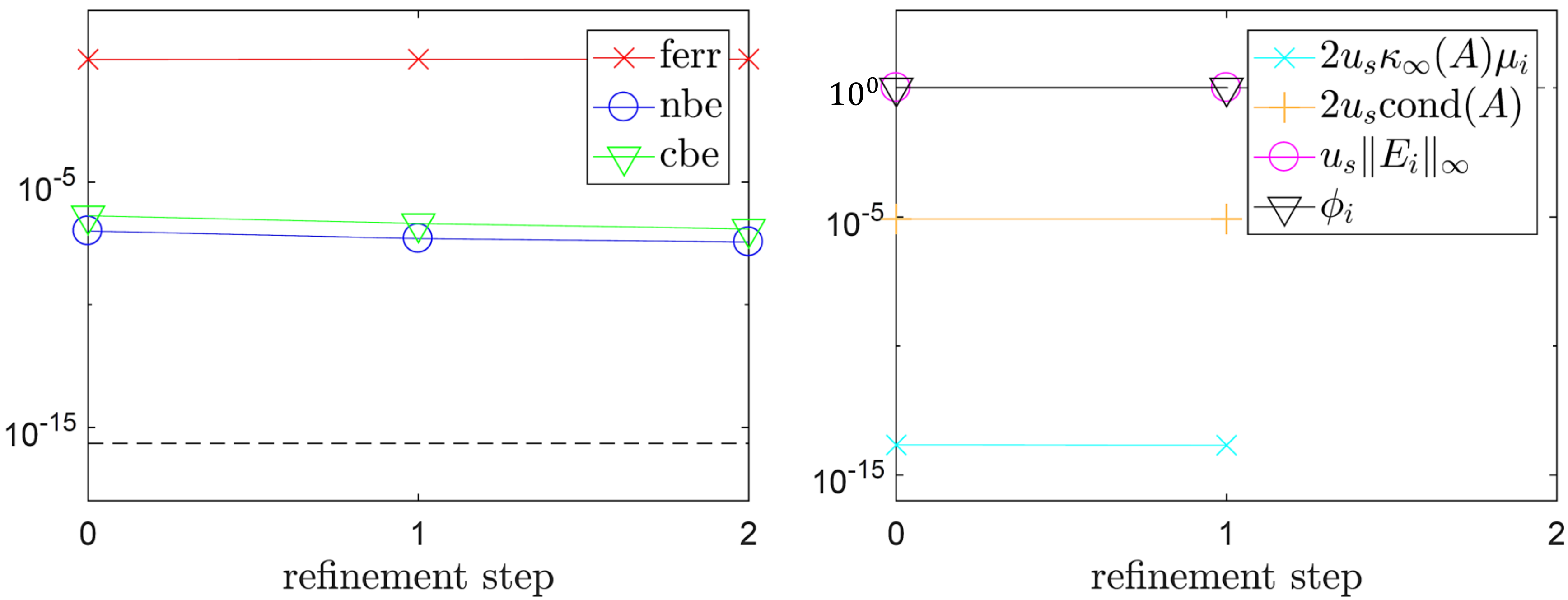
$x_{i+1} = x_i + d_i$

$u_s = u$

```
A = gallery('randsvd', 100, 1e9, 2)
b = randn(100,1)
```

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

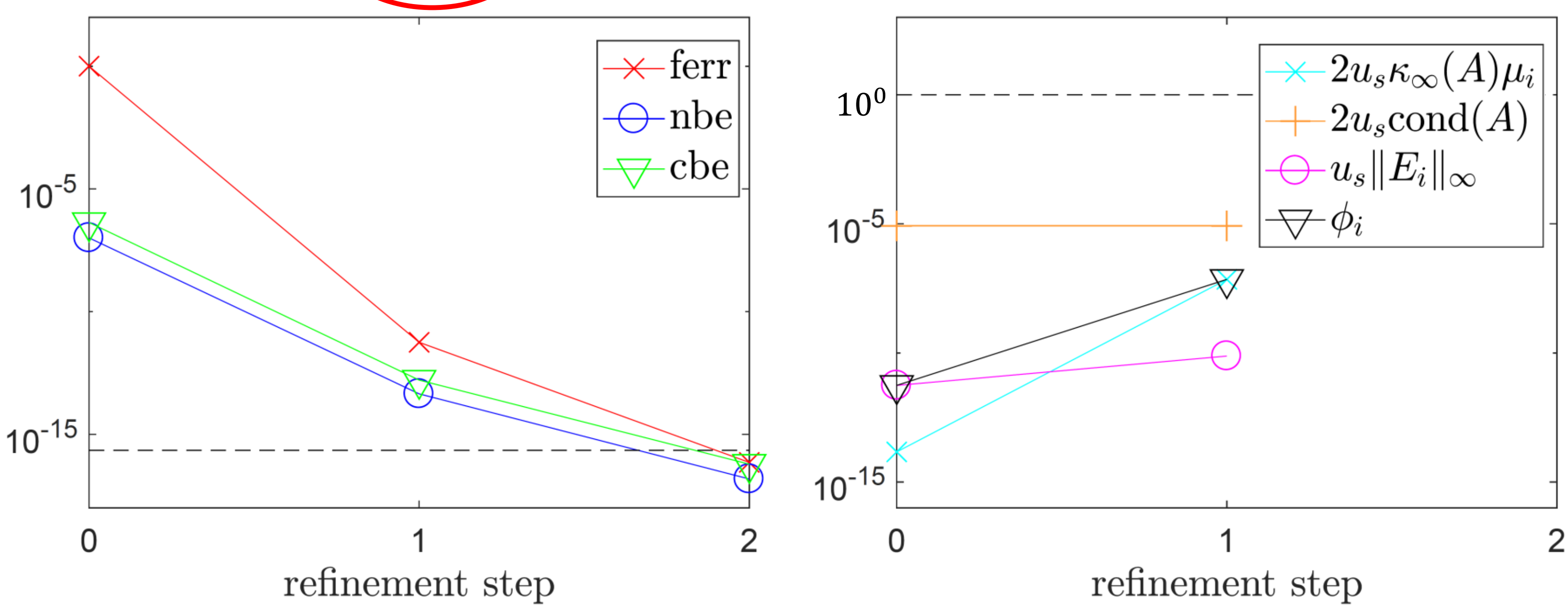
Standard (LU-based) IR with u_f: single, u: double, u_r: quad



```
A = gallery('randsvd', 100, 1e9, 2)
b = randn(100,1)
```

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

GMRES-IR with u_f: single, u: double, u_r: quad



GMRES-IR: Summary

Benefits of GMRES-IR:

	u_f	u	u_r	max $\kappa_\infty(A)$	Backward error norm	Backward error comp	Forward error
LU-IR	H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
GMRES-IR	H	S	D	10^8	10^{-8}	10^{-8}	10^{-8}
LU-IR	S	D	Q	10^8	10^{-16}	10^{-16}	10^{-16}
GMRES-IR	S	D	Q	10^{16}	10^{-16}	10^{-16}	10^{-16}
LU-IR	H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
GMRES-IR	H	D	Q	10^{12}	10^{-16}	10^{-16}	10^{-16}

GMRES-IR: Summary

Benefits of GMRES-IR:

	u_f	u	u_r	max $\kappa_\infty(A)$	Backward error norm	Backward error comp	Forward error
LU-IR	H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
GMRES-IR	H	S	D	10^8	10^{-8}	10^{-8}	10^{-8}
LU-IR	S	D	Q	10^8	10^{-16}	10^{-16}	10^{-16}
GMRES-IR	S	D	Q	10^{16}	10^{-16}	10^{-16}	10^{-16}
LU-IR	H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
GMRES-IR	H	D	Q	10^{12}	10^{-16}	10^{-16}	10^{-16}

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

GMRES-IR: Summary

Benefits of GMRES-IR:

	u_f	u	u_r	max $\kappa_\infty(A)$	Backward error norm	Backward error comp	Forward error
LU-IR	H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
GMRES-IR	H	S	D	10^8	10^{-8}	10^{-8}	10^{-8}
LU-IR	S	D	Q	10^8	10^{-16}	10^{-16}	10^{-16}
GMRES-IR	S	D	Q	10^{16}	10^{-16}	10^{-16}	10^{-16}
LU-IR	H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
GMRES-IR	H	D	Q	10^{12}	10^{-16}	10^{-16}	10^{-16}

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

$$\kappa_\infty(A) \le u^{-1/2}\, u_f^{-1}$$

GMRES-IR: Summary

Benefits of GMRES-IR:

	u_f	u	u_r	max $\kappa_\infty(A)$	Backward error norm	Backward error comp	Forward error
LU-IR	H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
GMRES-IR	H	S	D	10^8	10^{-8}	10^{-8}	10^{-8}
LU-IR	S	D	Q	10^8	10^{-16}	10^{-16}	10^{-16}
GMRES-IR	S	D	Q	10^{16}	10^{-16}	10^{-16}	10^{-16}
LU-IR	H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
GMRES-IR	H	D	Q	10^{12}	10^{-16}	10^{-16}	10^{-16}

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

GMRES-IR: Summary

Benefits of GMRES-IR:

	u_f	u	u_r	max $\kappa_\infty(A)$	Backward error norm	Backward error comp	Forward error
LU-IR	H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
GMRES-IR	H	S	D	10^8	10^{-8}	10^{-8}	10^{-8}
LU-IR	S	D	Q	10^8	10^{-16}	10^{-16}	10^{-16}
GMRES-IR	S	D	Q	10^{16}	10^{-16}	10^{-16}	10^{-16}
LU-IR	H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
GMRES-IR	H	D	Q	10^{12}	10^{-16}	10^{-16}	10^{-16}

Try IR3! MATLAB codes available at: https://github.com/eccarson/ir3

Comments and Caveats

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

Comments and Caveats

- Convergence tolerance τ 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 τ 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 τ 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.

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

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.

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

- 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 $(m + n)$:

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

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 $(m+n)$:

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

- 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 $(m+n)$:

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \qquad \tilde{A}\tilde{x} = \tilde{b}$$

- 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":

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

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 $(m+n)$:

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \qquad \tilde{A}\tilde{x} = \tilde{b}$$

- Refinement proceeds as follows:

1. Compute "residuals"

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

2. Solve for corrections

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \Delta r_i \\ \Delta x_i \end{bmatrix} = \begin{bmatrix} f_i \\ g_i \end{bmatrix} \qquad \tilde{A} d_i = \tilde{r}_i$$

3. Update "solution":

$$\begin{bmatrix} r_{i+1} \\ x_{i+1} \end{bmatrix} = \begin{bmatrix} r_i \\ x_i \end{bmatrix} + \begin{bmatrix} \Delta r_i \\ \Delta x_i \end{bmatrix} \qquad \tilde{x}_{i+1} = \tilde{x}_i + d_i$$

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 $(m+n)$:

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \qquad \tilde{A}\tilde{x} = \tilde{b}$$

- 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

$$\sqrt{2}\kappa_2(A) \le \min_\alpha \kappa_2\left(\tilde{A}_\alpha\right) \le 2\kappa_2(A), \qquad \max_\alpha \kappa_2\left(\tilde{A}_\alpha\right) > \kappa_2(A)^2$$

and $\min_\alpha \kappa_2\left(\tilde{A}_\alpha\right)$ is attained for $\alpha = 2^{-\frac{1}{2}}\sigma_{min}(A)$.

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

  $$\sqrt{2}\kappa_2(A) \le \min_\alpha \kappa_2\left(\tilde{A}_\alpha\right) \le 2\kappa_2(A), \qquad \max_\alpha \kappa_2\left(\tilde{A}_\alpha\right) > \kappa_2(A)^2$$

  and $\min_\alpha \kappa_2\left(\tilde{A}_\alpha\right)$ is attained for $\alpha = 2^{-\frac{1}{2}}\sigma_{min}(A)$.

- Scaling does not change the solution to least squares problem; further, if α is a power of the machine base, it doesn't affect rounding errors

  $\Rightarrow$ Safe to assume that $\kappa_2(\tilde{A})$ is the same order of magnitude as $\kappa_2(A)$

LS-IR in 3 precisions

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

LS-IR in 3 precisions

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

Compute $x_0 = U^{-1}Q_1^T b, r_0 = b - Ax_0 \longrightarrow$ 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 $i = 0, \ldots$

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

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

For $i = 0, \dots$

Compute residuals $\begin{bmatrix} f_i \\ g_i \end{bmatrix} = \begin{bmatrix} b - r_i - Ax_i \\ -A^T r_i \end{bmatrix}$ $\longrightarrow$ 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

$$\left.\begin{aligned} h &= U^{-T} g_i \\ \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} &= [Q_1, Q_2]^T f_i \\ \Delta r_i &= Q\begin{bmatrix} h \\ d_2 \end{bmatrix} \\ \Delta x_i &= U^{-1}(d_1 - h) \end{aligned}\right\} \text{precision } u$$

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 $i = 0, \ldots$

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

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

$$\left.\begin{aligned} h &= U^{-T} g_i \\ \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} &= [Q_1, Q_2]^T f_i \\ \Delta r_i &= Q\begin{bmatrix} h \\ d_2 \end{bmatrix} \\ \Delta x_i &= U^{-1}(d_1 - h) \end{aligned}\right\} \text{precision } u$$

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:

$$\left(\tilde{A} + \Delta\tilde{A}\right)\hat{d}_i = \tilde{r}_i, \qquad \|\Delta\tilde{A}\|_\infty \le c_{m,n}\boldsymbol{u_f}\|\tilde{A}\|_\infty$$

Returning to IR3 Analysis...

The backward error for the correction solve:

$$\left(\tilde{A} + \Delta\tilde{A}\right) \hat{d}_i = \tilde{r}_i, \qquad \|\Delta\tilde{A}\|_\infty \le c_{m,n}\boldsymbol{u_f}\|\tilde{A}\|_\infty$$

$$\boldsymbol{u_s} = \boldsymbol{u_f}$$

Returning to IR3 Analysis...

The backward error for the correction solve:

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

$\boldsymbol{u_s} = \boldsymbol{u_f}$

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

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

Returning to IR3 Analysis...

The backward error for the correction solve:

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

$\boldsymbol{u_s} = \boldsymbol{u_f}$

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

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

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

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

Returning to IR3 Analysis...

The backward error for the correction solve:

$$\left(\tilde{A} + \Delta\tilde{A}\right)\hat{d}_i = \tilde{r}_i, \qquad \|\Delta\tilde{A}\|_\infty \le c_{m,n}\boldsymbol{u_f}\|\tilde{A}\|_\infty$$

$\boldsymbol{u_s} = \boldsymbol{u_f}$

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

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

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

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

3. $\left|\hat{r}_i - A\hat{d}_i\right| \le \boldsymbol{u_s}G_i|\hat{d}_i|$

$\boldsymbol{u_s}\|G_i\|_\infty = O\left(\boldsymbol{u_f}\|\tilde{A}\|_\infty\right)$

Returning to IR3 Analysis...

The backward error for the correction solve:

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

$\boldsymbol{u_s} = \boldsymbol{u_f}$

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

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

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

$$\frac{\|\tilde{x} - \hat{\tilde{x}}\|_\infty}{\|\tilde{x}\|_\infty} \approx \boldsymbol{u_r}\,\mathrm{cond}(\tilde{A}, \tilde{x}) + \boldsymbol{u}$$

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

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

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

$\boldsymbol{u_s}\|G_i\|_\infty = O(\boldsymbol{u_f}\|\tilde{A}\|_\infty)$

Returning to IR3 Analysis...

The backward error for the correction solve:

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

$\boldsymbol{u_s} = \boldsymbol{u_f}$

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

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

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

$$\frac{\|\tilde{x} - \hat{\tilde{x}}\|_\infty}{\|\tilde{x}\|_\infty} \approx \boldsymbol{u_r}\,\mathrm{cond}(\tilde{A}, \tilde{x}) + \boldsymbol{u}$$

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

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

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

$\boldsymbol{u_s}\|G_i\|_\infty = O(\boldsymbol{u_f}\|\tilde{A}\|_\infty)$

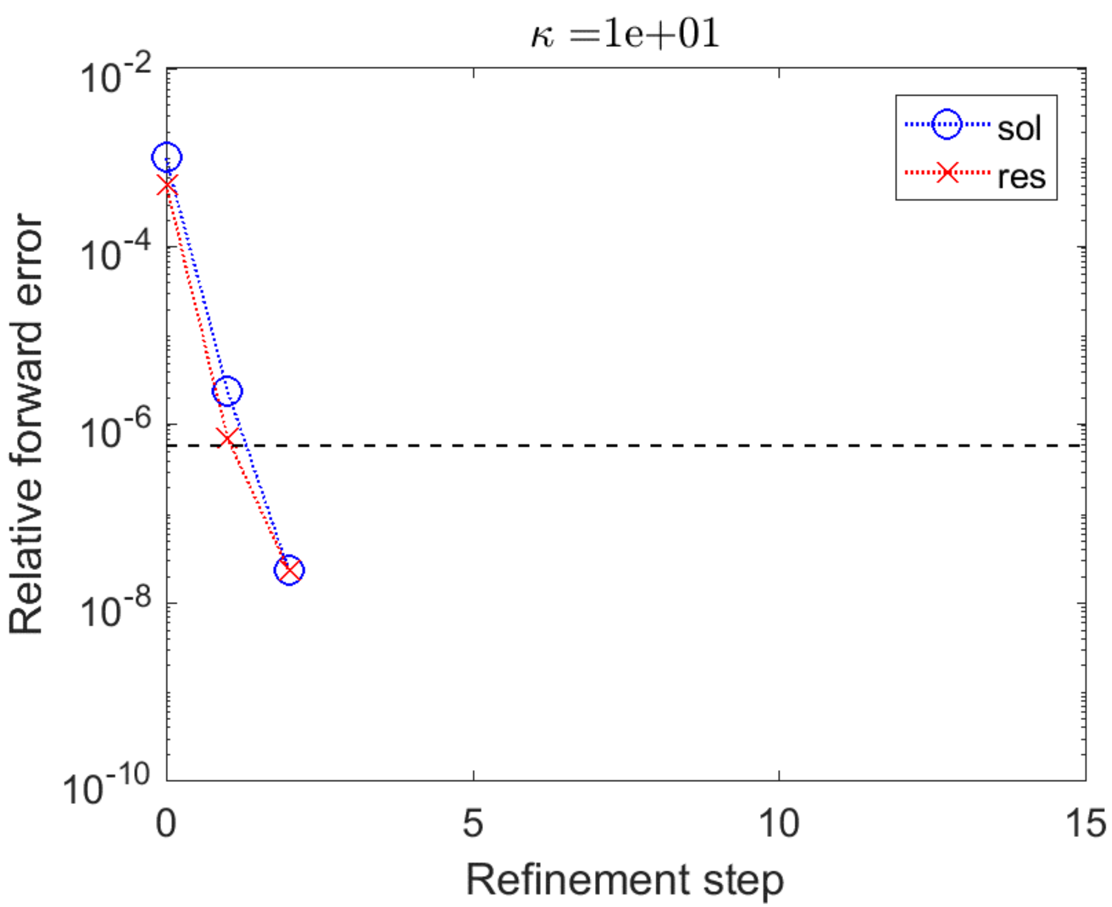
As long as $\kappa_\infty(\tilde{A}) \lesssim \boldsymbol{u_f^{-1}}$, expect normwise and componentwise backward errors to be $O(\boldsymbol{u})$

m n

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

Standard (QR-based) least squares IR with

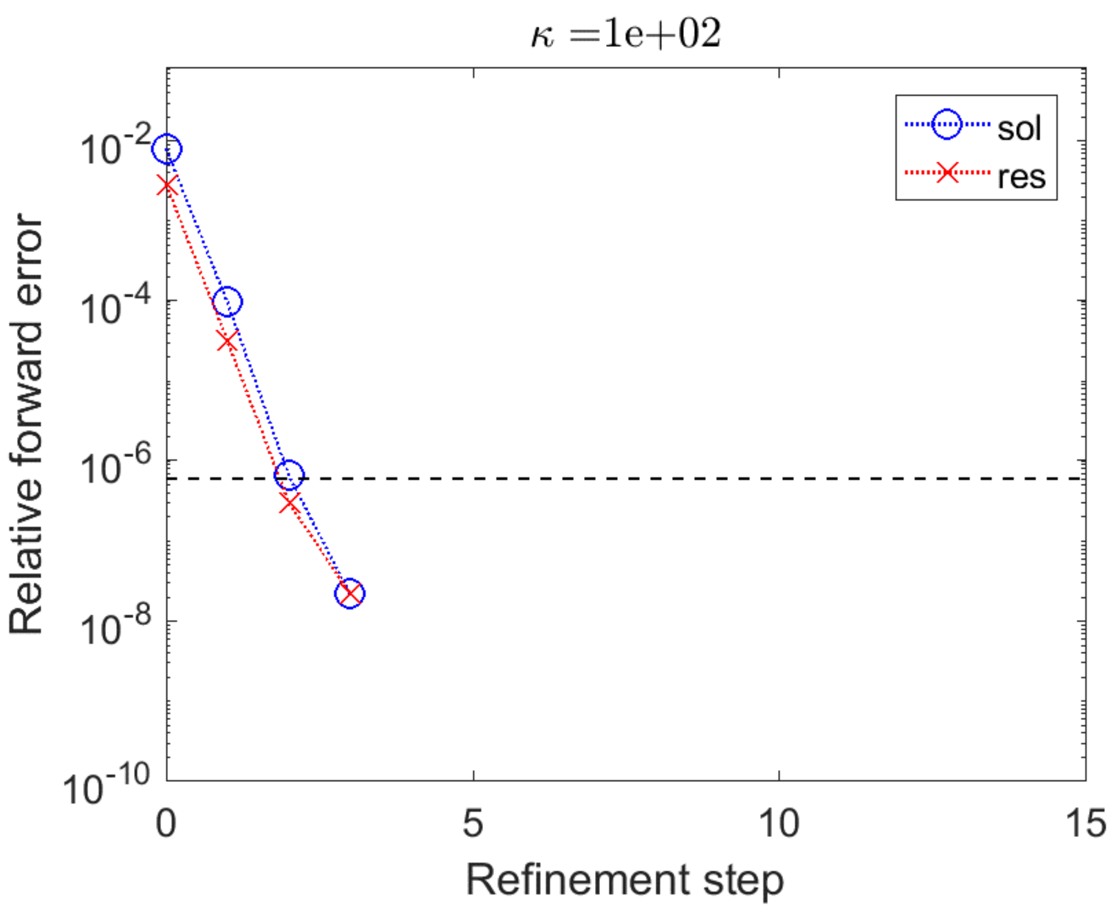
u_f: half, u: single, u_r: double



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

m n

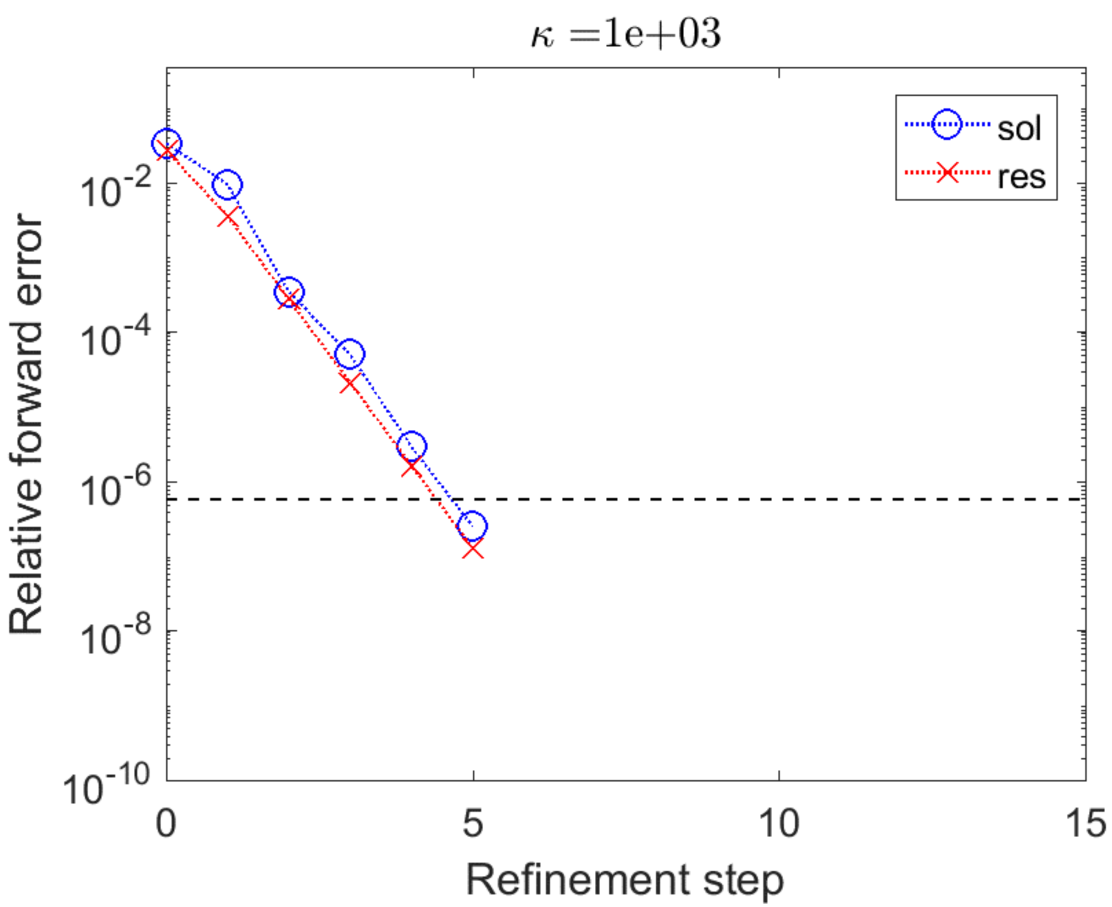
Standard (QR-based) least squares IR with
u_f: half, u: single, u_r: double



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

m n

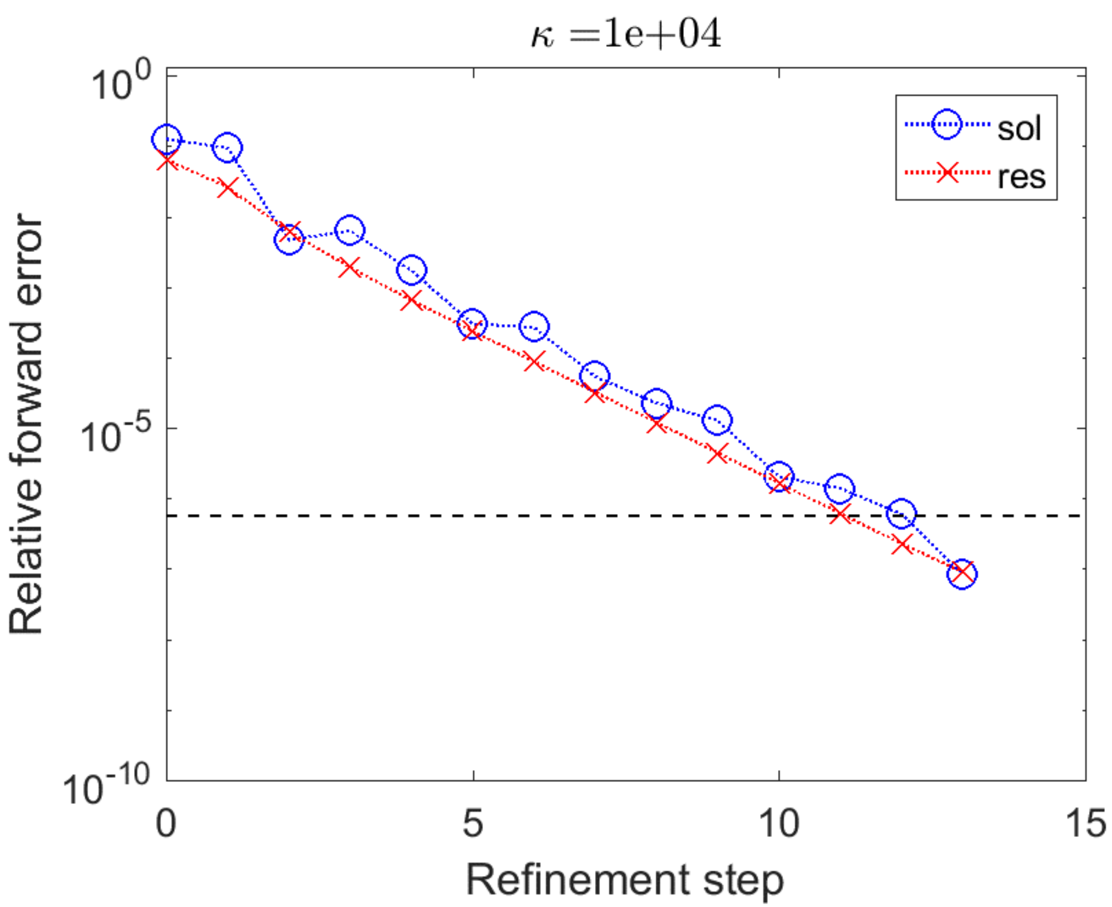
Standard (QR-based) least squares IR with
u_f: half, u: single, u_r: double



m n

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

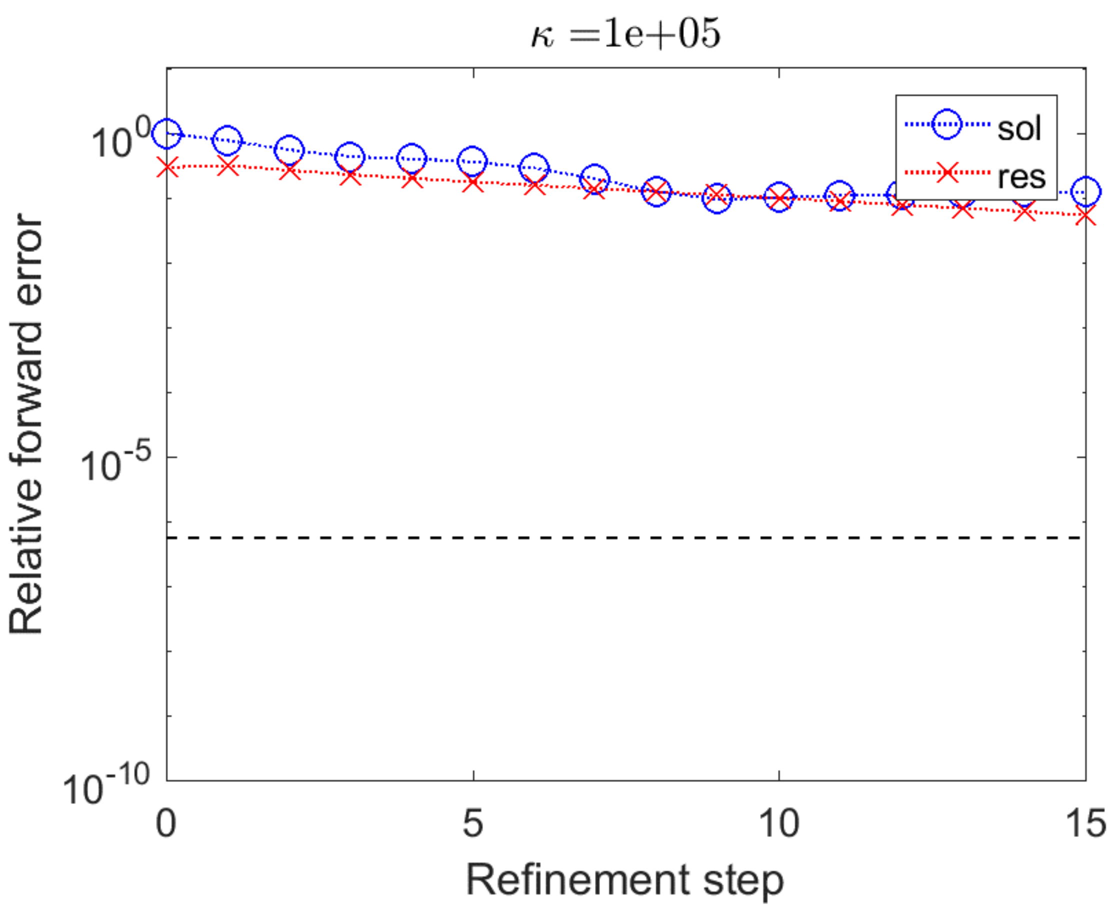
Standard (QR-based) least squares IR with
u_f: half, u: single, u_r: double



m n

```
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



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

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?

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:

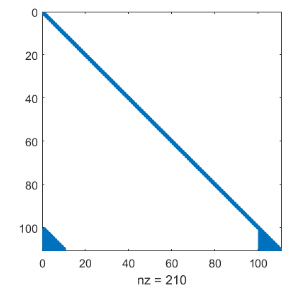
$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \approx \begin{bmatrix} I & 0 \\ R^T & U^T \end{bmatrix} \begin{bmatrix} I & R \\ 0 & -U \end{bmatrix}$$

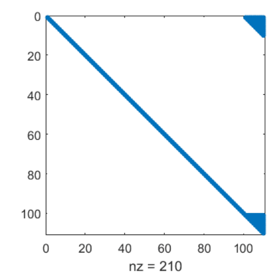
GMRES-IR for Least Squares

- 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:

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



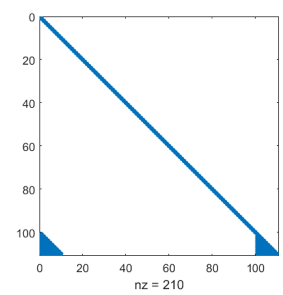


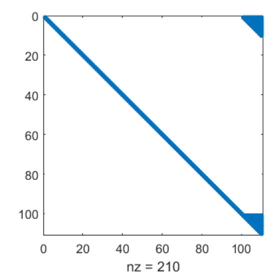
GMRES-IR for Least Squares

- 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

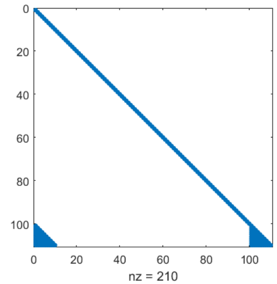
$$\begin{aligned} M &= (H + \alpha I)(S + \alpha I) \\ &= \begin{bmatrix} (\alpha + 1)I & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} \alpha I & R \\ R^T & \alpha I \end{bmatrix} \end{aligned}$$

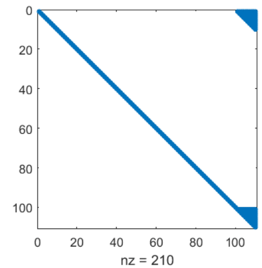
GMRES-IR for Least Squares

- 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:

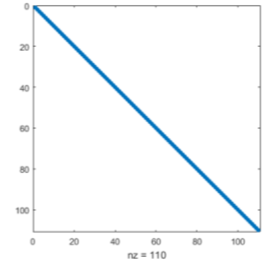
$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \approx \begin{bmatrix} I & 0 \\ R^T & U^T \end{bmatrix} \begin{bmatrix} I & R \\ 0 & -U \end{bmatrix}$$

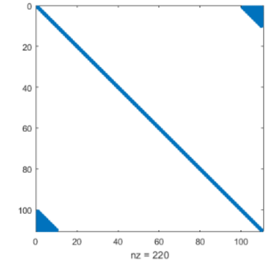




2. Use Hermitian/skew Hermitian splitting (HSS) preconditioning for saddlepoint systems; use left-preconditioned system matrix $M^{-1}\tilde{A}$ where

$$M = (H + \alpha I)(S + \alpha I)$$
$$= \begin{bmatrix} (\alpha+1)I & 0 \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} \alpha I & R \\ R^T & \alpha I \end{bmatrix}$$



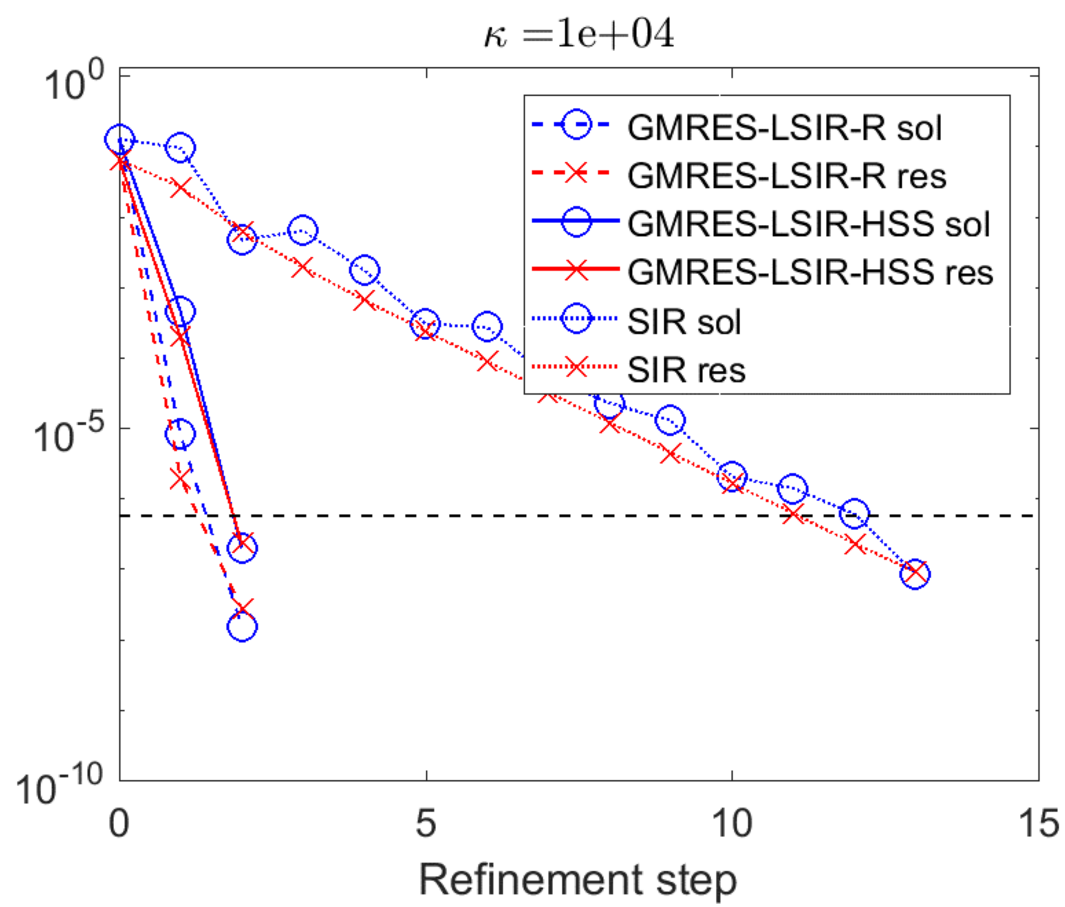


```
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)
```

m n

GMRES-LSIR and "Standard" LSIR with

u_f: half, u: single, u_r: double

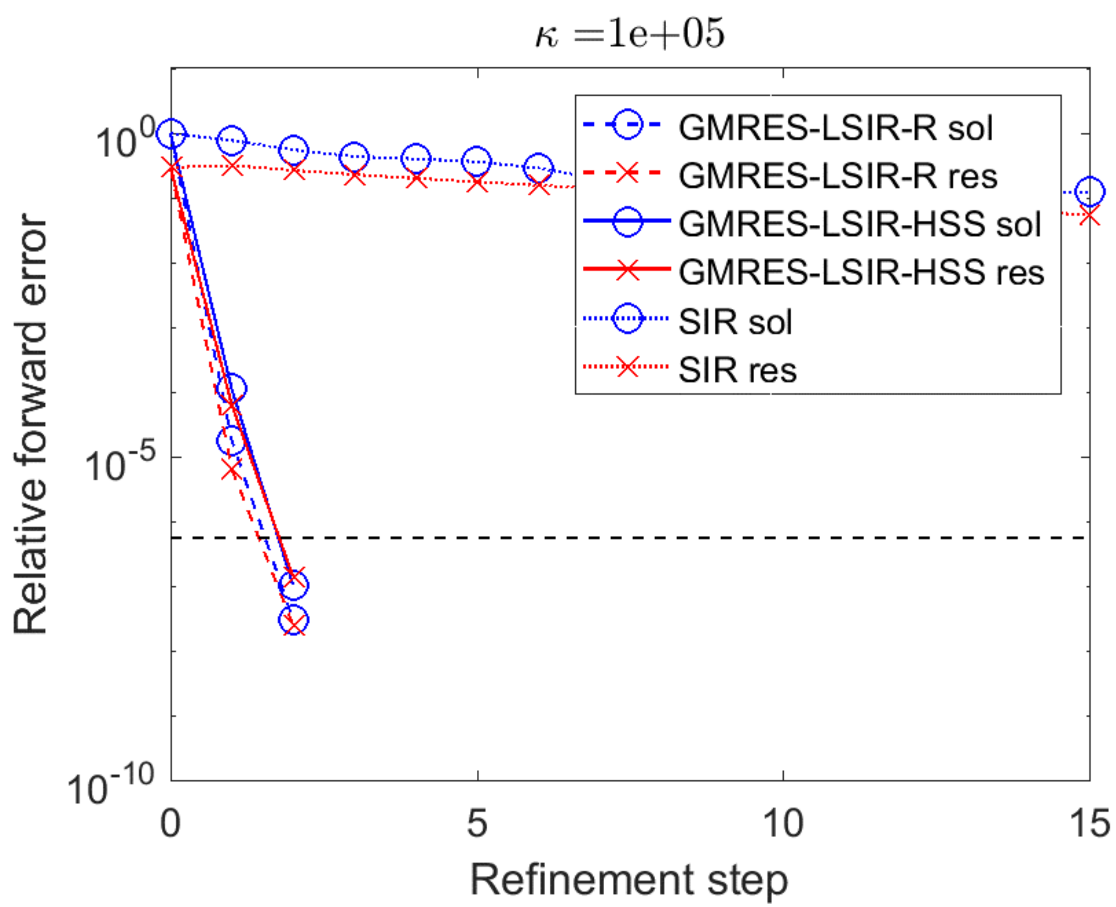


```
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)
```

m n

GMRES-LSIR and "Standard" LSIR with

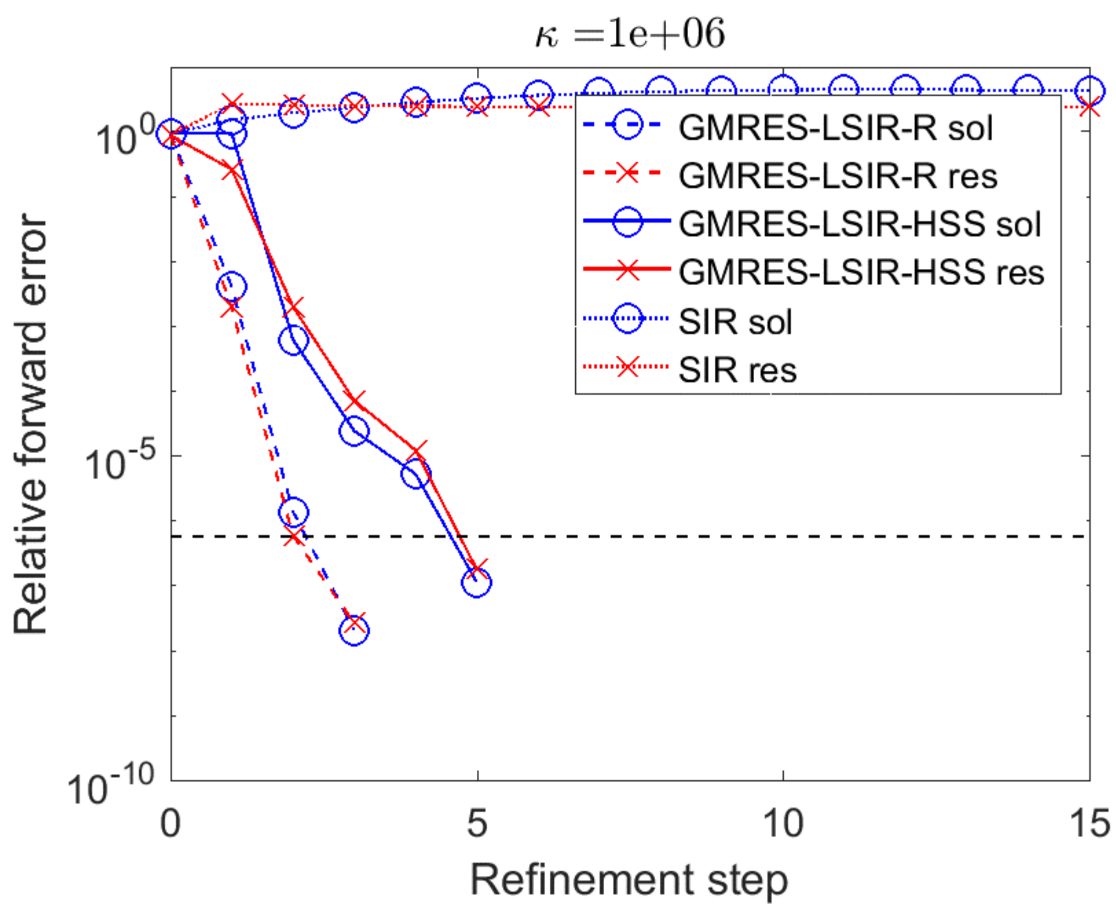
u_f: half, u: single, u_r: double



m n

```
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

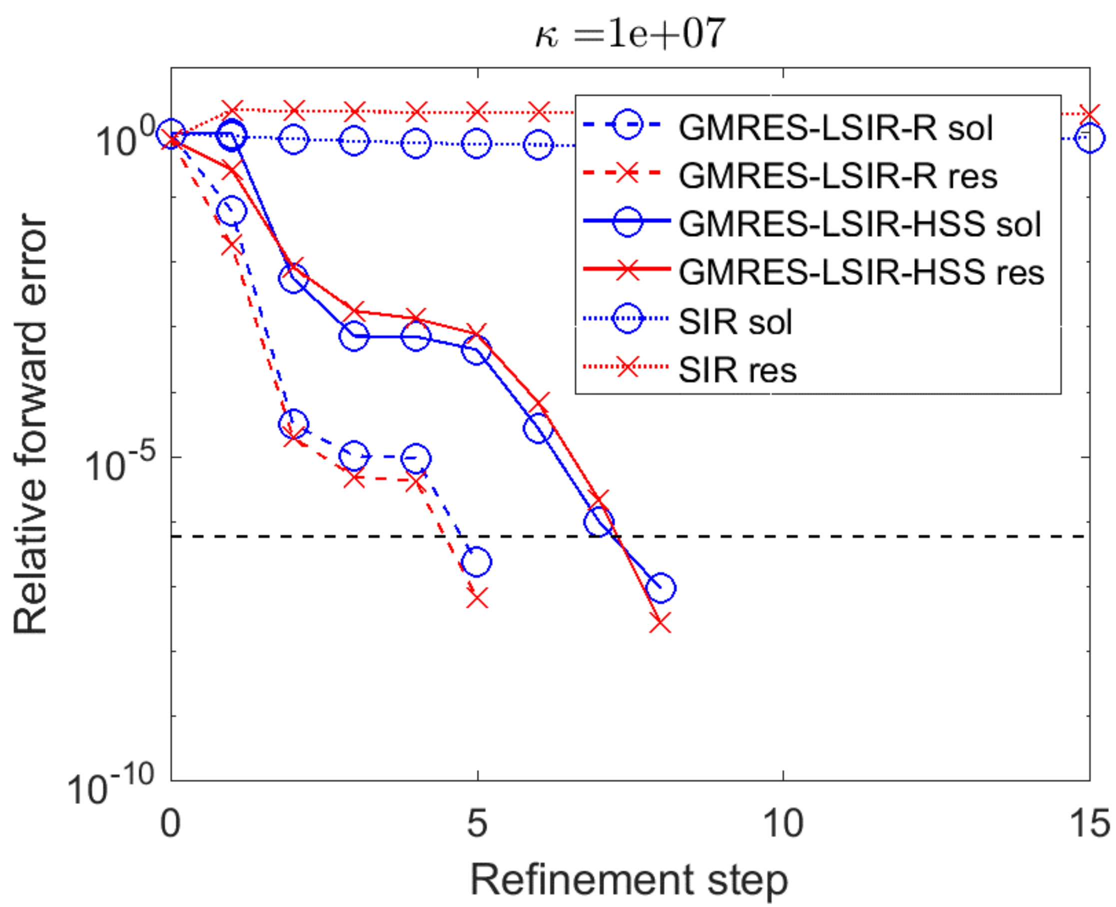


```
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)
```

m n

GMRES-LSIR and "Standard" LSIR with

u_f: half, u: single, u_r: double

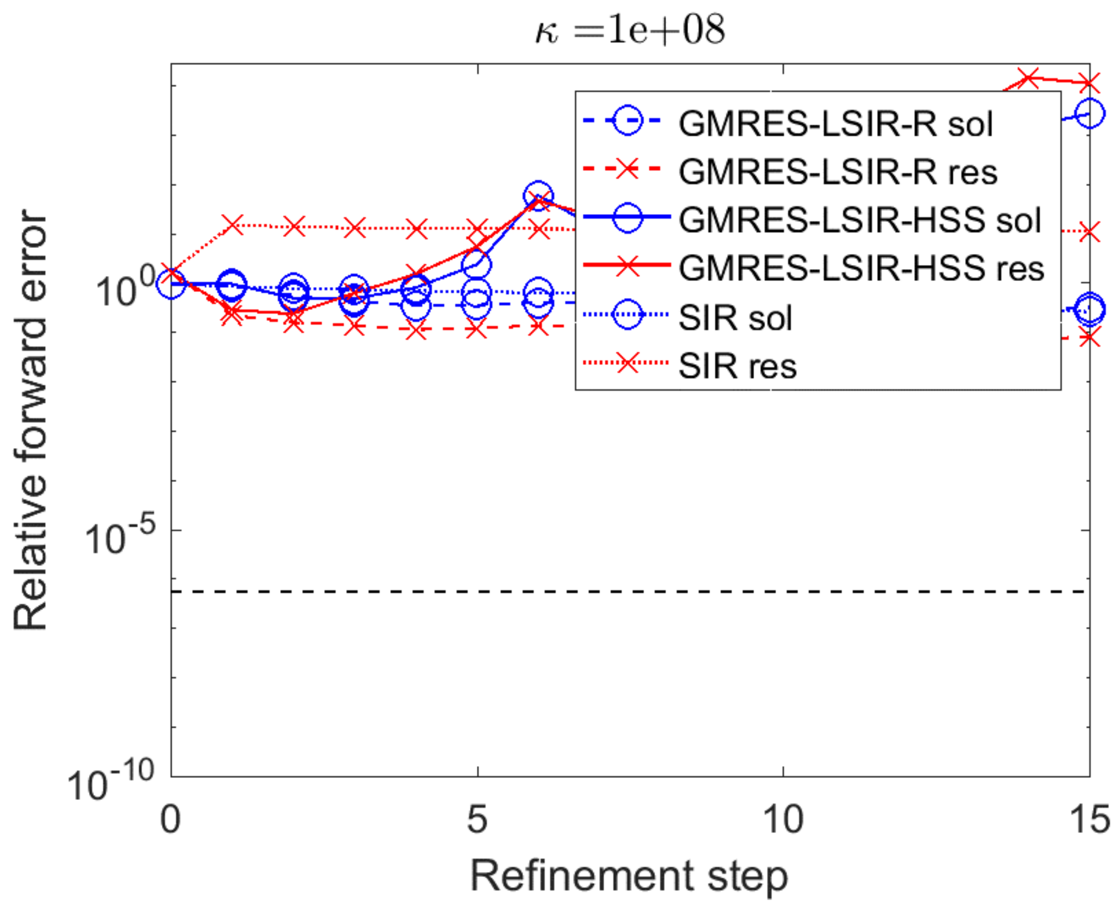


```
A = gallery('randsvd', 100, 10, kappa)
b = randn(100,1); b = b./norm(b)
```

m n

GMRES-LSIR and "Standard" LSIR with

u_f: half, u: single, u_r: double



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/