Mixed Precision Iterative Refinement

Erin Carson
Charles University

Householder Symposium XXI
Selva di Fasano, Italy, 2022

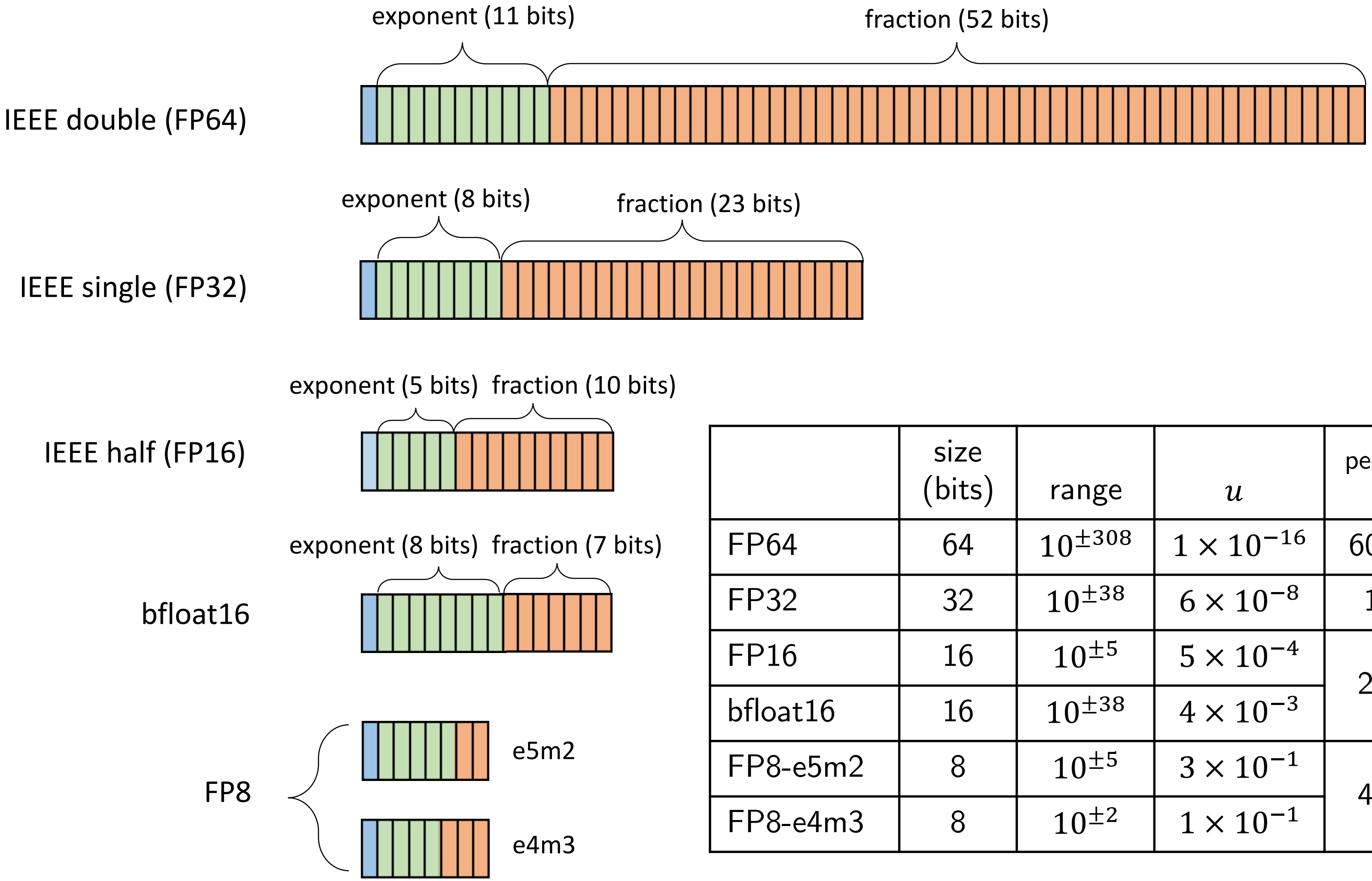
FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

Collaborators: Nicholas J. Higham (Manchester), Srikara Pranesh (V-Labs), Noaman Khan (Charles Univ.)

We acknowledge funding from Charles Univ. PRIMUS project No. PRIMUS/19/SCI/11, Charles Univ. Research Program No. UNCE/SCI/023, and the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Admin.

Floating Point Formats

$$(-1)^{\text{sign}} \times 2^{(\text{exponent}-\text{offset})} \times 1.\text{fraction}$$



IEEE double (FP64)

exponent (11 bits)

fraction (52 bits)

IEEE single (FP32)

exponent (8 bits)

fraction (23 bits)

IEEE half (FP16)

exponent (5 bits)

fraction (10 bits)

bfloat16

exponent (8 bits)

fraction (7 bits)

FP8

e5m2

e4m3

	size (bits)	range	u	perf. (NVIDIA H100)
FP64	64	$10^{\pm 308}$	1×10^{-16}	60 Tflops/s
FP32	32	$10^{\pm 38}$	6×10^{-8}	1 Pflop/s
FP16	16	$10^{\pm 5}$	5×10^{-4}	2 Pflops/s
bfloat16	16	$10^{\pm 38}$	4×10^{-3}	
FP8-e5m2	8	$10^{\pm 5}$	3×10^{-1}	4 Pflops/s
FP8-e4m3	8	$10^{\pm 2}$	1×10^{-1}	

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)
- NVIDIA A100, 2020: tensor cores with multiple supported precisions: FP16, FP64, Binary, INT4, INT8, bfloat16
- NVIDIA H100, 2022: now with quarter-precision (FP8) tensor cores
- Exascale supercomputers: Expected extensive support for reduced-precision arithmetic (Frontier: FP64, FP32, FP16, bfloat16, INT8, INT4)

Mixed precision in NLA

- BLAS: cuBLAS, MAGMA, [Agullo et al. 2009], [Abdelfattah et al., 2019], [Haidar et al., 2018]
- Iterative refinement:
    - Long history: [Wilkinson, 1963], [Moler, 1967], [Stewart, 1973], ...
    - More recently: [Langou et al., 2006], [C., Higham, 2017], [C., Higham, 2018], [C., Higham, Pranesh, 2020], [Amestoy et al., 2021]
- Matrix factorizations: [Haidar et al., 2017], [Haidar et al., 2018], [Haidar et al., 2020], [Abdelfattah et al., 2020]
- Eigenvalue problems: [Dongarra, 1982], [Dongarra, 1983], [Tisseur, 2001], [Davies et al., 2001], [Petschow et al., 2014], [Alvermann et al., 2019]
- Sparse direct solvers: [Buttari et al., 2008]
- Orthogonalization: [Yamazaki et al., 2015]
- Multigrid: [Tamstorf et al., 2020], [Richter et al., 2014], [Sumiyoshi et al., 2014], [Ljungkvist, Kronbichler, 2017, 2019]
- (Preconditioned) Krylov subspace methods: [Emans, van der Meer, 2012], [Yamagishi, Matsumura, 2016], [C., Gergelits, Yamazaki, 2021], [Clark, 2019], [Anzt et al., 2019], [Clark et al., 2010], [Gratton et al., 2020], [Arioli, Duff, 2009], [Hogg, Scott, 2010]

For survey and references, see [Abdelfattah et al., IJHPC, 2021]

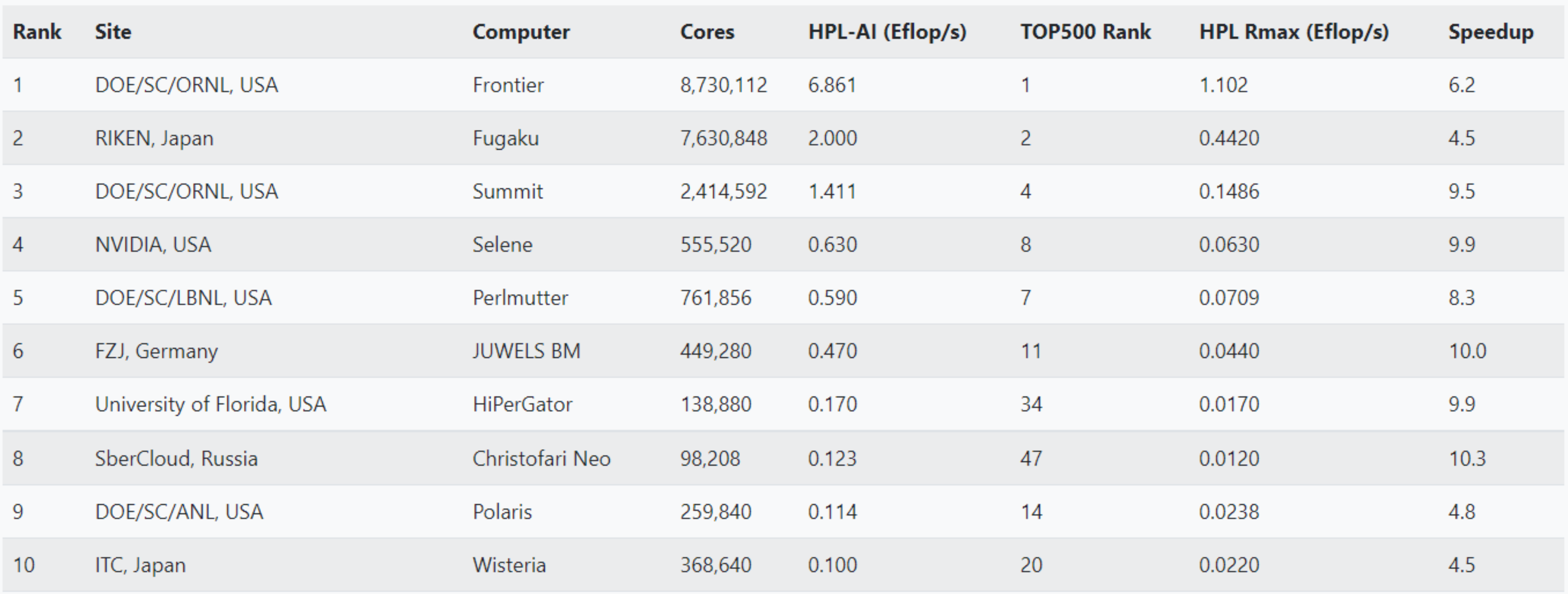
HPL-AI Benchmark

- Supercomputers traditionally ranked by performance on high-performance LINPACK (HPL) benchmark
  - Solves dense $Ax = b$ via Gaussian elimination with partial pivoting

- HPL-AI: Like HPL, solves dense $Ax = b$, results still to double precision accuracy
  - But achieves this via mixed-precision GMRES-based iterative refinement

HPL-AI Benchmark

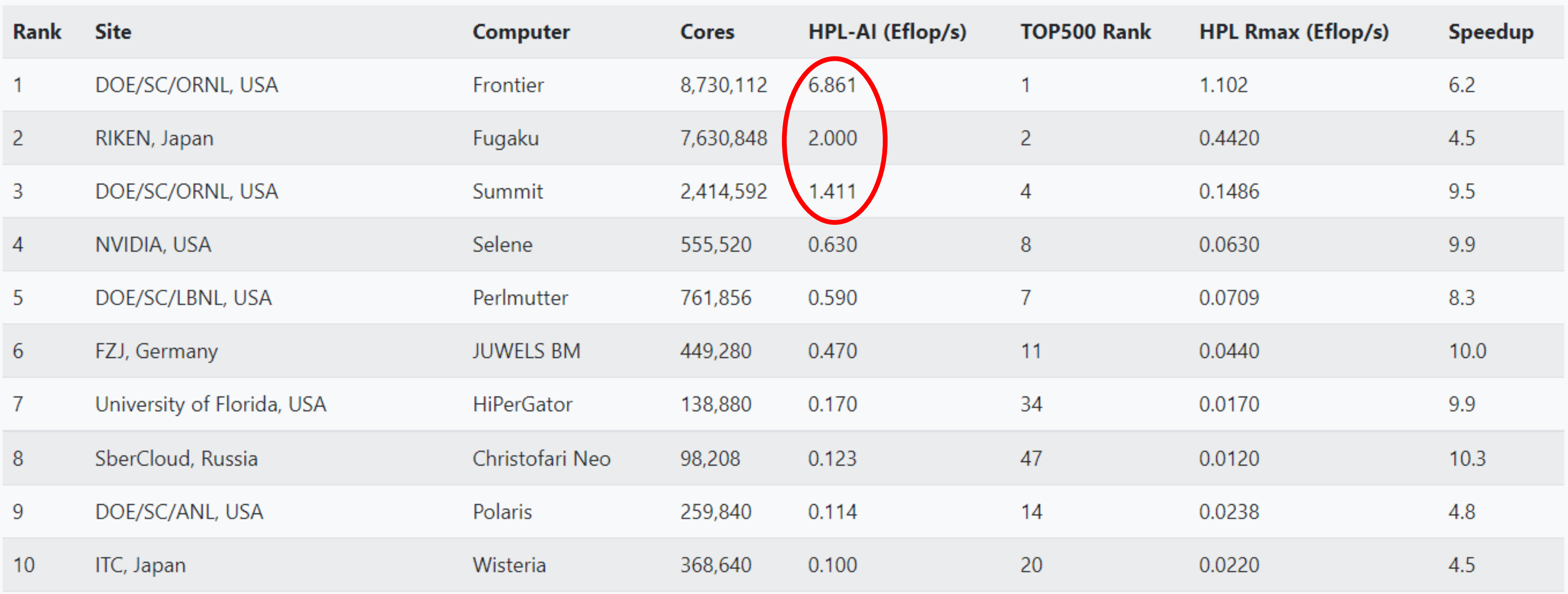
June 2022



Rank	Site	Computer	Cores	HPL-AI (Eflop/s)	TOP500 Rank	HPL Rmax (Eflop/s)	Speedup
1	DOE/SC/ORNL, USA	Frontier	8,730,112	6.861	1	1.102	6.2
2	RIKEN, Japan	Fugaku	7,630,848	2.000	2	0.4420	4.5
3	DOE/SC/ORNL, USA	Summit	2,414,592	1.411	4	0.1486	9.5
4	NVIDIA, USA	Selene	555,520	0.630	8	0.0630	9.9
5	DOE/SC/LBNL, USA	Perlmutter	761,856	0.590	7	0.0709	8.3
6	FZJ, Germany	JUWELS BM	449,280	0.470	11	0.0440	10.0
7	University of Florida, USA	HiPerGator	138,880	0.170	34	0.0170	9.9
8	SberCloud, Russia	Christofari Neo	98,208	0.123	47	0.0120	10.3
9	DOE/SC/ANL, USA	Polaris	259,840	0.114	14	0.0238	4.8
10	ITC, Japan	Wisteria	368,640	0.100	20	0.0220	4.5

HPL-AI Benchmark

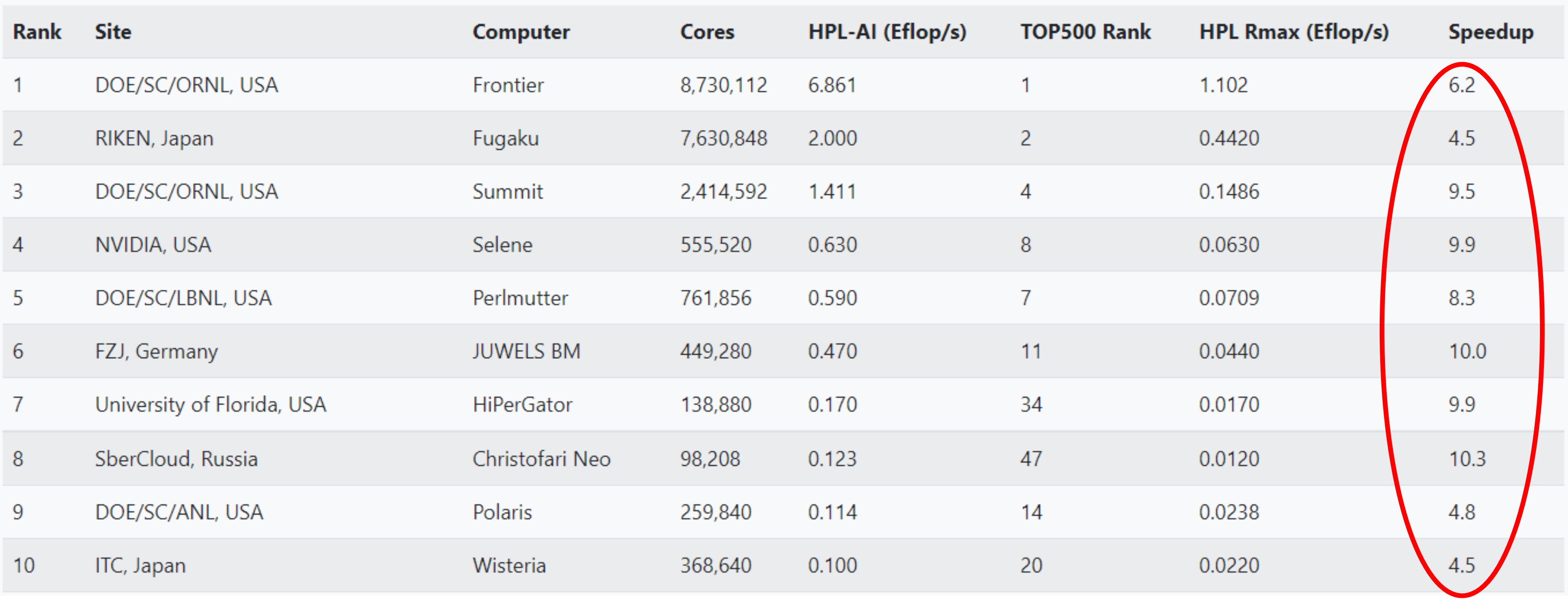
June 2022



Rank	Site	Computer	Cores	HPL-AI (Eflop/s)	TOP500 Rank	HPL Rmax (Eflop/s)	Speedup
1	DOE/SC/ORNL, USA	Frontier	8,730,112	6.861	1	1.102	6.2
2	RIKEN, Japan	Fugaku	7,630,848	2.000	2	0.4420	4.5
3	DOE/SC/ORNL, USA	Summit	2,414,592	1.411	4	0.1486	9.5
4	NVIDIA, USA	Selene	555,520	0.630	8	0.0630	9.9
5	DOE/SC/LBNL, USA	Perlmutter	761,856	0.590	7	0.0709	8.3
6	FZJ, Germany	JUWELS BM	449,280	0.470	11	0.0440	10.0
7	University of Florida, USA	HiPerGator	138,880	0.170	34	0.0170	9.9
8	SberCloud, Russia	Christofari Neo	98,208	0.123	47	0.0120	10.3
9	DOE/SC/ANL, USA	Polaris	259,840	0.114	14	0.0238	4.8
10	ITC, Japan	Wisteria	368,640	0.100	20	0.0220	4.5

HPL-AI Benchmark

June 2022



Rank	Site	Computer	Cores	HPL-AI (Eflop/s)	TOP500 Rank	HPL Rmax (Eflop/s)	Speedup
1	DOE/SC/ORNL, USA	Frontier	8,730,112	6.861	1	1.102	6.2
2	RIKEN, Japan	Fugaku	7,630,848	2.000	2	0.4420	4.5
3	DOE/SC/ORNL, USA	Summit	2,414,592	1.411	4	0.1486	9.5
4	NVIDIA, USA	Selene	555,520	0.630	8	0.0630	9.9
5	DOE/SC/LBNL, USA	Perlmutter	761,856	0.590	7	0.0709	8.3
6	FZJ, Germany	JUWELS BM	449,280	0.470	11	0.0440	10.0
7	University of Florida, USA	HiPerGator	138,880	0.170	34	0.0170	9.9
8	SberCloud, Russia	Christofari Neo	98,208	0.123	47	0.0120	10.3
9	DOE/SC/ANL, USA	Polaris	259,840	0.114	14	0.0238	4.8
10	ITC, Japan	Wisteria	368,640	0.100	20	0.0220	4.5

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 $Ax = b$

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

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

- 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 $Ax = b$

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

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

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

```
Solve $Ax_0 = b$ by LU factorization                          (in precision $u$)
for $i = 0\colon \mathrm{maxit}$
        $r_i = b - Ax_i$                                      (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 $\kappa_\infty(A) \le \boldsymbol{u}^{-\mathbf{1/2}}$,

- 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 $i = 0\colon \mathrm{maxit}$
        $r_i = b - Ax_i$                                 (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 $Ax = b$

3-precision iterative refinement [C. and Higham, 2018]

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

$$u_f \geq u \geq u_r$$

```
Solve $Ax_0 = b$ by LU factorization                (in precision $\boldsymbol{u_f}$)
for $i = 0\colon \mathrm{maxit}$
        $r_i = b - Ax_i$                            (in precision $\boldsymbol{u_r}$)
        Solve $Ad_i = r_i$                          (in precision $\boldsymbol{u_s}$)
        $x_{i+1} = x_i + d_i$                       (in precision $\boldsymbol{u}$)
```

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

Key Aspects of Analysis: Tighter Upper Bounds

Obtain tighter upper bounds:

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

Key Aspects of Analysis: Tighter Upper Bounds

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 Aspects of Analysis: Tighter Upper Bounds

Obtain tighter upper bounds:

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

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 Aspects of Analysis: Tighter Upper Bounds

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 Aspects of Analysis: Effective Solve Precision

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 Aspects of Analysis: Effective Solve Precision

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

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 Aspects of Analysis: Effective Solve Precision

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:

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

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 Aspects of Analysis: Effective Solve Precision

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:

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

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

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 Aspects of Analysis: Effective Solve Precision

Allow for general solver:

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

example: LU solve:

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

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

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

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

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

Key Aspects of Analysis: Effective Solve Precision

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:

example: LU solve:

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

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

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

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

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

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

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

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

Key Aspects of Analysis: Effective Solve Precision

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

example: LU solve:

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

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

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

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

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

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

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

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

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

Key Aspects of Analysis: Effective Solve Precision

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 $\max(c_1, c_2)\, u_s$

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

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

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

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

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

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

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 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.

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 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 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 $\approx 10^{-34}$

					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}	$\mathrm{cond}(A,x) \cdot 10^{-8}$
	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 $\approx 10^{-34}$

					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}
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 $\approx 10^{-34}$

					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}	$\mathrm{cond}(A,x) \cdot 10^{-8}$
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		Forward error
					norm	comp	
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 $\approx 10^{-34}$

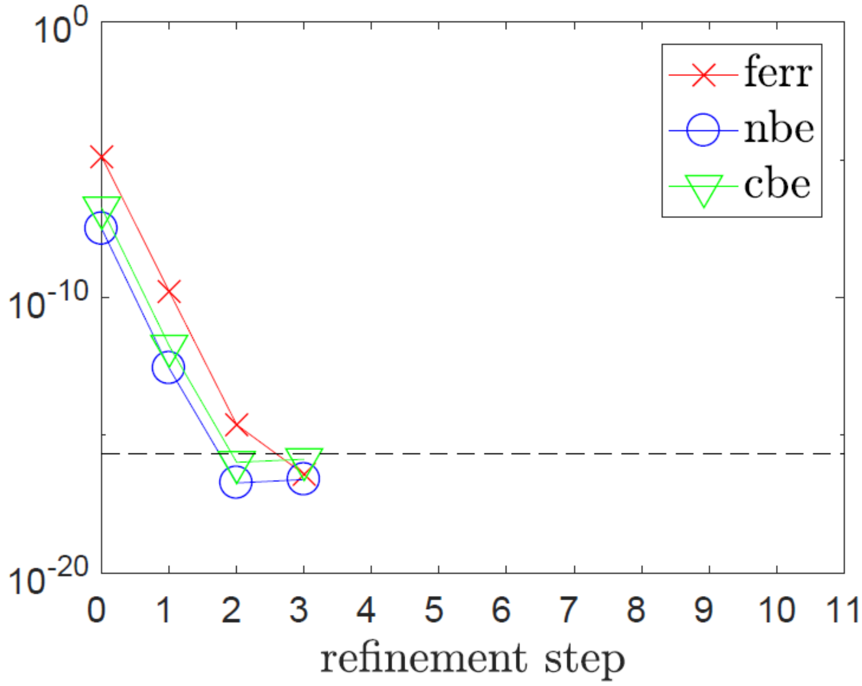
					Backward error		
	u_f	u	u_r	max $\kappa_\infty(A)$	norm	comp	Forward error
LP fact.	H	S	S	10^4	10^{-8}	10^{-8}	$\text{cond}(A,x) \cdot 10^{-8}$
New	H	S	D	10^4	10^{-8}	10^{-8}	10^{-8}
LP fact.	H	D	D	10^4	10^{-16}	10^{-16}	$\text{cond}(A,x) \cdot 10^{-16}$
New	H	D	Q	10^4	10^{-16}	10^{-16}	10^{-16}
Fixed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

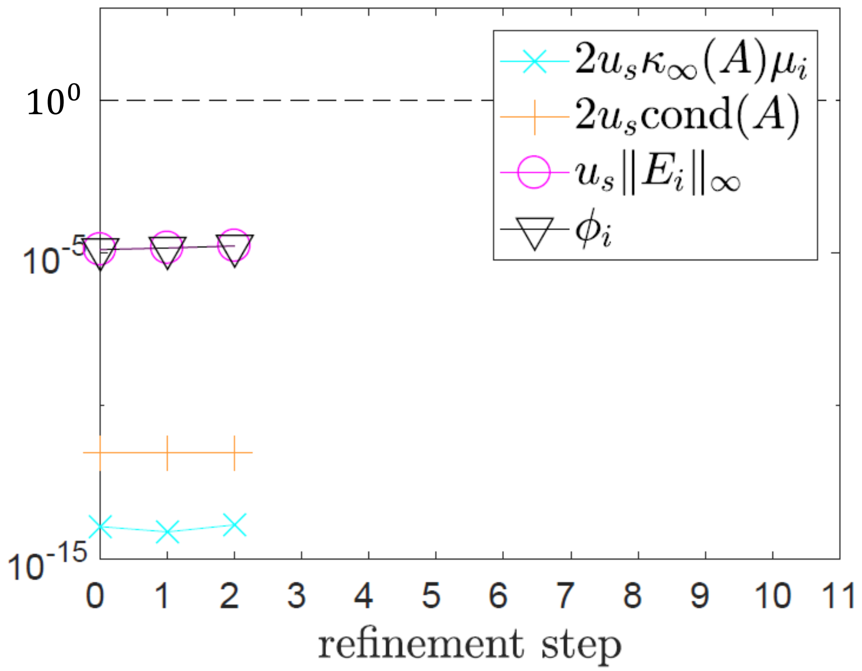
$\Rightarrow$ Benefit of IR3 vs. traditional IR: As long as $\kappa_\infty(A) \leq 10^4$, can use lower precision factorization w/no loss of accuracy!

```
A = gallery('randsvd', 100, 1e3)
b = randn(100,1)
```

$\boldsymbol{\kappa_\infty(A)} \approx$ **1e4**

Standard (LU-based) IR with $\boldsymbol{u_f}$: single, $\boldsymbol{u}$: double, $\boldsymbol{u_r}$: quad

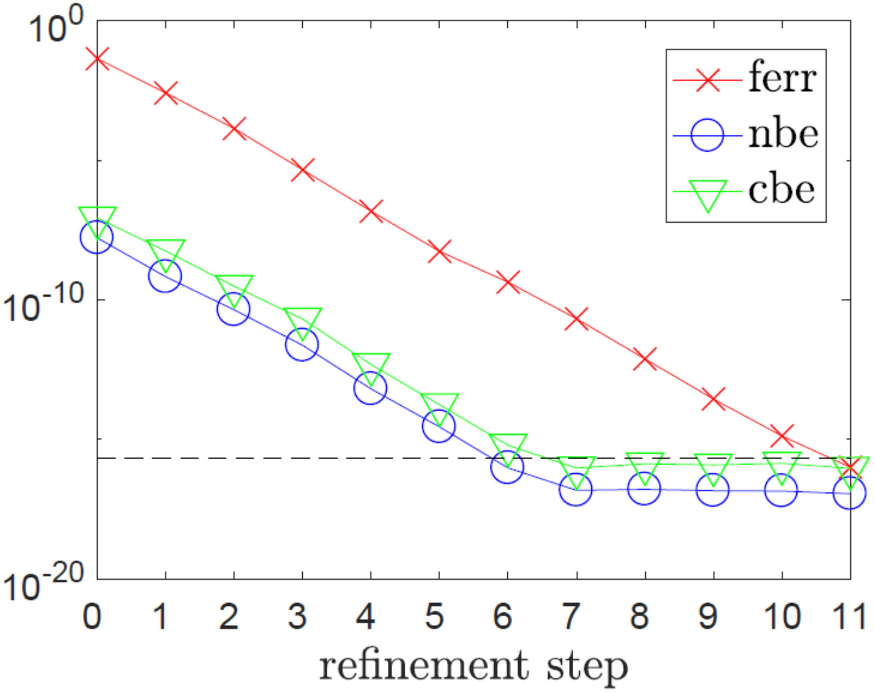


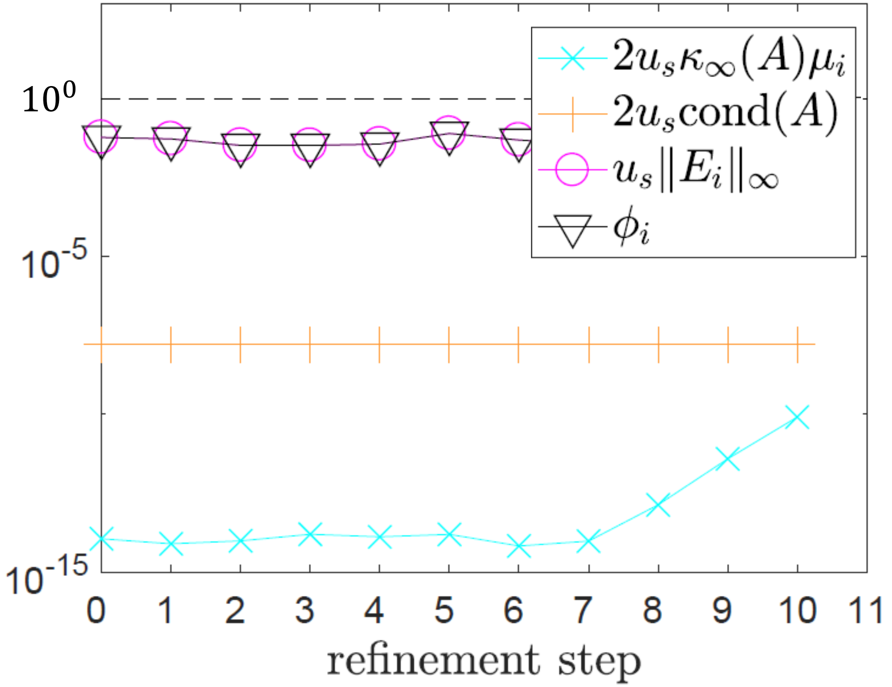


```
A = gallery('randsvd', 100, 1e7)
b = randn(100,1)
```

$\kappa_\infty(A) \approx$ 7e7

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

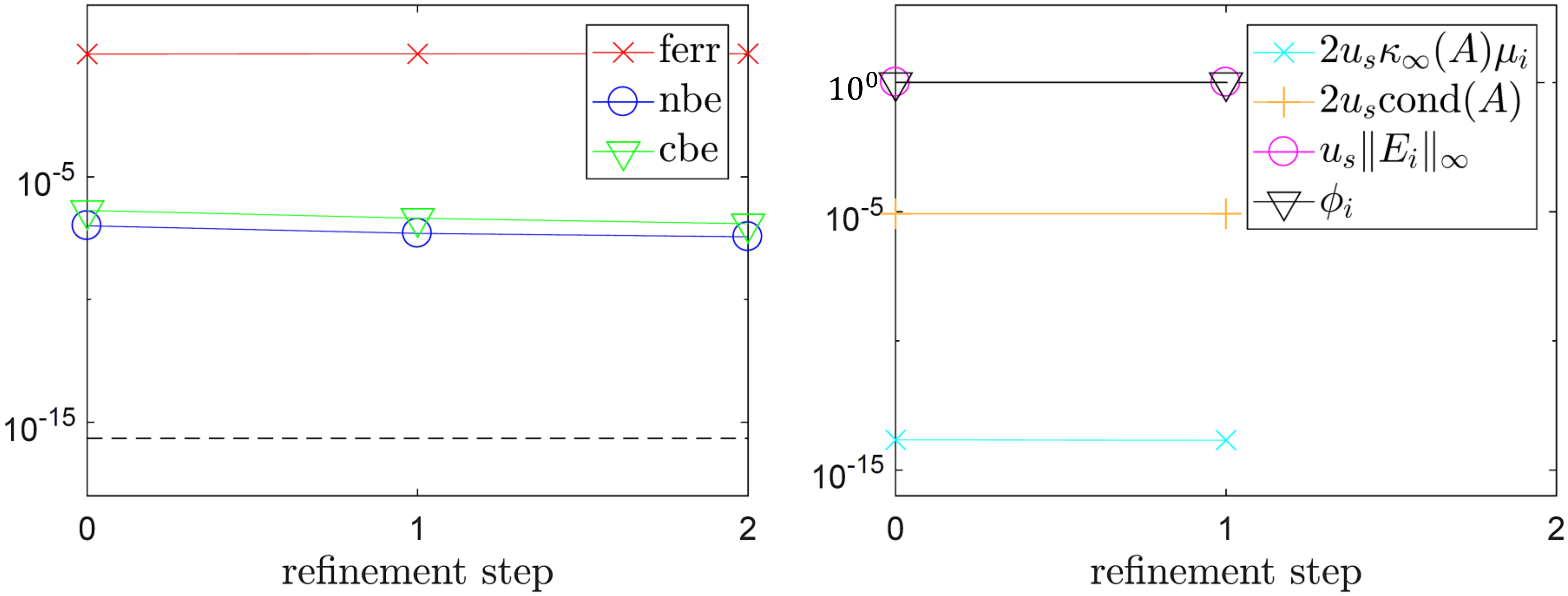




```
A = gallery('randsvd', 100, 1e9)
b = randn(100,1)
```

$\kappa_\infty(A) \approx$ **2e10**

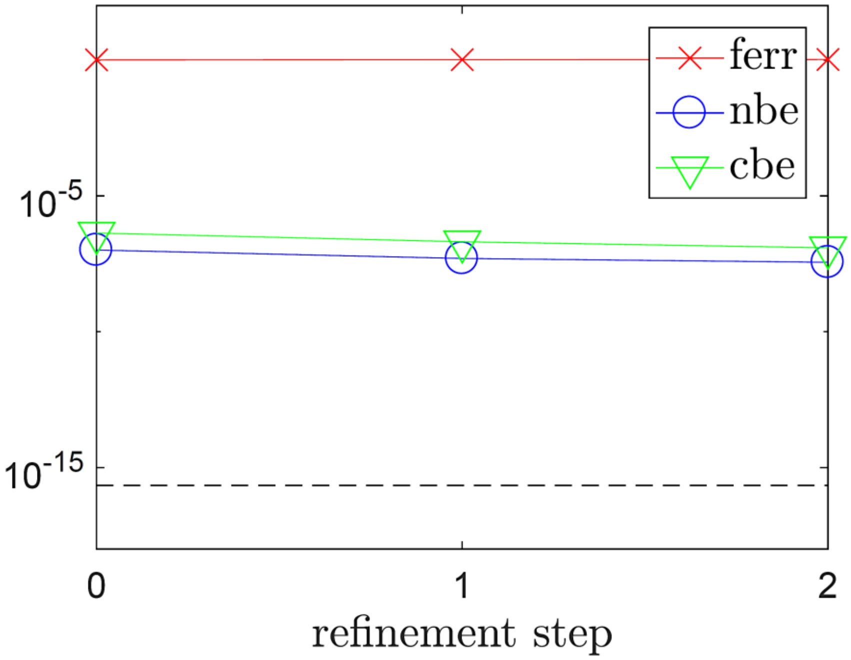
Standard (LU-based) IR with u_f: single, u: double, u_r: quad

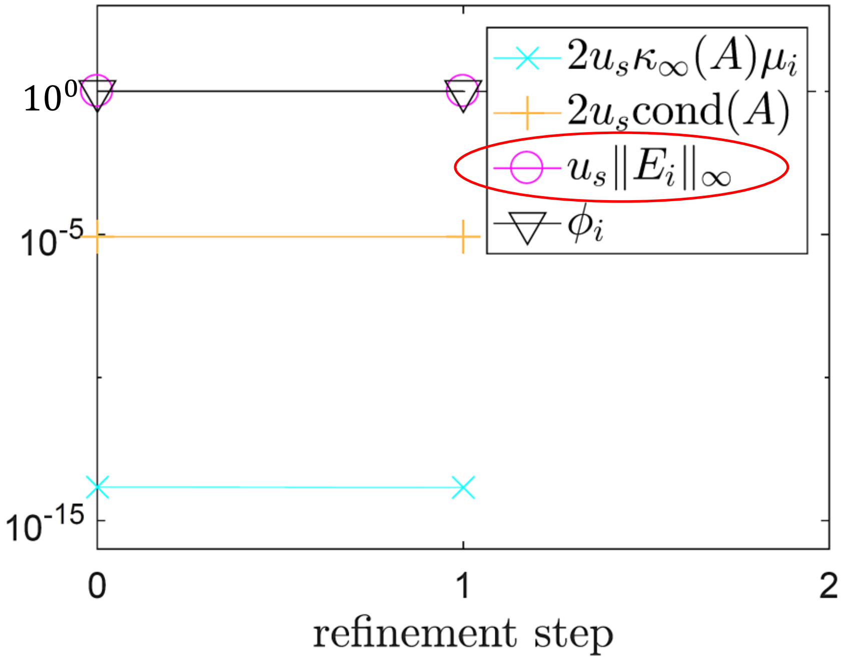


```
A = gallery('randsvd', 100, 1e9)
b = randn(100,1)
```

$\kappa_\infty(A) \approx$ 2e10

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

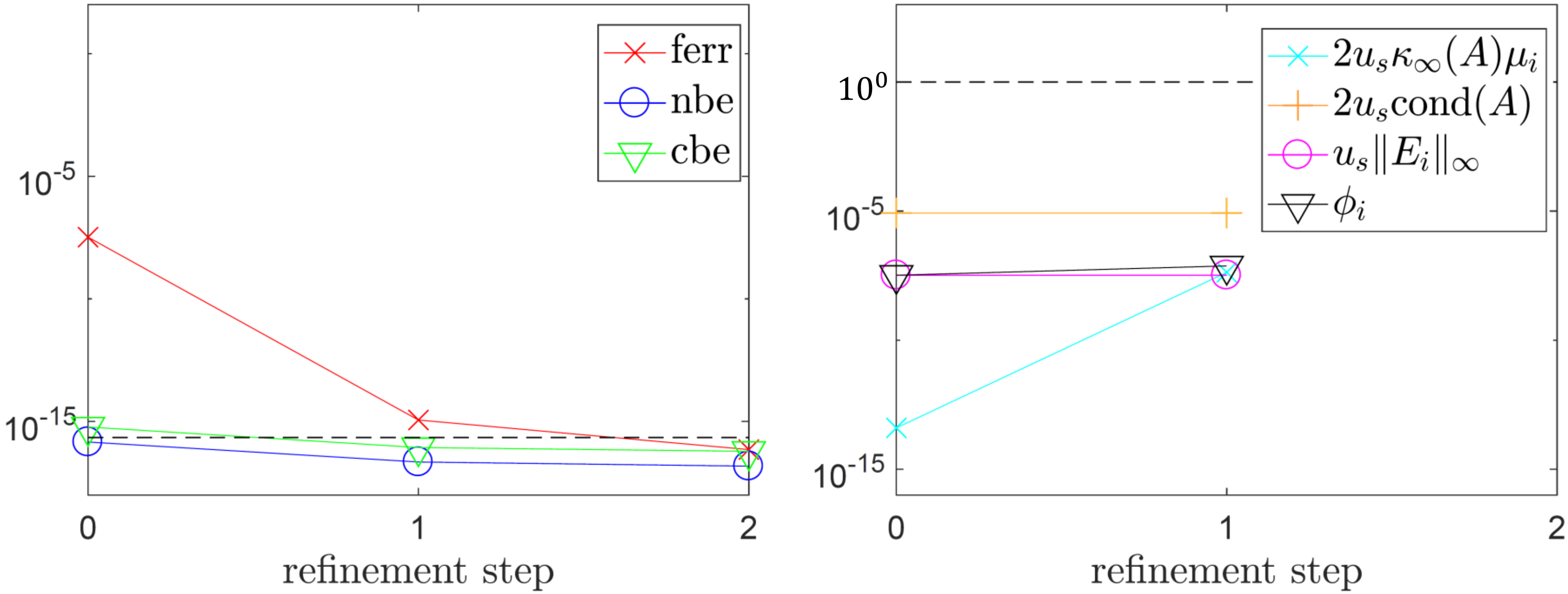




```
A = gallery('randsvd', 100, 1e9)
b = randn(100,1)
```

$\kappa_\infty(A) \approx$ **2e10**

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



GMRES-Based Iterative Refinement

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

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

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 $\boldsymbol{u_f}$, then

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

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

GMRES-Based Iterative Refinement

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

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

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 $\boldsymbol{u_f}$, then

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

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

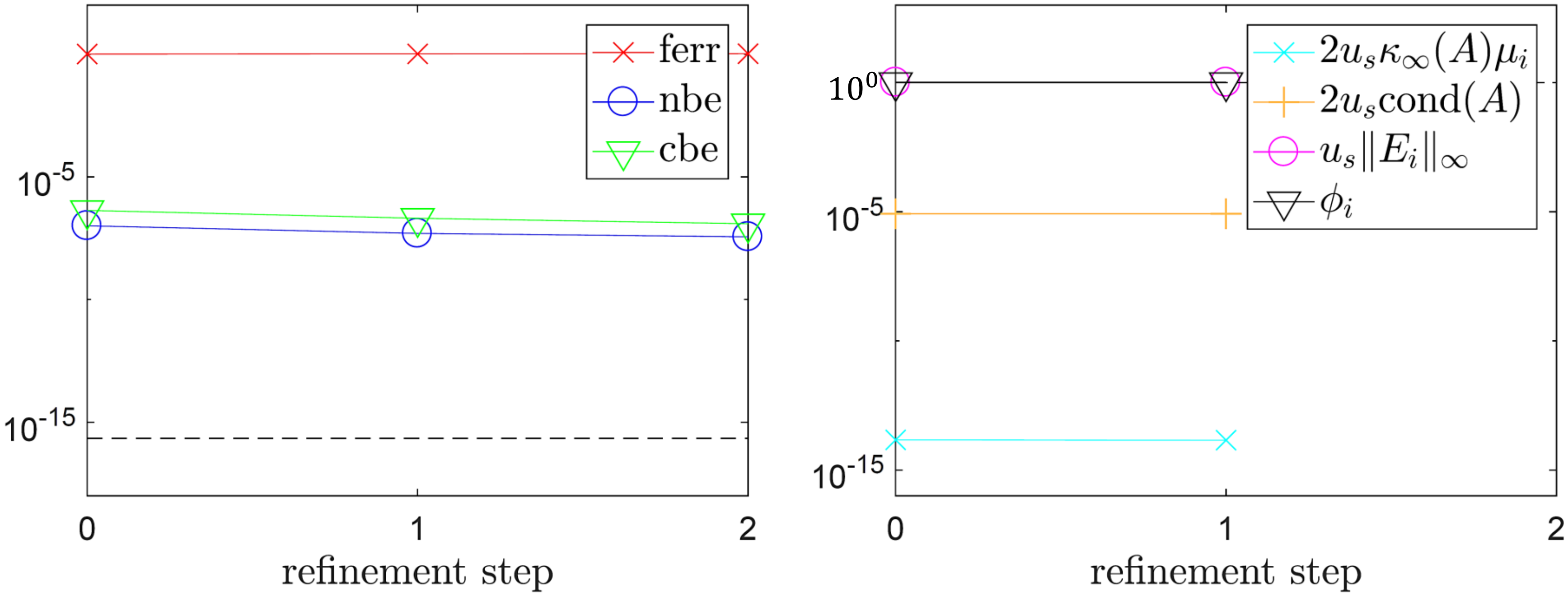
```
Solve $Ax_0 = b$ by LU factorization
for $i = 0$: maxit
        $r_i = b - Ax_i$
        Solve $Ad_i = r_i$    via GMRES on $\tilde{A}d_i = \tilde{r}_i$
        $x_{i+1} = x_i + d_i$
```

$\boldsymbol{u_s} = \boldsymbol{u}$

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

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

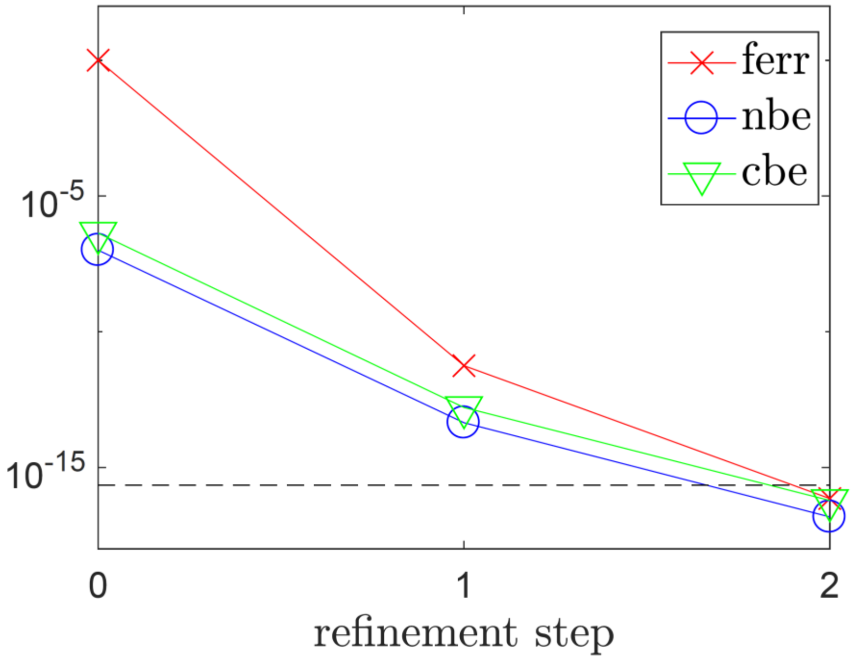
Standard (LU-based) IR with u_f: single, u: double, u_r: quad

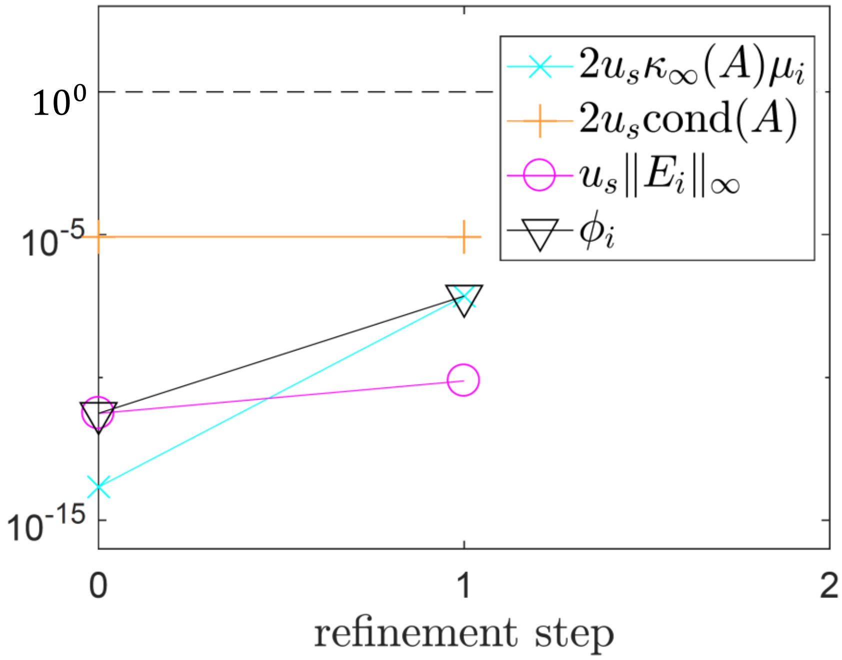


```
A = gallery('randsvd', 100, 1e9, 2)
b = randn(100,1)
```

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

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





Number of GMRES iterations: (2,3)

GMRES-IR: Summary

GMRES-IR: Solve for d_i via GMRES on $U^{-1}L^{-1}Ad_i = U^{-1}L^{-1}r_i$

GMRES-based IR in three precisions ($\boldsymbol{u_s} = \boldsymbol{u}$)

					Backward error		
	$\boldsymbol{u_f}$	$\boldsymbol{u}$	$\boldsymbol{u_r}$	max $\kappa_\infty(A)$	norm	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 $\kappa_\infty(A)$

GMRES-IR: Summary

GMRES-IR: Solve for d_i via GMRES on $U^{-1}L^{-1}Ad_i = U^{-1}L^{-1}r_i$

GMRES-based IR in three precisions ($\boldsymbol{u_s} = \boldsymbol{u}$)

					Backward error		
	$\boldsymbol{u_f}$	$\boldsymbol{u}$	$\boldsymbol{u_r}$	max $\kappa_\infty(A)$	norm	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}

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

⇒ As long as $\kappa_\infty(A) \leq 10^{12}$, can use half precision factorization and still obtain double precision accuracy!

Comments and Caveats I

- Convergence tolerance τ for GMRES?
    - Smaller τ → more GMRES iterations, potentially fewer refinement steps
    - Larger τ → fewer GMRES iterations, potentially more refinement steps

Comments and Caveats I

- Convergence tolerance τ for GMRES?
    - Smaller τ → more GMRES iterations, potentially fewer refinement steps
    - Larger τ → fewer GMRES iterations, potentially more refinement steps
- What about overflow, underflow, subnormal numbers?
    - Sophisticated scaling methods can help avoid this
        - "Squeezing a Matrix into Half Precision, with an Application to Solving Linear Systems" [Higham, Pranesh, Zounon, 2019]

Comments and Caveats II

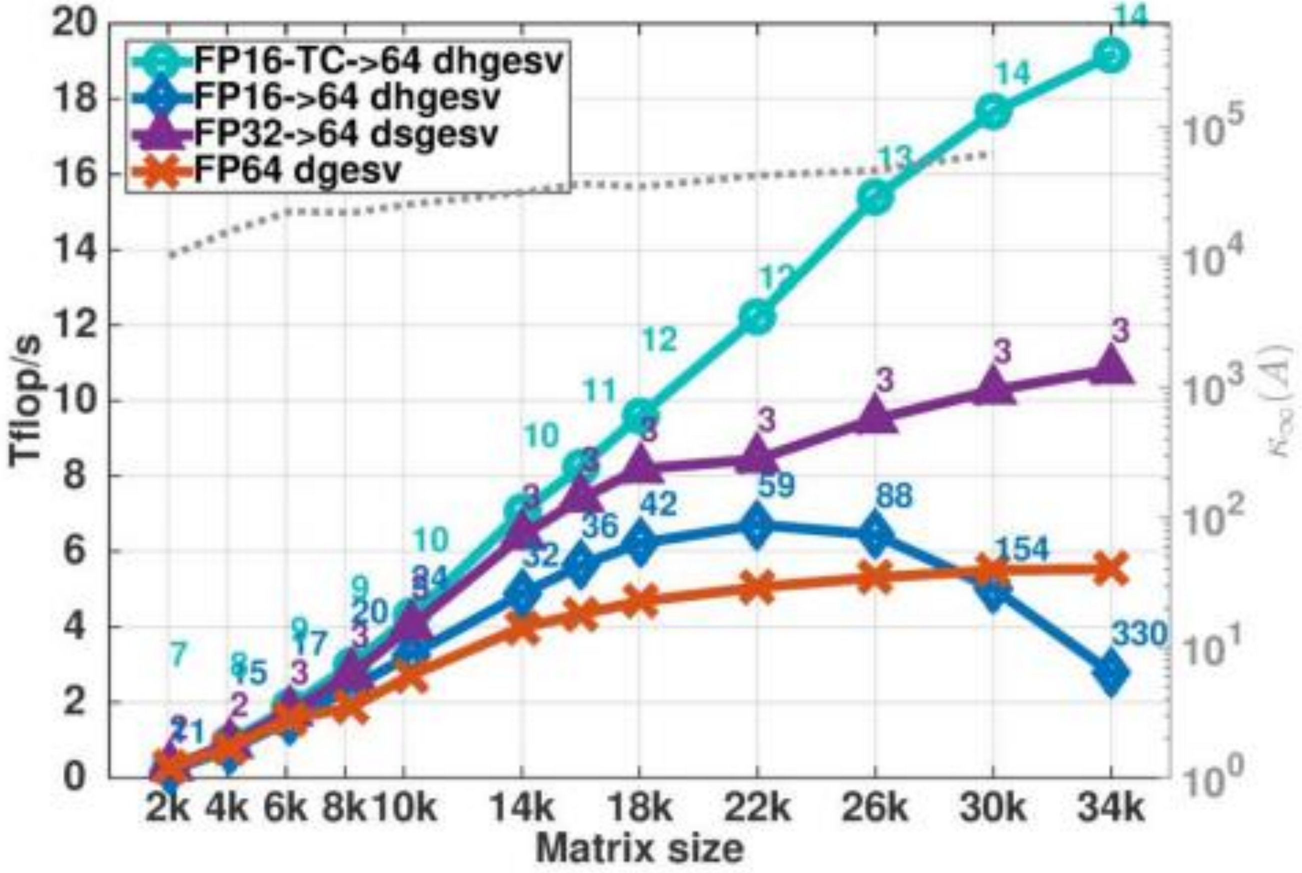
- Convergence rate of GMRES?

Comments and Caveats II

- 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 (normal) $\tilde{A}$ 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 [C., Oktay, 2022], using additional preconditioner

Performance Results (MAGMA)

- [Haidar, Tomov, Dongarra, Higham, 2018]



(b) Matrix of type 4: clustered singular values, σ_i=(1, $\cdots$, 1, $\frac{1}{cond}$).

Comments and Caveats II

- 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 (normal) $\tilde{A}$ 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 [C., Oktay, 2022], using additional preconditioner

Comments and Caveats II

- 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 (normal) $\tilde{A}$ 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 [C., Oktay, 2022], using additional preconditioner

- Depending on conditioning of A, applying $\tilde{A}$ to a vector must be done accurately (precision $\boldsymbol{u}^2$) in each GMRES iteration
  - Recent development of 5-precision GMRES-IR algorithm [Amestoy et al., 2021]
    - For GMRES entirely in precision $\boldsymbol{u}$,

$$\kappa_\infty(A) \leq \boldsymbol{u}^{-1/2}\,\boldsymbol{u_f}^{-1} \quad \rightarrow \quad \kappa_\infty(A) \leq \boldsymbol{u}^{-1/3}\,\boldsymbol{u_f}^{\boldsymbol{-2/3}}$$

Comments and Caveats II

- 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 (normal) $\tilde{A}$ 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 [C., Oktay, 2022], using additional preconditioner

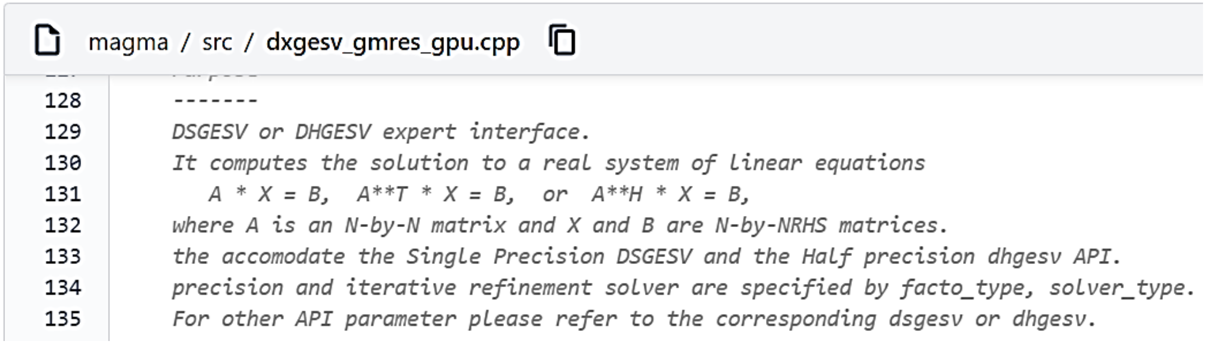
- Depending on conditioning of A, applying $\tilde{A}$ to a vector must be done accurately (precision $\boldsymbol{u}^2$) in each GMRES iteration
  - Recent development of 5-precision GMRES-IR algorithm [Amestoy et al., 2021]
    - For GMRES entirely in precision $\boldsymbol{u}$,

$$\kappa_\infty(A) \le \boldsymbol{u}^{-1/2}\,\boldsymbol{u}_{\boldsymbol{f}}^{-1} \quad \rightarrow \quad \kappa_\infty(A) \le \boldsymbol{u}^{-1/3}\,\boldsymbol{u}_{\boldsymbol{f}}^{\boldsymbol{-2/3}}$$

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

GMRES-IR in Libraries and Applications

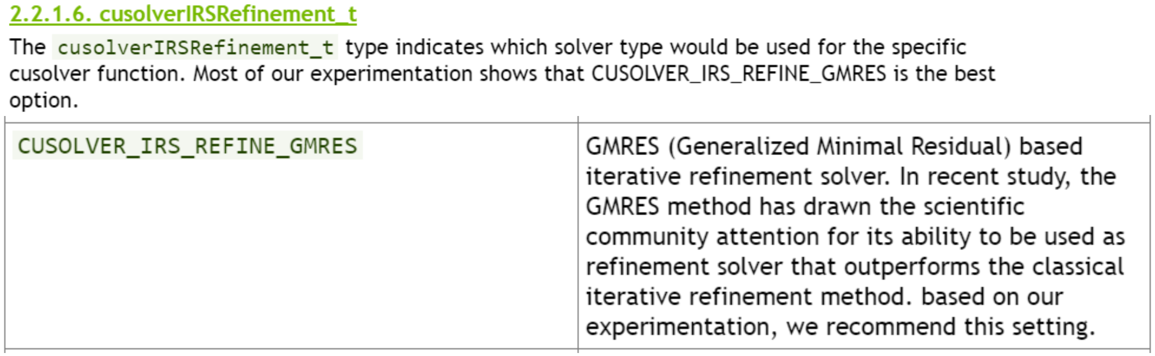
- MAGMA: Dense linear algebra routines for heterogeneous/hybrid architectures



magma / src / dxgesv_gmres_gpu.cpp

```
    -------
    DSGESV or DHGESV expert interface.
    It computes the solution to a real system of linear equations
       A * X = B,  A**T * X = B,  or  A**H * X = B,
    where A is an N-by-N matrix and X and B are N-by-NRHS matrices.
    the accomodate the Single Precision DSGESV and the Half precision dhgesv API.
    precision and iterative refinement solver are specified by facto_type, solver_type.
    For other API parameter please refer to the corresponding dsgesv or dhgesv.
```

- NVIDIA's cuSOLVER Library



2.2.1.6. cusolverIRSRefinement_t

The `cusolverIRSRefinement_t` type indicates which solver type would be used for the specific cusolver function. Most of our experimentation shows that CUSOLVER_IRS_REFINE_GMRES is the best option.

`CUSOLVER_IRS_REFINE_GMRES`	GMRES (Generalized Minimal Residual) based iterative refinement solver. In recent study, the GMRES method has drawn the scientific community attention for its ability to be used as refinement solver that outperforms the classical iterative refinement method. based on our experimentation, we recommend this setting.

- In production codes: FK6D/ASGarD code (Oak Ridge National Lab, USA) for tokomak containment problem

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!**

See [C., Higham, Pranesh, 2020]

GMRES-IR with Inexact Preconditioners

- Existing analyses of GMRES-IR assume we use full LU factors
- In practice, often want to use approximate preconditioners (ILU, SPAI, etc.)

GMRES-IR with Inexact Preconditioners

- Existing analyses of GMRES-IR assume we use full LU factors
- In practice, often want to use approximate preconditioners (ILU, SPAI, etc.)

- [Amestoy et al., 2022]
  - Analysis of **block low-rank (BLR) LU** within GMRES-IR
  - Analysis of use of **static pivoting** in LU within GMRES-IR

GMRES-IR with Inexact Preconditioners

- Existing analyses of GMRES-IR assume we use full LU factors
- In practice, often want to use approximate preconditioners (ILU, SPAI, etc.)

- [Amestoy et al., 2022]
  - Analysis of **block low-rank (BLR) LU** within GMRES-IR
  - Analysis of use of **static pivoting** in LU within GMRES-IR

- [C., Khan, 2022]
  - Analysis of **sparse approximate inverse (SPAI) preconditioners** within GMRES-IR

SPAI Preconditioners

Goal: Construct sparse matrix $M \approx A^{-1}$ (for survey see [Benzi, 2002])

Approach of [Grote, Huckle, 1997]: Construct columns m_k of M dynamically

```
Given matrix $A$, initial sparsity structure $J$, and tolerance $\varepsilon$
For each column $k$:
    Compute QR factorization of submatrix of $A$ defined by $J$
    Use QR factorization to solve $\min_{m_k} \|e_k - Am_k\|_2$
    If $\|r_k\|_2 = \|e_k - Am_k\|_2 \leq \varepsilon$
        break;
    Else
        add select nonzeros to $J$, repeat.
```

SPAI Preconditioners

Goal: Construct sparse matrix $M \approx A^{-1}$ (for survey see [Benzi, 2002])

Approach of [Grote, Huckle, 1997]: Construct columns m_k of M dynamically

```
Given matrix $A$, initial sparsity structure $J$, and tolerance $\varepsilon$
For each column $k$:
    Compute QR factorization of submatrix of $A$ defined by $J$
    Use QR factorization to solve $\min_{m_k} \|e_k - Am_k\|_2$
    If $\|r_k\|_2 = \|e_k - Am_k\|_2 \leq \varepsilon$
        break;
    Else
        add select nonzeros to $J$, repeat.
```

Benefits: Highly parallelizable

But construction can still be costly, esp. for large-scale problems

[Gao, Chen, He, 2021], [Chao, 2001], [Benzi, Tůma, 1999], [He, Yin, Gao, 2020]

SPAI Preconditioners in Low Precision

What is the effect of using low precision in SPAI construction?

Notes and assumptions:

- We will assume that the SPAI construction is performed in some precision $\boldsymbol{u_f}$
- We will denote quantities computed in finite precision with hats
- In our application, we want a left preconditioner, so we will run the algorithm on A^T and set $M \leftarrow M^T$.
- We will assume that the QR factorization of the submatrix of A^T is computed fully using HouseholderQR/TSQR

SPAI Preconditioners in Low Precision

Two interesting questions:

1. Assuming we impose no maximum sparsity pattern on $\widehat{M}$, under what constraint on $\boldsymbol{u_f}$ can we guarantee that $\|\hat{r}_k\|_2 \leq \boldsymbol{\varepsilon}$, with $\hat{r}_k = fl_{\boldsymbol{u_f}}(e_k - A^T \widehat{m}_k^T)$ for the computed $\widehat{m}_k^T$?

SPAI Preconditioners in Low Precision

Two interesting questions:

1. Assuming we impose no maximum sparsity pattern on $\widehat{M}$, under what constraint on $\boldsymbol{u_f}$ can we guarantee that $\|\hat{r}_k\|_2 \leq \boldsymbol{\varepsilon}$, with $\hat{r}_k = fl_{\boldsymbol{u_f}}(e_k - A^T \widehat{m}_k^T)$ for the computed $\widehat{m}_k^T$?

2. Assume that when M is computed in exact arithmetic, we quit as soon as $\|r_k\| \leq \boldsymbol{\varepsilon}$. For $\widehat{M}$ computed in precision $\boldsymbol{u_f}$ with the same sparsity pattern as M, what is $\left\|e_k - A^T \widehat{m}_k^T\right\|_2$?

SPAI Preconditioning in Low Precision

Using standard rounding error analysis and perturbation results for LS problems, we have

$$\|\hat{r}_k\|_2 \leq n^3 \boldsymbol{u_f} \Big\| |e_k| + |A^T| \big|\widehat{m}_k^T\big| \Big\|_2.$$

So in order to guarantee we eventually reach a solution with $\|\hat{r}_k\|_2 \leq \boldsymbol{\varepsilon}$, we need

$$n^3 \boldsymbol{u_f} \Big\| |e_k| + |A^T| \big|\widehat{m}_k^T\big| \Big\|_2 \leq \boldsymbol{\varepsilon}.$$

SPAI Preconditioning in Low Precision

Using standard rounding error analysis and perturbation results for LS problems, we have

$$\|\hat{r}_k\|_2 \leq n^3 \boldsymbol{u_f} \big\| |e_k| + |A^T| |\widehat{m}_k^T| \big\|_2.$$

So in order to guarantee we eventually reach a solution with $\|\hat{r}_k\|_2 \leq \boldsymbol{\varepsilon}$, we need

$$n^3 \boldsymbol{u_f} \big\| |e_k| + |A^T| |\widehat{m}_k^T| \big\|_2 \leq \boldsymbol{\varepsilon}.$$

→ problem must not be so ill-conditioned WRT $\boldsymbol{u_f}$ that we incur an error greater than $\boldsymbol{\varepsilon}$ just computing the residual

SPAI Preconditioning in Low Precision

Can turn this into the looser but more descriptive a priori bound:

$$\operatorname{cond}_2(A^T) \lesssim \boldsymbol{\varepsilon} \boldsymbol{u_f}^{-1},$$

where $\operatorname{cond}_2(A^T) = \||A^{-T}||A^T|\|_2$.

SPAI Preconditioning in Low Precision

Can turn this into the looser but more descriptive a priori bound:

$$\mathrm{cond}_2(A^T) \lesssim \boldsymbol{\varepsilon} \boldsymbol{u}_{\boldsymbol{f}}^{-1},$$

where $\mathrm{cond}_2(A^T) = \| |A^{-T}||A^T| \|_2$.

Another view: with a given matrix A and a given precision $\boldsymbol{u}_{\boldsymbol{f}}$, one must set $\boldsymbol{\varepsilon}$ such that

$$\boldsymbol{\varepsilon} \geq \boldsymbol{u}_{\boldsymbol{f}} \mathrm{cond}_2(A^T).$$

Confirms intuition: **The more approximate the inverse, the lower the precision we can use**.

SPAI Preconditioning in Low Precision

Can turn this into the looser but more descriptive a priori bound:

$$\mathrm{cond}_2(A^T) \lesssim \boldsymbol{\varepsilon} \boldsymbol{u}_{\boldsymbol{f}}^{-1},$$

where $\mathrm{cond}_2(A^T) = \| |A^{-T}| |A^T| \|_2$.

Another view: with a given matrix A and a given precision $\boldsymbol{u}_{\boldsymbol{f}}$, one must set $\boldsymbol{\varepsilon}$ such that

$$\boldsymbol{\varepsilon} \geq \boldsymbol{u}_{\boldsymbol{f}} \mathrm{cond}_2(A^T).$$

Confirms intuition: **The more approximate the inverse, the lower the precision we can use**.

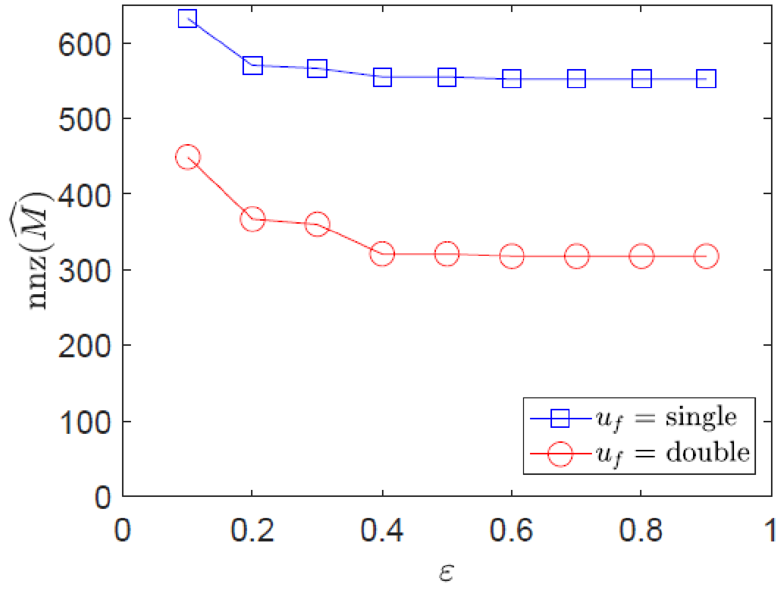
Resulting bounds for $\widehat{M}$:

$$\left\| I - A^T \widehat{M}^T \right\|_F \leq 2\sqrt{n}\boldsymbol{\varepsilon}, \qquad \left\| I - \widehat{M} A \right\|_\infty \leq 2n\boldsymbol{\varepsilon}$$

Size of SPAI Preconditioner in Low Precision

How does precision used affect the number of nonzeros in $\widehat{M}$?

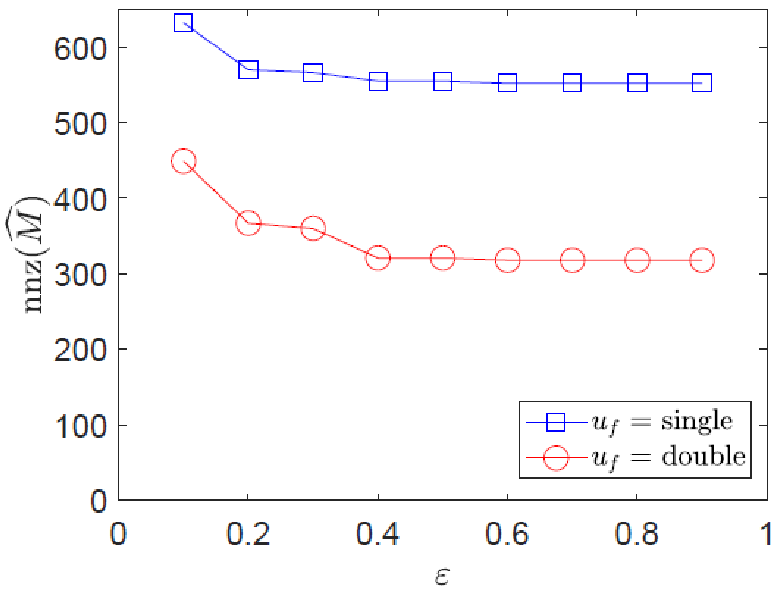
steam3



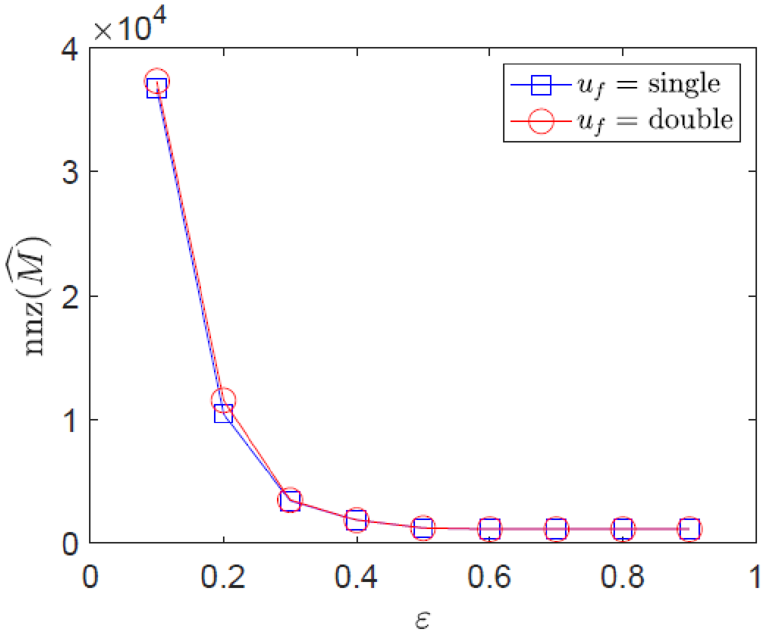
Size of SPAI Preconditioner in Low Precision

How does precision used affect the number of nonzeros in $\widehat{M}$?

steam3



saylr1



Second Question

Assume that when M is computed in exact arithmetic, we quit as soon as $\|r_k\| \leq \boldsymbol{\varepsilon}$. For $\widehat{M}$ computed in precision $\boldsymbol{u_f}$ with the same sparsity pattern as M, what is $\left\|e_k - A^T \widehat{m}_k^T\right\|_2$?

Second Question

Assume that when M is computed in exact arithmetic, we quit as soon as $\|r_k\| \leq \boldsymbol{\varepsilon}$. For $\widehat{M}$ computed in precision $\boldsymbol{u_f}$ with the same sparsity pattern as M, what is $\left\|e_k - A^T \widehat{m}_k^T\right\|_2$?

In this case, we obtain the bound

$$\left\|I - \widehat{M}A\right\|_\infty \leq n\left(\boldsymbol{\varepsilon} + n^{7/2}\boldsymbol{u_f}\kappa_\infty(A)\right).$$

→ If $\kappa_\infty(A) \gg \boldsymbol{\varepsilon}\boldsymbol{u_f}^{-1}$, then computed $\widehat{M}$ with same sparsity structure as M can be of much lower quality.

SPAI-GMRES-IR

SPAI-GMRES-IR

To compute the updates d_i, apply GMRES to $\widehat{M}Ad_i = \widehat{M}r_i$

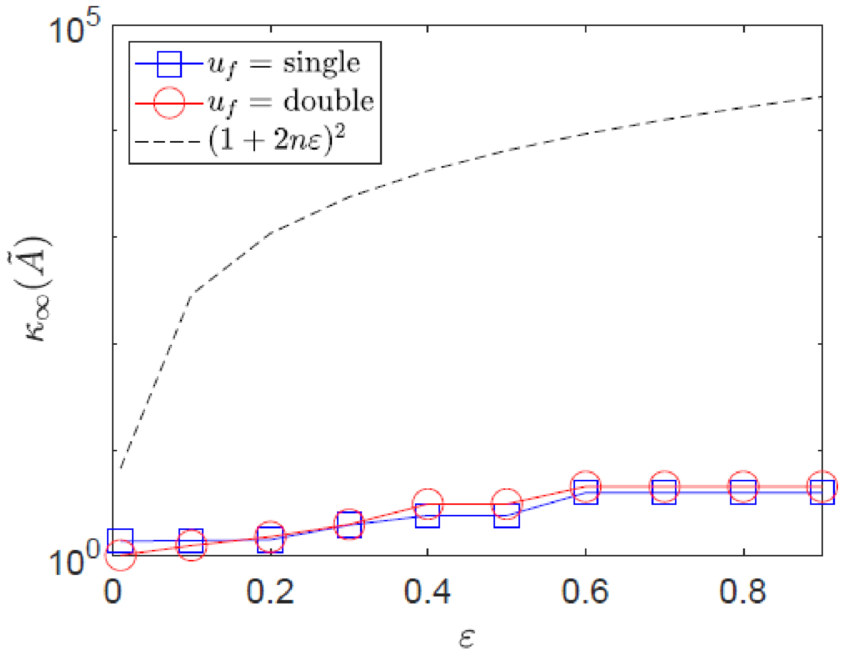
```
Solve $\widehat{M}Ax_0 = \widehat{M}b$
for $i = 0$: maxit
        $r_i = b - Ax_i$
        Solve $Ad_i = r_i$   via GMRES on $\widehat{M}Ad_i = \widehat{M}r_i$
        $x_{i+1} = x_i + d_i$
```

Low Precision SPAI within GMRES-IR

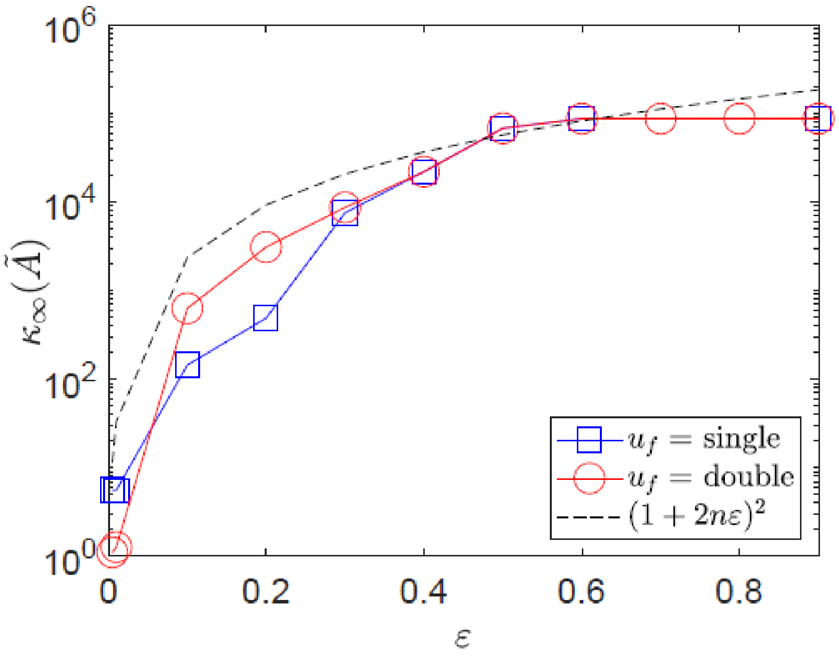
Using $\widehat{M}$ computed in precision $\boldsymbol{u_f}$, for the preconditioned system $\tilde{A} = \widehat{M}A$,

$$\kappa_\infty(\tilde{A}) \lesssim (1 + 2n\boldsymbol{\varepsilon})^2.$$

steam3



saylr1



Low Precision SPAI within GMRES-IR

To guarantee that both SPAI construction will complete and the GMRES-based iterative refinement scheme will converge, we must have roughly

$$n\boldsymbol{u_f}\,\mathrm{cond}_2(A^T) \lesssim n\boldsymbol{\varepsilon} \lesssim \boldsymbol{u}^{-1/2}.$$

Low Precision SPAI within GMRES-IR

To guarantee that both SPAI construction will complete and the GMRES-based iterative refinement scheme will converge, we must have roughly

$$\underbrace{n\boldsymbol{u}_{\boldsymbol{f}}\mathrm{cond}_2(A^T) \lesssim n\boldsymbol{\varepsilon}}_{\widehat{M}\text{ can be constructed}} \lesssim \boldsymbol{u}^{-1/2}.$$

Low Precision SPAI within GMRES-IR

To guarantee that both SPAI construction will complete and the GMRES-based iterative refinement scheme will converge, we must have roughly

$$n\boldsymbol{u_f}\,\mathrm{cond}_2(A^T) \lesssim n\boldsymbol{\varepsilon} \lesssim \boldsymbol{u}^{-1/2}.$$

$\widehat{M}$ can be constructed

$\widehat{M}$ is a good enough preconditioner

Low Precision SPAI within GMRES-IR

To guarantee that both SPAI construction will complete and the GMRES-based iterative refinement scheme will converge, we must have roughly

$$n\boldsymbol{u_f}\,\mathrm{cond}_2(A^T) \lesssim n\boldsymbol{\varepsilon} \lesssim \boldsymbol{u}^{-1/2}.$$

$\widehat{M}$ can be constructed

$\widehat{M}$ is a good enough preconditioner

If ε satisfies these constraints, then the constraints on condition number for forward and backward errors to converge are the same as for GMRES-IR with full LU factorization.

Low Precision SPAI within GMRES-IR

To guarantee that both SPAI construction will complete and the GMRES-based iterative refinement scheme will converge, we must have roughly

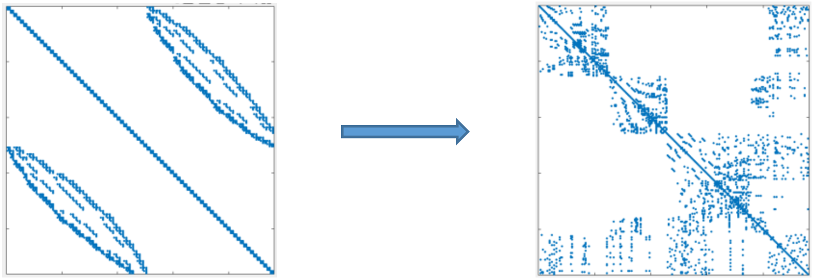
$$\underbrace{n \boldsymbol{u_f} \operatorname{cond}_2(A^T) \lesssim \mathrlap{n\boldsymbol{\varepsilon}}}_{\widehat{M}\text{ can be constructed}}\underbrace{\phantom{n\boldsymbol{\varepsilon}} \lesssim \boldsymbol{u}^{-1/2}.}_{\widehat{M}\text{ is a good enough preconditioner}}$$

If ε satisfies these constraints, then the constraints on condition number for forward and backward errors to converge are the same as for GMRES-IR with full LU factorization.

Compared to GMRES-IR with full LU factorization, in general expect **slower convergence, but much sparser preconditioner**.

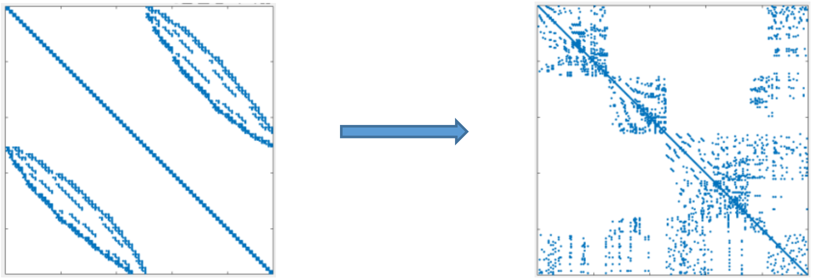
SPAI-GMRES-IR Example

Matrix: steam1, $n = 240$, nnz $= 2{,}248$, $\kappa_\infty(A) = 3 \cdot 10^7$, cond$(A^T) = 3 \cdot 10^3$

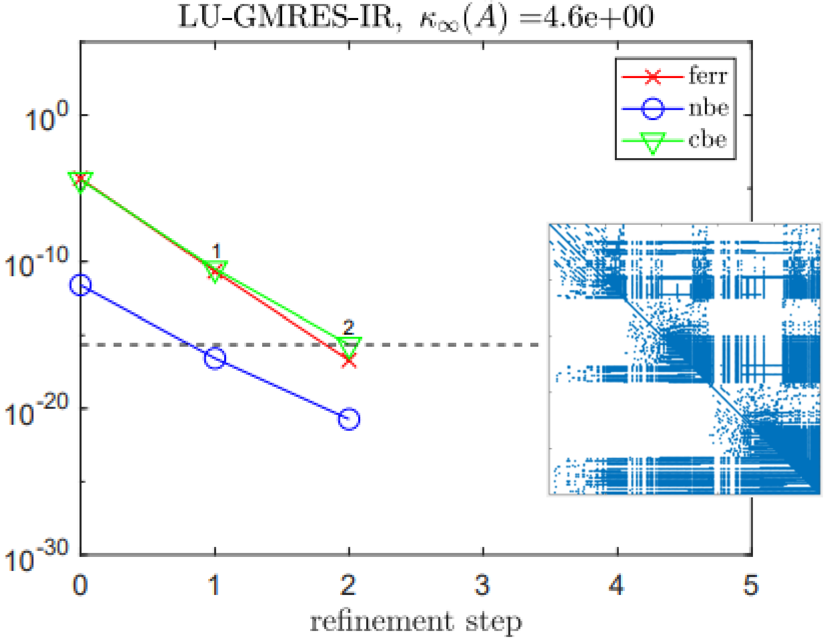


SPAI-GMRES-IR Example

Matrix: steam1, $n = 240$, nnz $= 2{,}248$, $\kappa_\infty(A) = 3 \cdot 10^7$, cond$(A^T) = 3 \cdot 10^3$



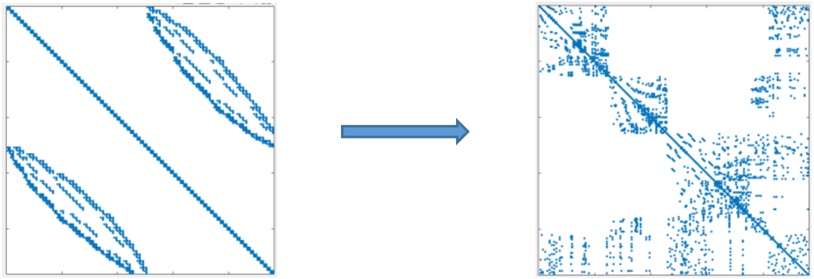
$(\boldsymbol{u_f}, \boldsymbol{u}, \boldsymbol{u_r})$ = (single, double, quad)



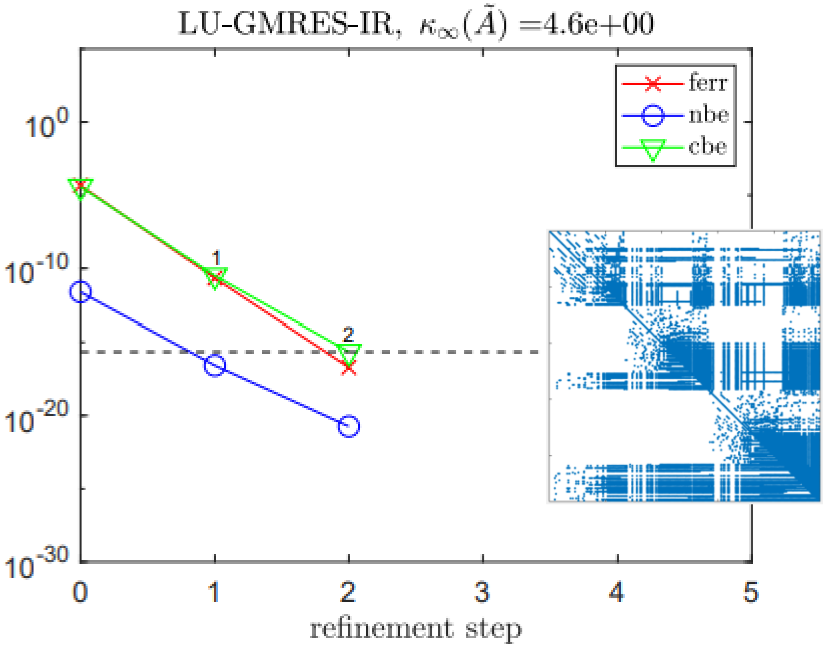
nnz$(L + U) = 13{,}765$

SPAI-GMRES-IR Example

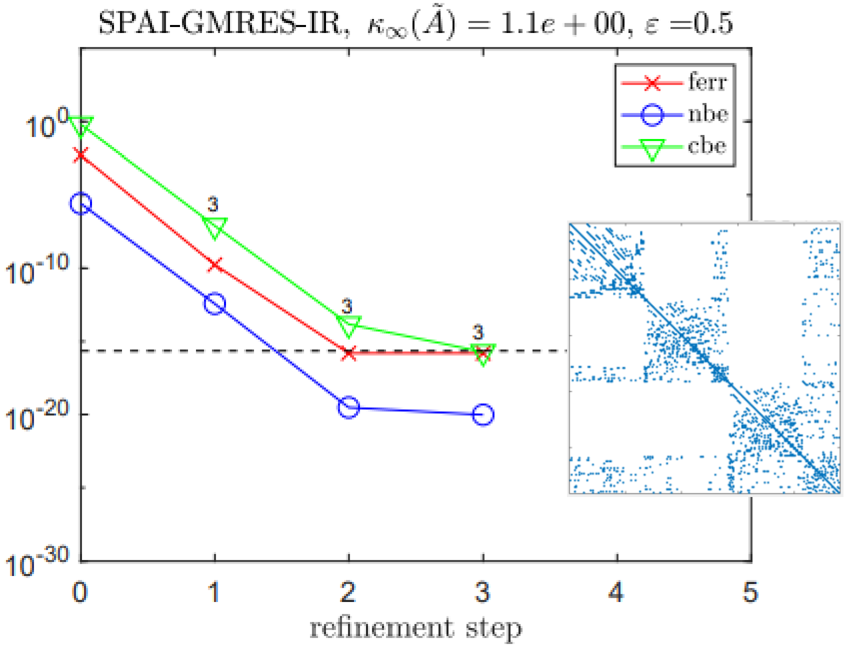
Matrix: steam1, $n = 240$, nnz $= 2{,}248$, $\kappa_\infty(A) = 3 \cdot 10^7$, cond$(A^T) = 3 \cdot 10^3$



$(\boldsymbol{u_f}, \boldsymbol{u}, \boldsymbol{u_r})$ = (single, double, quad)



LU-GMRES-IR, $\kappa_\infty(\tilde{A})$ =4.6e+00



SPAI-GMRES-IR, $\kappa_\infty(\tilde{A}) = 1.1e + 00$, ε =0.5

nnz$(L + U) = 13{,}765$

nnz$(M) = 2{,}248$

Is there a point in using precision higher than that dictated by $\boldsymbol{u_f}\text{cond}_2(A^T) \leq \boldsymbol{\varepsilon}$?

Matrix: bfwa782, $n = 782$, nnz $= 7514$, $\kappa_\infty(A) = 7 \cdot 10^3$, $\text{cond}(A^T) = 1 \cdot 10^3$

$(\boldsymbol{u_f}, \boldsymbol{u}, \boldsymbol{u_r})$ = (**half**, single, double)

Preconditioner	$\kappa_\infty(\tilde{A})$	Precond. nnz	GMRES-IR steps/iteration
SPAI ($\boldsymbol{\varepsilon} = 0.2$)	$2.1e + 02$	28053	67 (31, 36)
SPAI ($\boldsymbol{\varepsilon} = 0.5$)	$9.7e + 02$	7528	153 (71, 82)

Is there a point in using precision higher than that dictated by $u_f \mathrm{cond}_2(A^T) \leq \varepsilon$?

Matrix: bfwa782, $n = 782$, nnz $= 7514$, $\kappa_\infty(A) = 7 \cdot 10^3, \mathrm{cond}(A^T) = 1 \cdot 10^3$

$(u_f, u, u_r) =$ (**half**, single, double)

Preconditioner	$\kappa_\infty(\tilde{A})$	Precond. nnz	GMRES-IR steps/iteration
SPAI ($\varepsilon = 0.2$)	$2.1e + 02$	28053	67 (31, 36)
SPAI ($\varepsilon = 0.5$)	$9.7e + 02$	7528	153 (71, 82)

$(u_f, u, u_r) =$ (**single**, single, double)

Preconditioner	$\kappa_\infty(\tilde{A})$	Precond. nnz	GMRES-IR steps/iteration
SPAI ($\varepsilon = 0.2$)	$2.2e + 02$	26801	69 (32, 37)
SPAI ($\varepsilon = 0.5$)	$9.7e + 02$	7529	153 (71, 82)

Related and Current Work

- Multistage mixed precision iterative refinement

  [Oktay, C., 2021]

  If IR not converging, first try changing the solver before increasing precision

- Low-precision randomized preconditioners

  [C., Daužickaitė, 2022]

  Single-pass Nyström can be run in precision $u_p \approx \frac{\lambda_{k+1}}{\sqrt{n}\lambda_1}$ without affecting the quality of limited memory preconditioner.

- Low-precision in ILU-type preconditioners

  What can we prove?

Summary and Takeaway

- We now have a multi-precision ecosystem

- Huge opportunities for using mixed precision in matrix computations

- But also big challenges!

Thank You!

carson@karlin.mff.cuni.cz

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

Key Analysis Innovations I

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

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

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

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

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

Key Analysis Innovations I

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

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

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

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 $\text{span}(U_k)$ for any k s.t. $\sigma_{n+1-k} \approx \sigma_n$

Key Analysis Innovations I

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

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

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

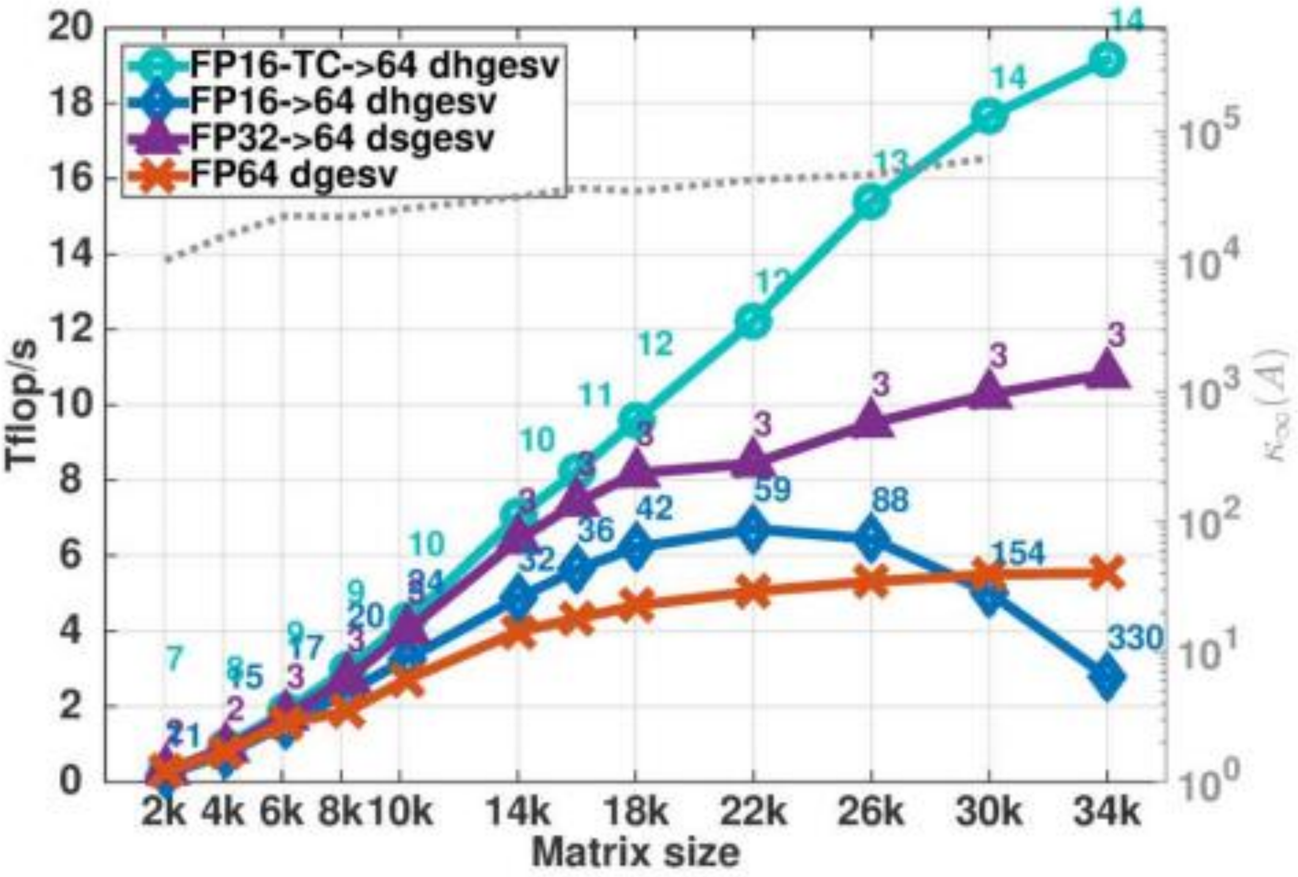
where $P_k = U_k U_k^T, U_k = [u_{n+1-k}, \ldots, u_n]$

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

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

Performance Results (MAGMA)

- [Haidar, Tomov, Dongarra, Higham, 2018]



(b) Matrix of type 4: clustered singular values, σ_i=(1, ⋯, 1, $\frac{1}{cond}$).

Randomized Limited Memory Preconditioners

Let $A \in \mathbb{R}^{n \times n}$ be a symmetric positive semidefinite matrix. Want to solve

$$(A + \mu I)x = b$$

where $\mu \geq 0$ is set so that $A + \mu I$ is positive definite. Assume A has rapidly decreasing eigenvalues or cluster of large eigenvalues.

Randomized Limited Memory Preconditioners

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive semidefinite matrix. Want to solve

$$(A + \mu I)x = b$$

where $\mu \geq 0$ is set so that $A + \mu I$ is positive definite. Assume A has rapidly decreasing eigenvalues or cluster of large eigenvalues.

Want to solve using PCG using **spectral limited memory preconditioner** [Gratton, Sartenaer, Tshimanga, 2011], [Tshimanga et al., 2008]:

$$P = I - UU^T + \tfrac{1}{\alpha+\mu} U(\Theta + \mu I)U^T$$
$$P^{-1} = I - UU^T + (\alpha + \mu)U(\Theta + \mu I)^{-1}U^T$$

where columns of $U \in \mathbb{R}^{n\times k}$ are k approximate eigenvectors of A and $U^T U = I$, Θ is diagonal with approximations to eigenvalues of A, and $\alpha \geq 0$.

Used in data assimilation [Laloyaux et al., 2018], [Mogensen, Alonso Balmaseda, Weaver, 2012], [Moore et al., 2011], [Daužickaitė, Lawless, Scott, van Leeuwen, 2021]

Randomized Nyström Approximation

Want to compute a rank-k approximation $A \approx U\Theta U^T$ via the randomized Nyström method.

Nyström approximation:

$$A_N = (AQ)(Q^TAQ)^+(AQ)^T$$

where Q is an $n \times k$ sampling matrix (random projection).

Randomized Nyström Approximation

In the case that A is very large, matrix-matrix products with A are the bottleneck.

This motivates the single-pass version of the Nyström method.

Stabilized Single-Pass Nyström method [Tropp et al., 2017]

```
Given sym. PSD matrix $A$, target rank $k$
$G = \text{randn}(n, k)$
$[Q, \sim] = \text{qr}(G, 0)$
$\boldsymbol{Y = AQ}$
Compute shift $\nu$; $Y_\nu = Y + \nu Q$
$B = Q^T Y_\nu$
$C = \text{chol}((B + B^T)/2)$
Solve $F = Y_\nu / C$
$[U, \Sigma, \sim] = \text{svd}(F, 0)$
$\Theta = \max(0, \Sigma^2 - \nu I)$
```

Randomized Nyström Approximation

In the case that A is very large, matrix-matrix products with A are the bottleneck.

This motivates the single-pass version of the Nyström method.

Stabilized Single-Pass Nyström method [Tropp et al., 2017]

```
Given sym. PSD matrix $A$, target rank $k$
$G = \text{randn}(n, k)$
$[Q, \sim] = \text{qr}(G, 0)$
$\boldsymbol{Y = AQ}$
Compute shift $\nu$; $Y_\nu = Y + \nu Q$
$B = Q^T Y_\nu$
$C = \text{chol}((B + B^T)/2)$
Solve $F = Y_\nu / C$
$[U, \Sigma, \sim] = \text{svd}(F, 0)$
$\Theta = \max(0, \Sigma^2 - \nu I)$
```

Can we further reduce the cost of the matrix-matrix product with A by using low precision?

Error Bounds

$$\left\|A - \hat{A}_N\right\|_2 = \left\|A - A_N + A_N - \hat{A}_N\right\|_2 \le \underbrace{\|A - A_N\|_2}_{\text{exact approximation error}} + \underbrace{\left\|A_N - \hat{A}_N\right\|_2}_{\text{finite precision error}}$$

Error Bounds

$$\left\|A - \hat{A}_N\right\|_2 = \left\|A - A_N + A_N - \hat{A}_N\right\|_2 \leq \underbrace{\|A - A_N\|_2}_{\text{exact approximation error}} + \underbrace{\left\|A_N - \hat{A}_N\right\|_2}_{\text{finite precision error}}$$

Deterministic bound [Gittens, Mahoney, 2016]:

$$\|A - A_N\|_2 \leq \lambda_{k+1} + \left\|\Sigma_2^{1/2} U_2^T Q (U_1 Q)^+\right\|_2^2$$

with $A = [U_1 \;\; U_2] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} [U_1 \; U_2]^T$.

Error Bounds

$$\left\|A-\hat{A}_N\right\|_2 = \left\|A-A_N+A_N-\hat{A}_N\right\|_2 \le \underbrace{\|A-A_N\|_2}_{\text{exact approximation error}} + \underbrace{\left\|A_N-\hat{A}_N\right\|_2}_{\text{finite precision error}}$$

Deterministic bound [Gittens, Mahoney, 2016]:

$$\boxed{\|A-A_N\|_2 \le \lambda_{k+1} + \left\|\Sigma_2^{1/2}U_2^T Q(U_1 Q)^+\right\|_2^2}$$

with $A = [U_1\ \ U_2]\begin{bmatrix}\Sigma_1 & \\ & \Sigma_2\end{bmatrix}[U_1\ U_2]^T$.

Expected value bound [Frangella, Tropp, Udell, 2021]:

$$\boxed{\mathbb{E}\|A-A_N\|_2 \le \min_{2\le p\le k-2}\left(\left(1+\frac{2(k-p)}{p-1}\right)\lambda_{k-p+1} + \frac{2e^2k}{p^2-1}\sum_{j=k-p+1}^{n}\lambda_j\right)}$$

where $\lambda_i \ge \lambda_{i+1}$ are the eigenvalues of A.

Finite Precision Error Bound

Finite precision error: $A_N - \hat{A}_N$

Assumptions:

- A is stored in precision u_p and matrix-matrix product AQ is computed in precision u_p
- All other quantities stored and computed in precision $u \ll u_p$

Finite Precision Error Bound

Finite precision error: $A_N - \hat{A}_N$

Assumptions:

- A is stored in precision u_p and matrix-matrix product AQ is computed in precision u_p
- All other quantities stored and computed in precision $u \ll u_p$

[C., Daužickaitė, 2022]:

$$\|A_N - \hat{A}_N\|_2 \leq O(u_p) n^{5/2} \|A\|_2$$

Finite Precision Error Bound

Finite precision error: $A_N - \hat{A}_N$

Assumptions:

- A is stored in precision u_p and matrix-matrix product AQ is computed in precision u_p
- All other quantities stored and computed in precision $u \ll u_p$

[C., Daužickaitė, 2022]:

$$\left\| A_N - \hat{A}_N \right\|_2 \leq O\left(u_p\right) n^{5/2} \|A\|_2$$

Interpretation: $\left\| A_N - \hat{A}_N \right\|_2 \gtrsim \|A - A_N\|_2$ when

$$\frac{\lambda_{k+1}}{\lambda_1} \lesssim \sqrt{n} u_p$$

Finite Precision Error Bound

Finite precision error: $A_N - \hat{A}_N$

Assumptions:

- A is stored in precision u_p and matrix-matrix product AQ is computed in precision u_p
- All other quantities stored and computed in precision $u \ll u_p$

[C., Daužickaitė, 2022]:

$$\left\|A_N - \hat{A}_N\right\|_2 \leq O(u_p) n^{5/2} \|A\|_2$$

Interpretation: $\left\|A_N - \hat{A}_N\right\|_2 \gtrsim \|A - A_N\|_2$ when

$$\frac{\lambda_{k+1}}{\lambda_1} \lesssim \sqrt{n} u_p$$

The more approximate the low-rank representation, the lower the precision we can use!

Condition Number Bounds

Let $E = A - A_N$, $\mathcal{E} = A_N - \hat{A}_N$, and assume $(A + \mu I)$ is SPD.

Let

$$\hat{P}^{-1} = I - \widehat{U}\widehat{U}^T + \left(\hat{\lambda}_k + \mu\right)\widehat{U}\left(\widehat{\Theta} + \mu I\right)^{-1}\widehat{U}^T$$

be the LMP preconditioner constructed using the mixed precision Nyström approximation $\hat{A}_N = \widehat{U}\widehat{\Theta}\widehat{U}^T$.

Condition Number Bounds

Let $E = A - A_N$, $\mathcal{E} = A_N - \hat{A}_N$, and assume $(A + \mu I)$ is SPD.

Let

$$\hat{P}^{-1} = I - \widehat{U}\widehat{U}^T + \left(\hat{\lambda}_k + \mu\right)\widehat{U}\left(\widehat{\Theta} + \mu I\right)^{-1}\widehat{U}^T$$

be the LMP preconditioner constructed using the mixed precision Nyström approximation $\hat{A}_N = \widehat{U}\widehat{\Theta}\widehat{U}^T$.

Then

$$\max\left\{1, \frac{\hat{\lambda}_k + \mu - \|\mathcal{E}\|_2}{\mu + \lambda_{min}(A)}\right\} \leq \kappa\left(\hat{P}^{-1/2}(A + \mu I)\hat{P}^{-1/2}\right) \leq 1 + \frac{\hat{\lambda}_k + \|E\|_2 + 2\|\mathcal{E}\|_2}{\mu - \|\mathcal{E}\|_2}$$

where the upper bound holds if $\mu > \|\mathcal{E}\|_2$.

Regardless of this constraint, if A is positive definite, then

$$\kappa\left(\hat{P}^{-1/2}(A + \mu I)\hat{P}^{-1/2}\right) \leq \left(\hat{\lambda}_k + \mu + \|E\|_2 + \|\mathcal{E}\|_2\right)\left(\frac{1}{\hat{\lambda}_k + \mu} + \frac{\|\mathcal{E}\|_2 + 1}{\lambda_{min}(A) + \mu}\right).$$

Condition Number Bounds

Let $E = A - A_N$, $\mathcal{E} = A_N - \hat{A}_N$, and assume $(A + \mu I)$ is SPD.

Let

$$\hat{P}^{-1} = I - \widehat{U}\widehat{U}^T + \left(\hat{\lambda}_k + \mu\right)\widehat{U}\left(\widehat{\Theta} + \mu I\right)^{-1}\widehat{U}^T$$

be the LMP preconditioner constructed using the mixed precision Nyström approximation $\hat{A}_N = \widehat{U}\widehat{\Theta}\widehat{U}^T$.

If $\mathcal{E} = 0$, reduces to bounds of [Frangella, Tropp, Udell, 2021] for exact case.

Then

$$\max\left\{1, \frac{\hat{\lambda}_k + \mu - \|\mathcal{E}\|_2}{\mu + \lambda_{min}(A)}\right\} \leq \kappa\left(\hat{P}^{-1/2}(A + \mu I)\hat{P}^{-1/2}\right) \leq 1 + \frac{\hat{\lambda}_k + \|E\|_2 + 2\|\mathcal{E}\|_2}{\mu - \|\mathcal{E}\|_2}$$

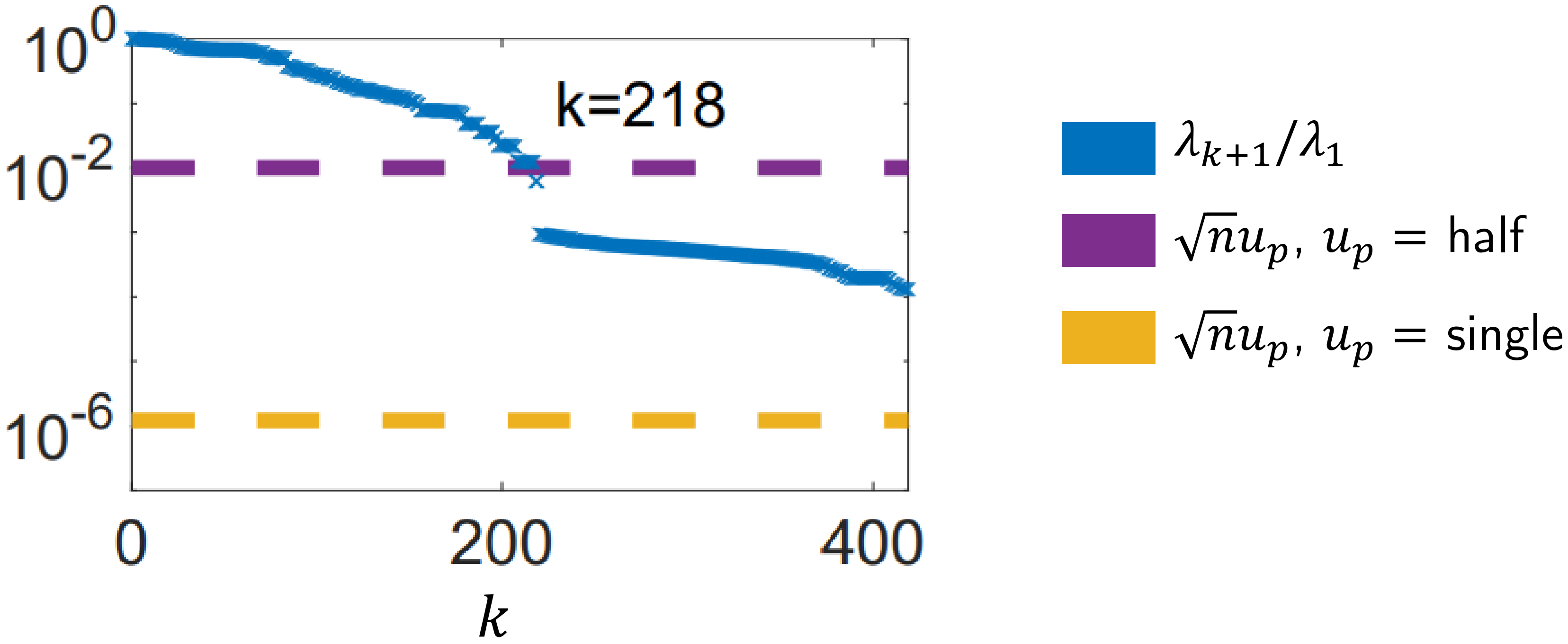
where the upper bound holds if $\mu > \|\mathcal{E}\|_2$.

Regardless of this constraint, if A is positive definite, then

$$\kappa\left(\hat{P}^{-1/2}(A + \mu I)\hat{P}^{-1/2}\right) \leq \left(\hat{\lambda}_k + \mu + \|E\|_2 + \|\mathcal{E}\|_2\right)\left(\frac{1}{\hat{\lambda}_k + \mu} + \frac{\|\mathcal{E}\|_2 + 1}{\lambda_{min}(A) + \mu}\right).$$

Numerical Experiment

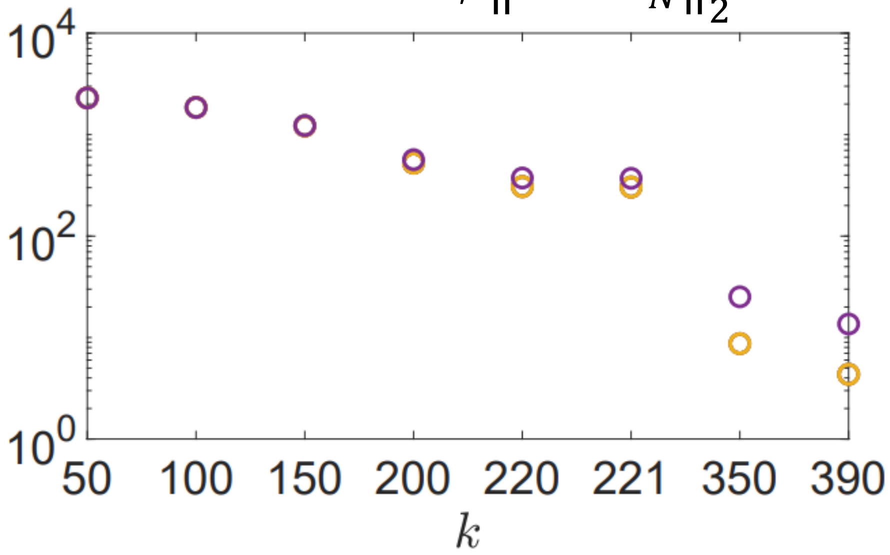
Matrix: bcsstm07, $n = 420$



Numerical Experiment

Matrix: bcsstm07, $n = 420$

total error, $\|A - \hat{A}_N\|_2$



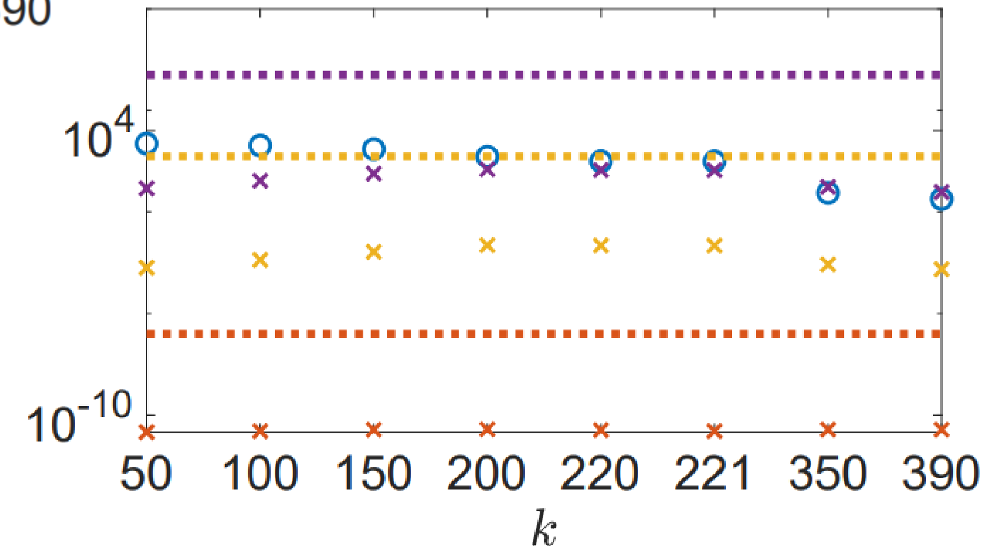
🟦 exact

🟪 mixed, u_p = half

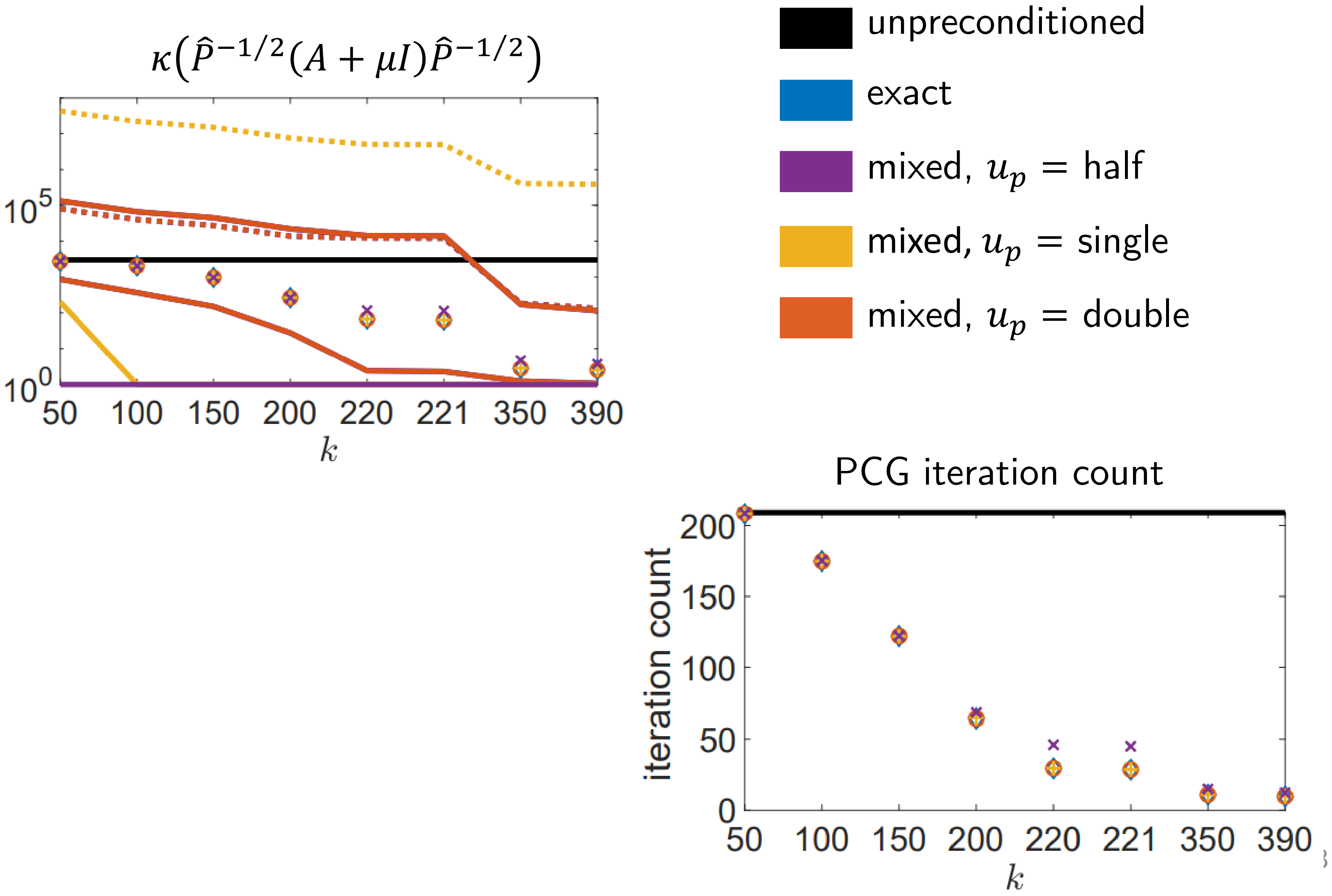
🟨 mixed, u_p = single

🟧 mixed, u_p = double

mean finite prec. error, $\|A_N - \hat{A}_N\|_2$



Numerical Experiment



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?

- Note that augmented system is a saddle-point system; lots of existing work (block diagonal, triangular, constraint-based, ...)

GMRES-IR for Least Squares

- Ex: block diagonal preconditioner ([Murphy, Golub, Wathen, 2000], [Ipsen, 2001])

$$\begin{bmatrix} \alpha I & 0 \\ 0 & \frac{1}{\alpha}\hat{R}^T\hat{R} \end{bmatrix} = \begin{bmatrix} \sqrt{\alpha} I & 0 \\ 0 & \frac{1}{\sqrt{\alpha}}\hat{R}^T \end{bmatrix} \begin{bmatrix} \sqrt{\alpha} I & 0 \\ 0 & \frac{1}{\sqrt{\alpha}}\hat{R} \end{bmatrix} \equiv M_1 M_2$$

GMRES-IR for Least Squares

- Ex: block diagonal preconditioner ([Murphy, Golub, Wathen, 2000], [Ipsen, 2001])

$$\begin{bmatrix} \alpha I & 0 \\ 0 & \frac{1}{\alpha}\hat{R}^T\hat{R} \end{bmatrix} = \begin{bmatrix} \sqrt{\alpha} I & 0 \\ 0 & \frac{1}{\sqrt{\alpha}}\hat{R}^T \end{bmatrix} \begin{bmatrix} \sqrt{\alpha} I & 0 \\ 0 & \frac{1}{\sqrt{\alpha}}\hat{R} \end{bmatrix} \equiv M_1 M_2$$

- Assuming QR factorization is exact,

$$M_2{}^{-1} M_1^{-1} \tilde{A} = \begin{bmatrix} I & \frac{1}{\alpha} A \\ \alpha\, \hat{R}^{-1}\hat{R}^{-T}A^T & 0 \end{bmatrix}$$

is nonsymmetric, diagonalizable, with eigenvalues $\left\{1, \frac{1}{2}\left(1 \pm \sqrt{5}\right)\right\}$.

  - However, condition number can still be quite large; unsuitable for proving backward stability of GMRES

GMRES-IR for Least Squares

- Ex: block diagonal preconditioner ([Murphy, Golub, Wathen, 2000], [Ipsen, 2001])

$$\begin{bmatrix} \alpha I & 0 \\ 0 & \frac{1}{\alpha}\hat{R}^T\hat{R} \end{bmatrix} = \begin{bmatrix} \sqrt{\alpha} I & 0 \\ 0 & \frac{1}{\sqrt{\alpha}}\hat{R}^T \end{bmatrix} \begin{bmatrix} \sqrt{\alpha} I & 0 \\ 0 & \frac{1}{\sqrt{\alpha}}\hat{R} \end{bmatrix} \equiv M_1 M_2$$

- Assuming QR factorization is exact,

$$M_2{}^{-1} M_1^{-1} \tilde{A} = \begin{bmatrix} I & \frac{1}{\alpha} A \\ \alpha\, \hat{R}^{-1}\hat{R}^{-T}A^T & 0 \end{bmatrix}$$

is nonsymmetric, diagonalizable, with eigenvalues $\left\{1, \frac{1}{2}\left(1 \pm \sqrt{5}\right)\right\}$.

  - However, condition number can still be quite large; unsuitable for proving backward stability of GMRES

- If we take split preconditioner

$$M_1^{-1} \tilde{A} M_2{}^{-1} = \begin{bmatrix} I & A\hat{R} \\ \hat{R}^{-T}A^T & 0 \end{bmatrix}$$

we will have a well-conditioned system

  - However, split-preconditioned GMRES is not backward stable
  - Potentially useful in practice, not but in theory

GMRES-IR for Least Squares

- One option:

$$M = \begin{bmatrix} \alpha I & \hat{Q}_1 \hat{R} \\ \hat{R}^T \hat{Q}_1^T & 0 \end{bmatrix}$$

- Then we can prove that for the left-preconditioned system,

$$\kappa\left(M^{-1}\tilde{A}\right) \leq \left(1 + \boldsymbol{u_f} c\, \kappa(A)\right)^2$$

where $c = O(m^2)$, where we note this bound is pessimistic.

- Thus even if $\kappa(A) \gg \boldsymbol{u_f}^{-1}$, the preconditioned system can still be reasonably well conditioned

GMRES-IR for Least Squares

- One option:

$$M = \begin{bmatrix} \alpha I & \hat{Q}_1\hat{R} \\ \hat{R}^T\hat{Q}_1^T & 0 \end{bmatrix}$$

- Then we can prove that for the left-preconditioned system,

$$\kappa\left(M^{-1}\tilde{A}\right) \leq \left(1 + \boldsymbol{u_f} c\,\kappa(A)\right)^2$$

where $c = O(m^2)$, where we note this bound is pessimistic.

- Thus even if $\kappa(A) \gg \boldsymbol{u_f}^{-1}$, the preconditioned system can still be reasonably well conditioned
- GMRES run on $\tilde{A}$ with left-preconditioner M gives

$$\boldsymbol{u_s}\|E_i\|_\infty \equiv \boldsymbol{u}\,f(m+n)\kappa_\infty(M^{-1}\tilde{A})$$

where f is a quadratic polynomial

GMRES-IR for Least Squares

- One option:

$$M = \begin{bmatrix} \alpha I & \hat{Q}_1 \hat{R} \\ \hat{R}^T \hat{Q}_1^T & 0 \end{bmatrix}$$

- Then we can prove that for the left-preconditioned system,

$$\kappa\left(M^{-1}\tilde{A}\right) \leq \left(1 + \boldsymbol{u_f} c\, \kappa(A)\right)^2$$

where $c = O(m^2)$, where we note this bound is pessimistic.

- Thus even if $\kappa(A) \gg \boldsymbol{u}_{\boldsymbol{f}}^{-1}$, the preconditioned system can still be reasonably well conditioned
- GMRES run on $\tilde{A}$ with left-preconditioner M gives

$$\boldsymbol{u_s} \|E_i\|_\infty \equiv \boldsymbol{u}\, f(m+n)\kappa_\infty(M^{-1}\tilde{A})$$

where f is a quadratic polynomial

- So for GMRES-based LSIR, $\boldsymbol{u_s} \equiv \boldsymbol{u}$; expect convergence of forward error when $\kappa_\infty(A) < \boldsymbol{u}^{-1/2}\boldsymbol{u}_{\boldsymbol{f}}^{-1}$

[C., Higham, Pranesh, SISC 2020]