

Balancing Inexactness in Matrix Computations

Erin C. Carson
Charles University

Computational Mathematics and Applications Seminar
Mathematical Institute, University of Oxford
May 25, 2023

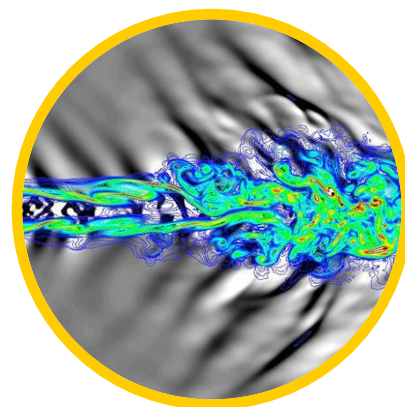
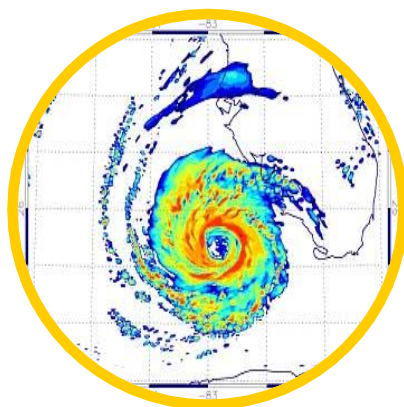
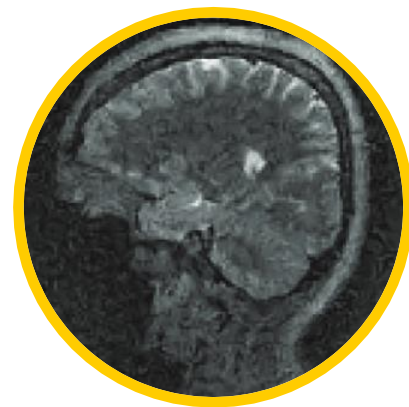
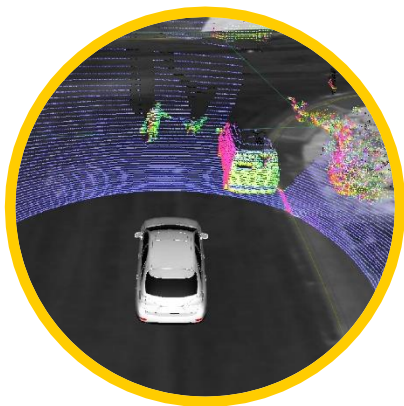
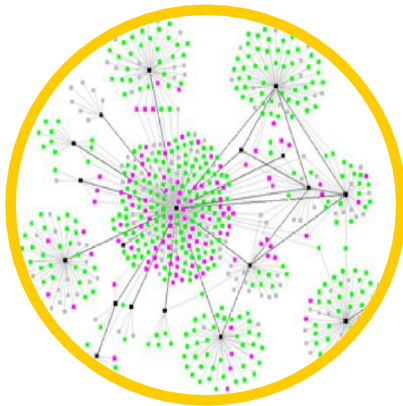


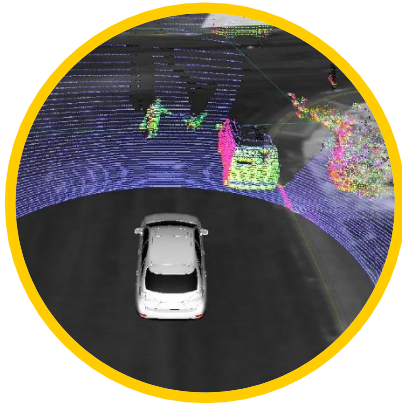
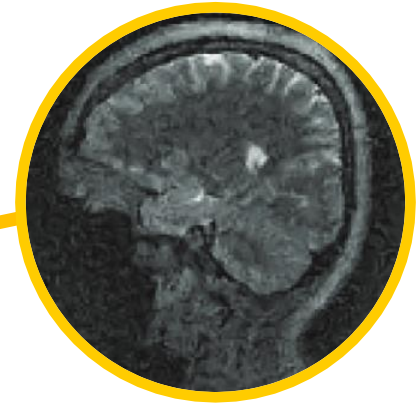
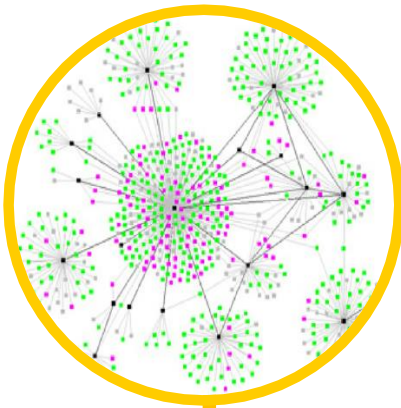
FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University



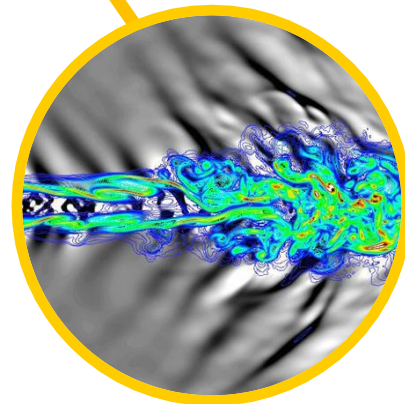
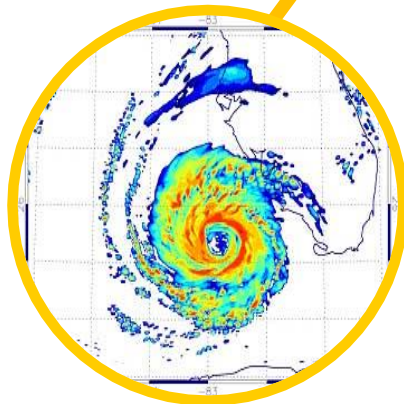
Co-funded by the
European Union

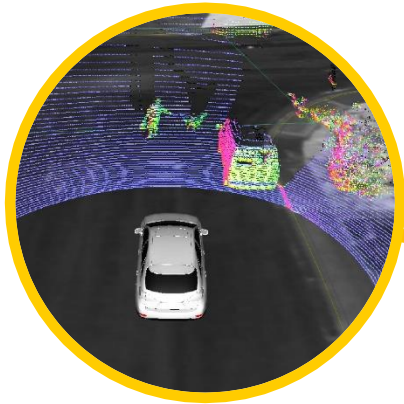
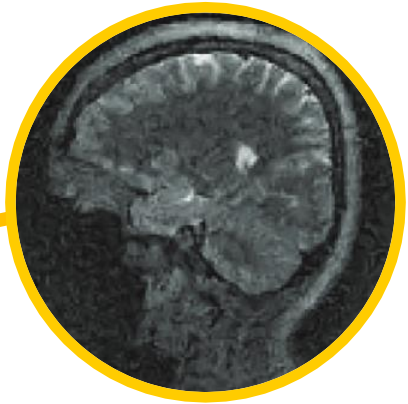
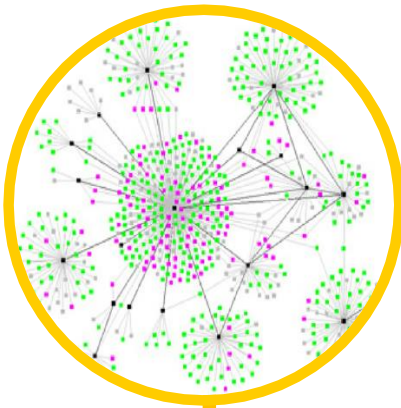
We acknowledge funding from ERC Starting Grant No. 101075632 and the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Admin. Views and opinions expressed are however those of the author only and do not necessarily reflect those of the European Union or the ERC. Neither the European Union nor the granting authority can be held responsible for them.



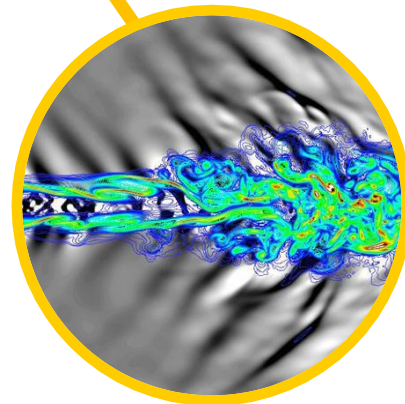
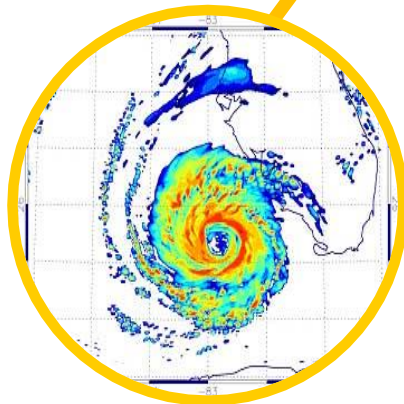


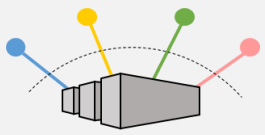
matrix
computations





finite precision
matrix
computations

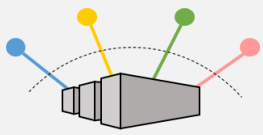




The Exascale Era

We have now entered the “Exascale Era”

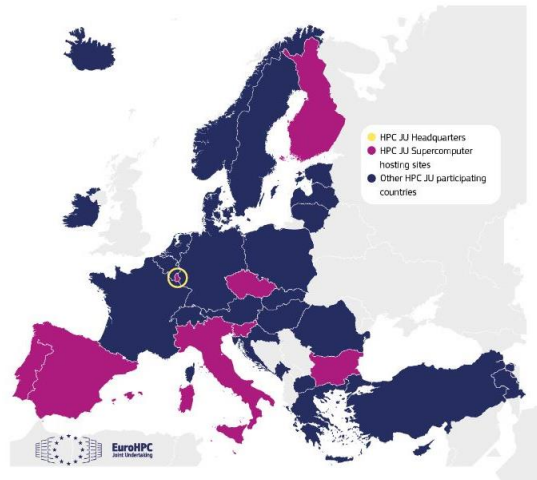
- 10^{18} floating point operations per second



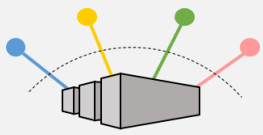
The Exascale Era

We have now entered the “Exascale Era”

- 10^{18} floating point operations per second



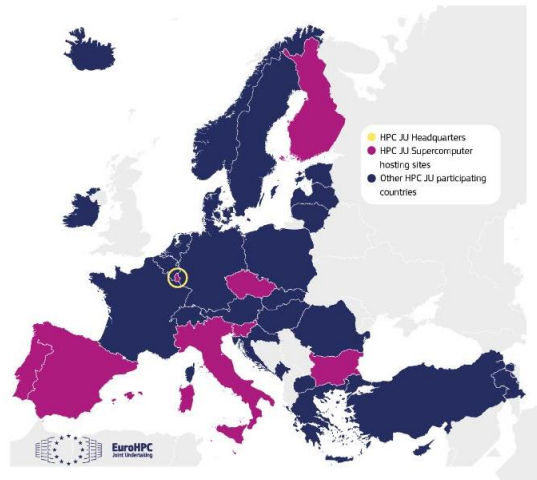
<https://eurohpc-ju.europa.eu/pictures>



The Exascale Era

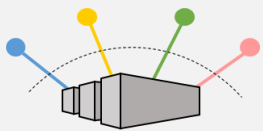
We have now entered the “Exascale Era”

- 10^{18} floating point operations per second

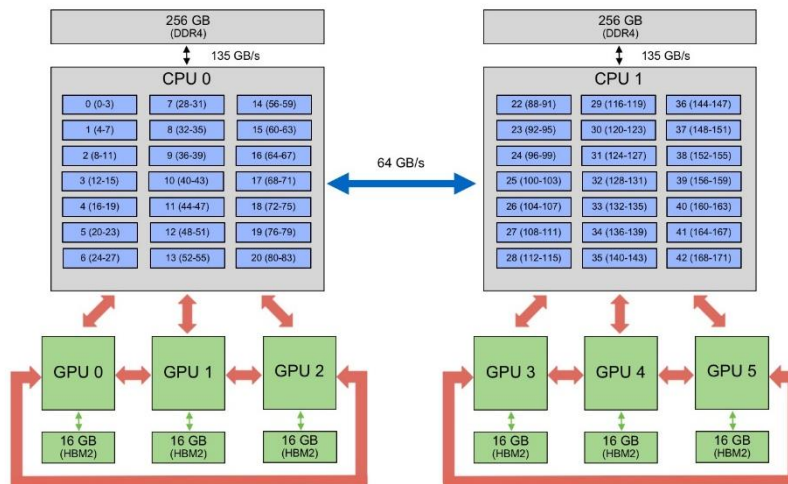


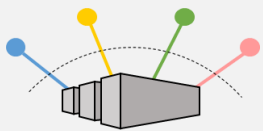
<https://eurohpc-ju.europa.eu/pictures>

Significant opportunity ...
Significant challenges

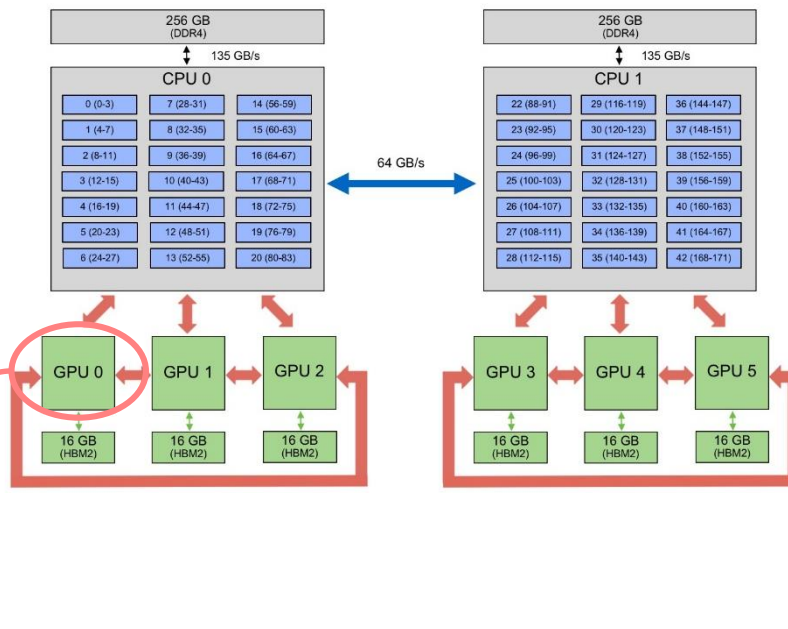


Exascale Hardware

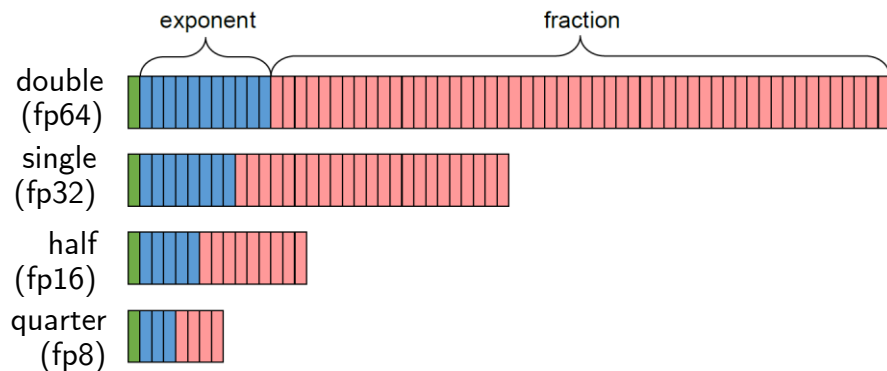




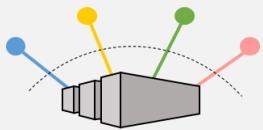
Exascale Hardware



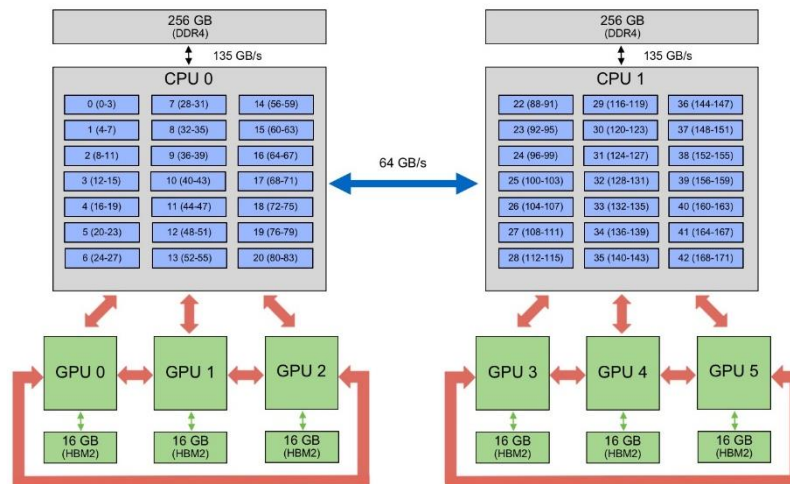
$$(-1)^{\text{sign}} \times 2^{(\text{exponent}-\text{offset})} \times 1.\text{fraction}$$



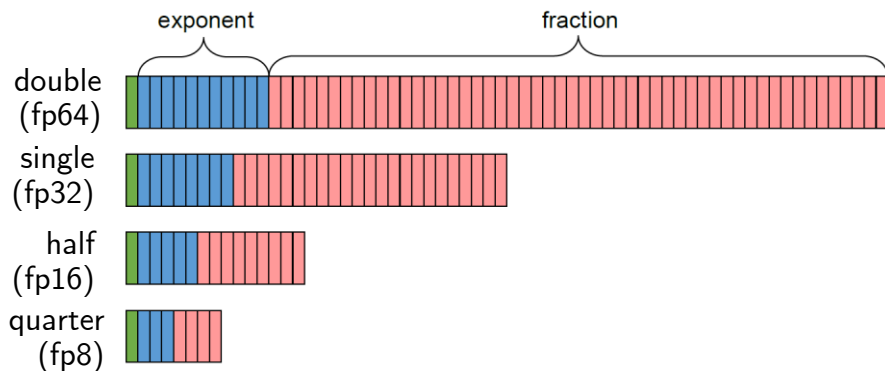
	size (bits)	range	u	perf. (NVIDIA H100)
fp64	64	$10^{\pm 308}$	1×10^{-16}	60 Tflops/s
fp32	32	$10^{\pm 38}$	6×10^{-8}	1 Pflop/s
fp16	16	$10^{\pm 5}$	5×10^{-4}	2 Pflops/s
bfloat16	16	$10^{\pm 38}$	4×10^{-3}	
fp8-e5m2	8	$10^{\pm 5}$	1×10^{-1}	4 Pflops/s
fp8-e4m3	8	$10^{\pm 2}$	6×10^{-2}	



Exascale Hardware



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



HPL-MxP

NUMBER 1 SYSTEM

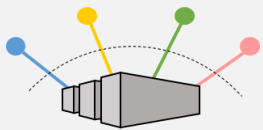
1

Frontier
Oak Ridge National Laboratory
US

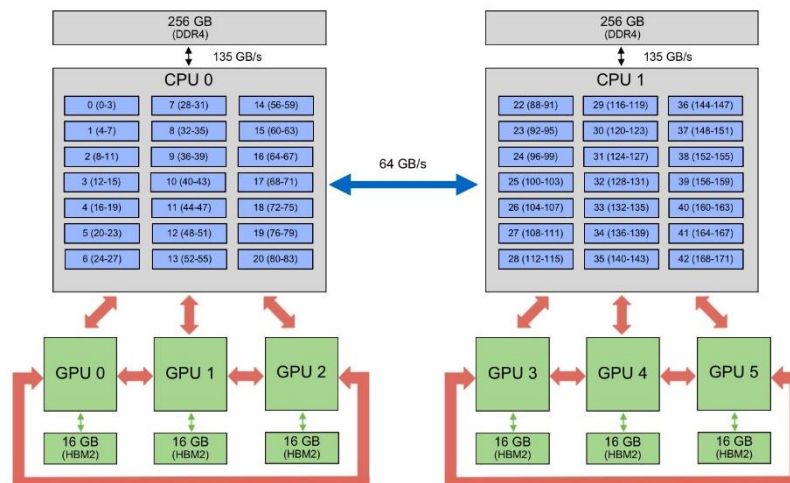
ACHIEVED
7.9 Eflops/s

Jack Dongarra
Piotr Luszczek

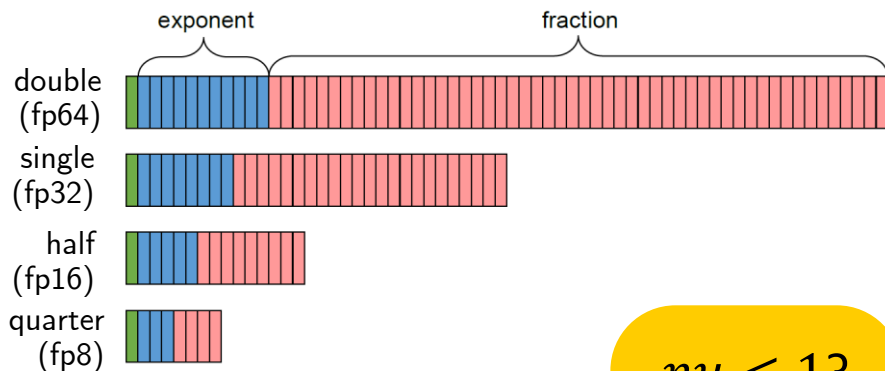
ICL INNOVATIVE



Exascale Hardware



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



$nu < 1?$

HPL-MxP

NUMBER 1 SYSTEM

1

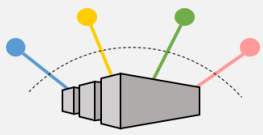
Frontier

Oak Ridge National Laboratory
US

ACHIEVED
7.9 Eflops/s

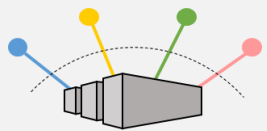
Jack Dongarra
Piotr Luszczek

ICL INNOVATIVE



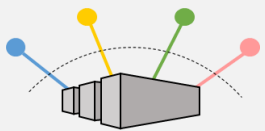
Mixed precision in NLA

- **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]



When Can I Use Low Precision?

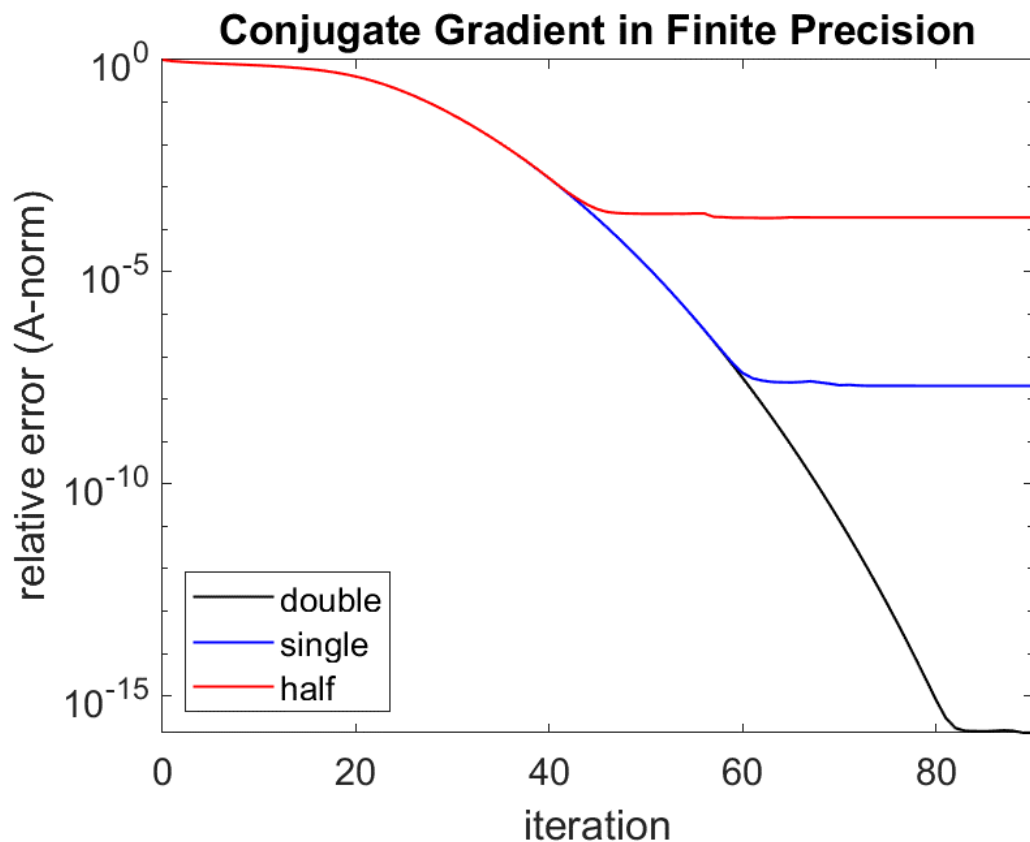
1. When low accuracy is needed

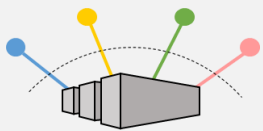


When Can I Use Low Precision?

1. When low accuracy is needed

```
A = diag(linspace(.001,1,100));  
b = ones(n,1);
```





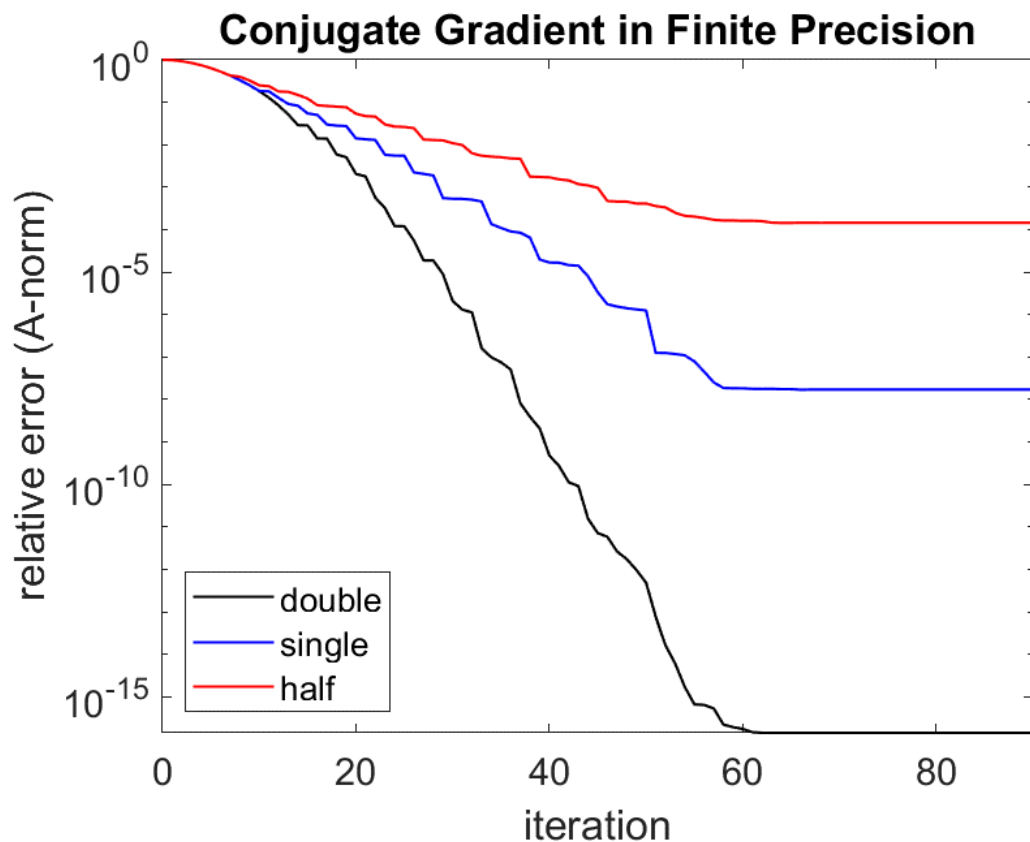
When Can I Use Low Precision?

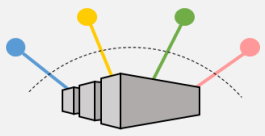
1. When low accuracy is needed

$$n = 100, \lambda_1 = 10^{-3}, \lambda_n = 1$$

$$\lambda_i = \lambda_1 + \left(\frac{i-1}{n-1}\right) (\lambda_n - \lambda_1) (0.65)^{n-i}, \quad i = 2, \dots, n-1$$

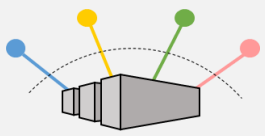
$$b = \text{ones}(n, 1);$$





When Can I Use Low Precision?

1. When low accuracy is needed
2. When a self-correction mechanism is available



When Can I Use Low Precision?

1. When low accuracy is needed
2. When a self-correction mechanism is available

Example: Iterative refinement

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

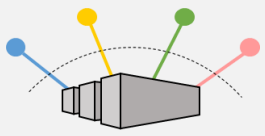
for $i = 0$: maxit

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

Solve $Ad_i = r_i$ (in precision u_s)

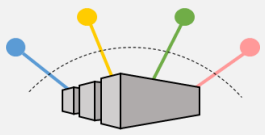
$x_{i+1} = x_i + d_i$ (in precision u)

e.g., [Langou et al., 2006], [Arioli and Duff, 2009], [Hogg and Scott, 2010], [Abdelfattah et al., 2016], [C. and Higham, 2018], [Amestoy et al., 2021]



When Can I Use Low Precision?

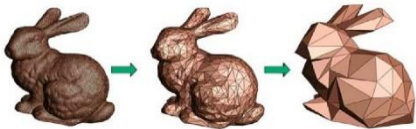
1. When low accuracy is needed
2. When a self-correction mechanism is available
3. When other approximations are being used



When Can I Use Low Precision?

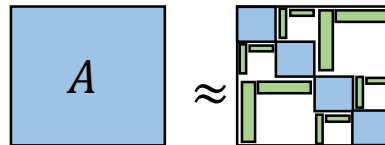
1. When low accuracy is needed
 2. When a self-correction mechanism is available
 3. When other approximations are being used
- E.g., reduced models, sparsification, low-rank approximations, randomization

Model Reduction



[Schilders, van der Vorst, Rommes, 2008]

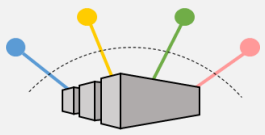
Low-rank approximation



Sparsification, randomization



[Sinha, 2018]

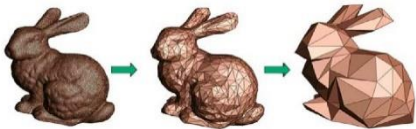


When Can I Use Low Precision?

1. When low accuracy is needed
2. When a self-correction mechanism is available
3. When other approximations are being used

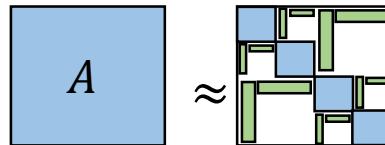
- E.g., reduced models, **sparsification**, **low-rank approximations**, **randomization**

Model Reduction



[Schilders, van der Vorst, Rommes, 2008]

Low-rank approximation

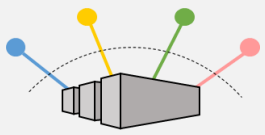


Sparsification, randomization



[Sinha, 2018]

Mixed Precision Sparse Approximate Inverse Preconditioners



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 ϵ

For each column k :

 Compute QR factorization of submatrix of A defined by J

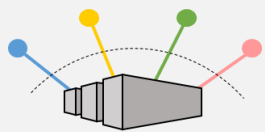
 Use QR factorization to solve $\min_{m_k} \|e_k - Am_k\|_2$

 If $\|r_k\|_2 = \|e_k - Am_k\|_2 \leq \epsilon$

 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 ϵ

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 $\|r_k\|_2 = \|e_k - Am_k\|_2 \leq \epsilon$

 break;

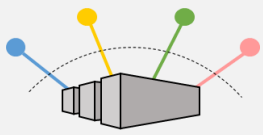
 Else

 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, Tuma, 1999], [He, Yin, Gao, 2020]

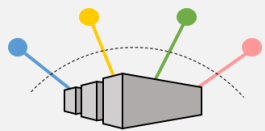


SPAI Preconditioners in Low Precision

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

Notes and assumptions:

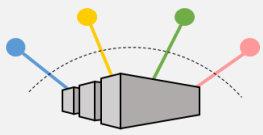
- We will assume that the SPAI construction is performed in some precision u_f
- We will denote quantities computed in finite precision with hats
- In our application, we want a left preconditioner, so we will run the algorithm on A^T and set $M \leftarrow M^T$.
- We will assume that the QR factorization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



SPAI Preconditioners in Low Precision

Two interesting questions:

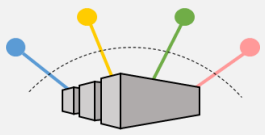
1. Assuming we impose no maximum sparsity pattern on \widehat{M} , under what constraint on \mathbf{u}_f can we guarantee that $\|\hat{r}_k\|_2 \leq \epsilon$, with $\hat{r}_k = fl_{\mathbf{u}_f}(e_k - A^T \widehat{m}_k^T)$ for the computed \widehat{m}_k^T ?



SPAI Preconditioners in Low Precision

Two interesting questions:

1. Assuming we impose no maximum sparsity pattern on \widehat{M} , under what constraint on u_f can we guarantee that $\|\hat{r}_k\|_2 \leq \epsilon$, with $\hat{r}_k = fl_{u_f}(e_k - A^T \widehat{m}_k^T)$ for the computed \widehat{m}_k^T ?
2. Assume that when M is computed in exact arithmetic, we quit as soon as $\|r_k\| \leq \epsilon$. For \widehat{M} computed in precision u_f with the same sparsity pattern as M , what is $\|e_k - A^T \widehat{m}_k^T\|_2$?



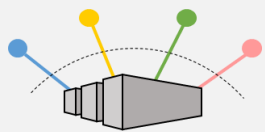
SPAI Preconditioning in Low Precision

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

$$\|\hat{r}_k\|_2 \leq n^3 \mathbf{u}_f \left(\|e_k\| + \|A^T\| \|\hat{m}_k^T\| \right).$$

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

$$n^3 \mathbf{u}_f \left(\|e_k\| + \|A^T\| \|\hat{m}_k^T\| \right) \leq \epsilon.$$



SPAI Preconditioning in Low Precision

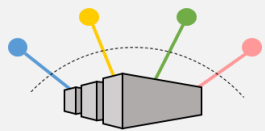
Using standard rounding error analysis and perturbation results for LS problems, we have

$$\|\hat{r}_k\|_2 \leq n^3 \mathbf{u}_f \left(\|e_k\| + |A^T| \|\hat{m}_k^T\| \right)_2.$$

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

$$n^3 \mathbf{u}_f \left(\|e_k\| + |A^T| \|\hat{m}_k^T\| \right)_2 \leq \epsilon.$$

→ problem must not be so ill-conditioned WRT \mathbf{u}_f that we incur an error greater than ϵ just computing the residual

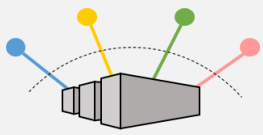


SPAI Preconditioning in Low Precision

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

$$\text{cond}_2(A^T) \lesssim \epsilon u_f^{-1},$$

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



SPAI Preconditioning in Low Precision

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

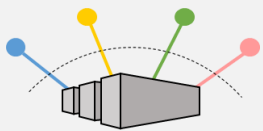
$$\text{cond}_2(A^T) \lesssim \varepsilon u_f^{-1},$$

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

Another view: with a given matrix A and a given precision u_f , one must set ε such that

$$\varepsilon \geq u_f \text{cond}_2(A^T).$$

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:

$$\text{cond}_2(A^T) \lesssim \varepsilon \mathbf{u}_f^{-1},$$

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

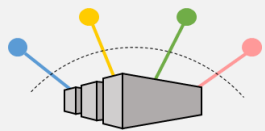
Another view: with a given matrix A and a given precision \mathbf{u}_f , one must set ε such that

$$\varepsilon \geq \mathbf{u}_f \text{cond}_2(A^T).$$

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

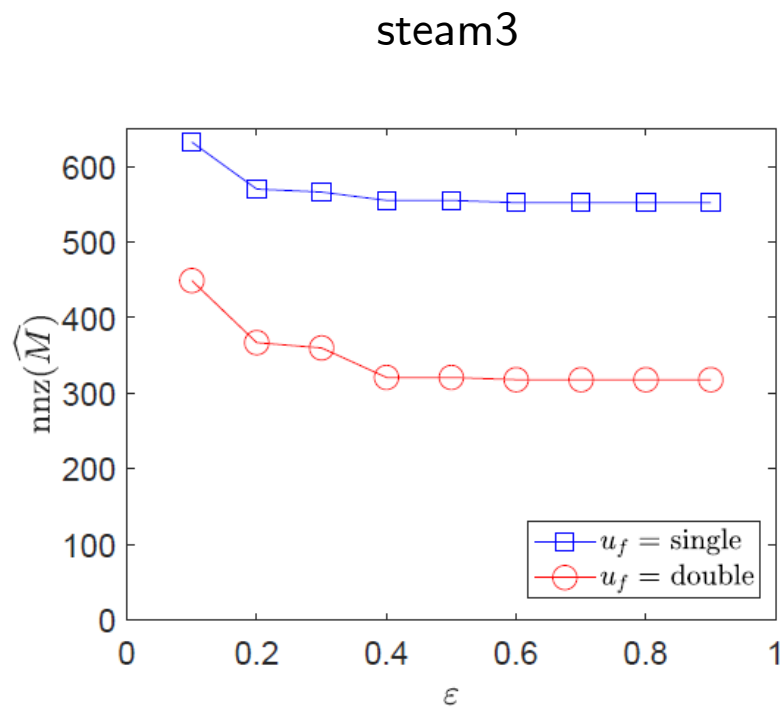
Resulting bounds for \hat{M} :

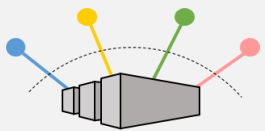
$$\| \|I - A^T \hat{M}^T \| \|_F \leq 2\sqrt{n}\varepsilon, \quad \| \|I - \hat{M}A \| \|_\infty \leq 2n\varepsilon$$



Size of SPAI Preconditioner in Low Precision

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

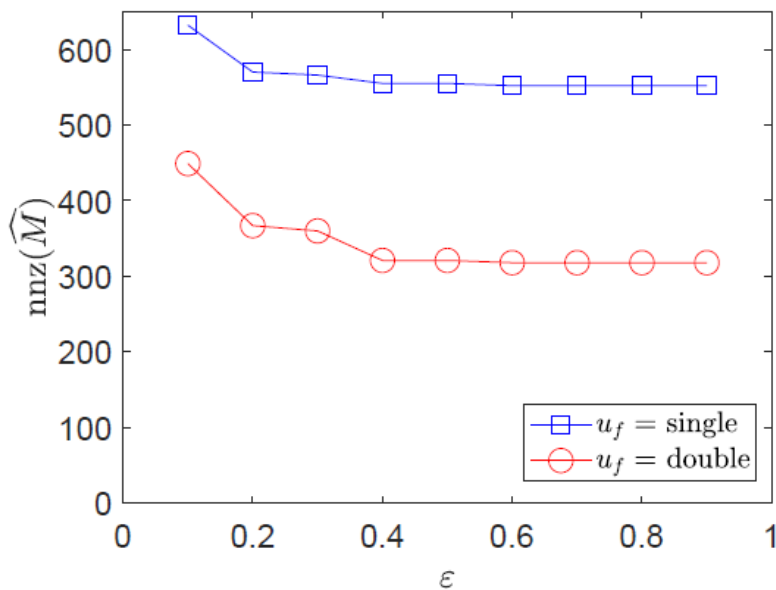




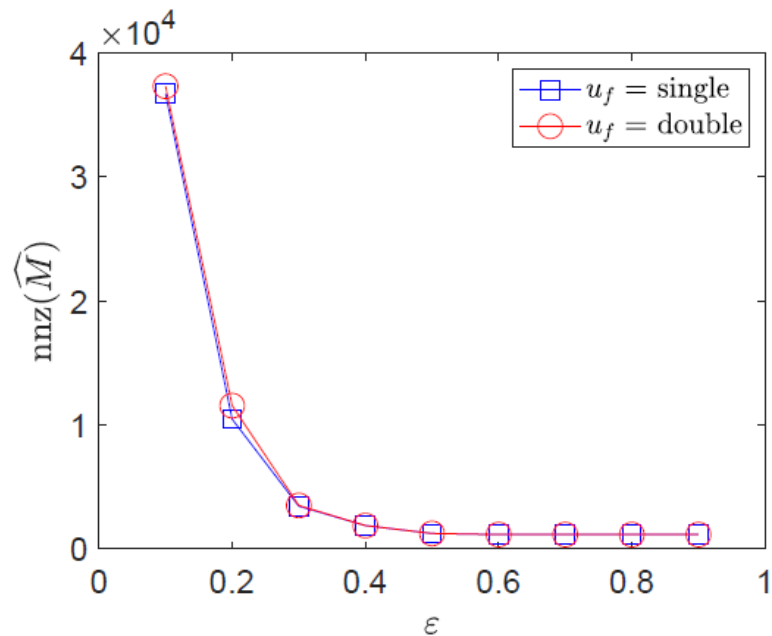
Size of SPAI Preconditioner in Low Precision

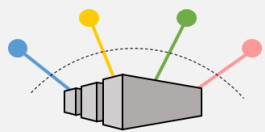
How does precision used affect the number of nonzeros in \widehat{M} ?

steam3



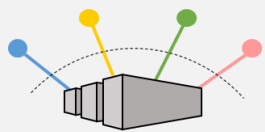
saylr1





Second Question

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



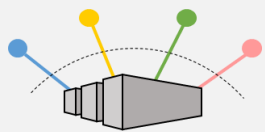
Second Question

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

In this case, we obtain the bound

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

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



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 u_f)

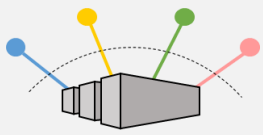
for $i = 0$: maxit

$$r_i = b - Ax_i \quad (\text{in precision } u_r)$$

$$\text{Solve } Ad_i = r_i \quad (\text{in precision } u_s)$$

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

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



GMRES-Based Iterative Refinement

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

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

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

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

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

Solve $Ax_0 = b$ by LU factorization

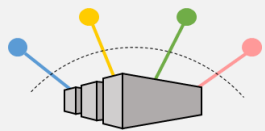
for $i = 0$: maxit

$$r_i = b - Ax_i$$

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

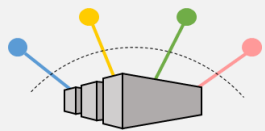
$$x_{i+1} = x_i + d_i$$

$$u_s = u$$



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-GMRES-IR

SPAI-GMRES-IR

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

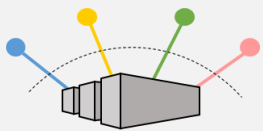
Solve $\widehat{M}Ax_0 = \widehat{M}b$

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

$$r_i = b - Ax_i$$

Solve $Ad_i = r_i$ via GMRES on $\widehat{M}Ad_i = \widehat{M}r_i$

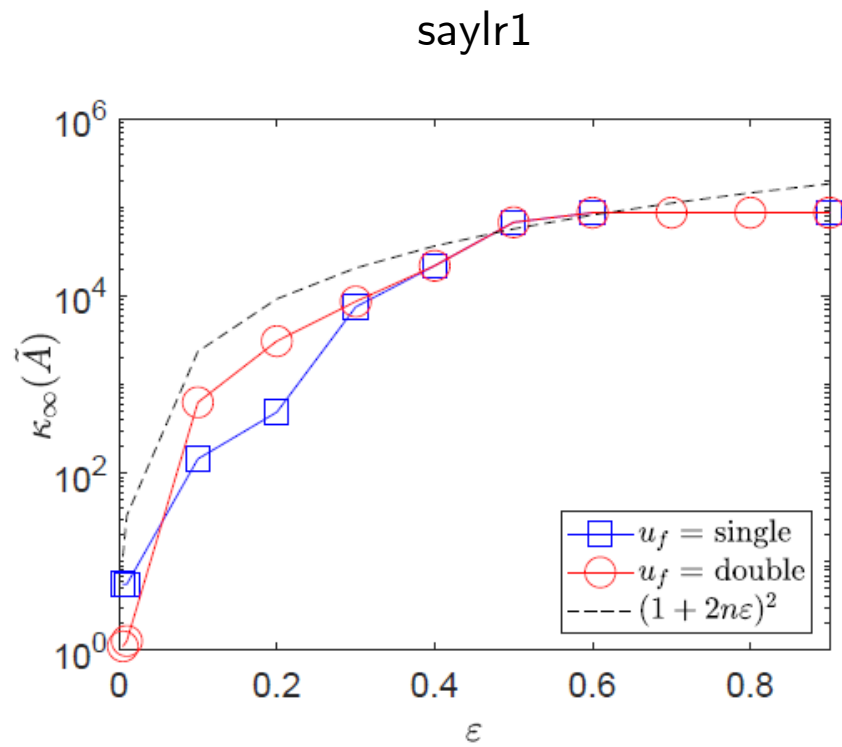
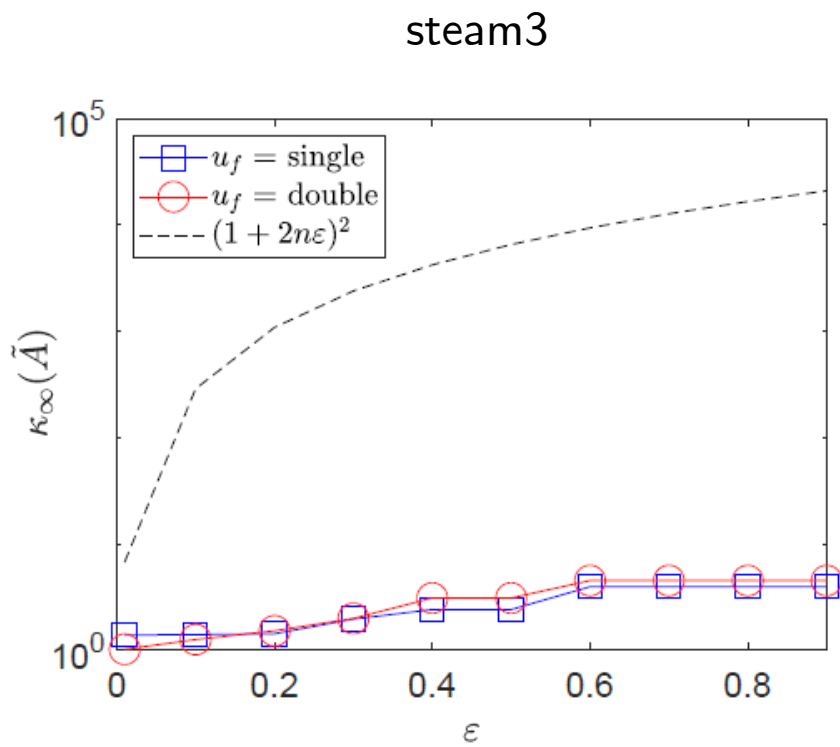
$$x_{i+1} = x_i + d_i$$

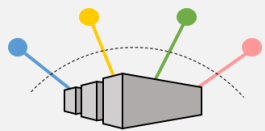


Low Precision SPAI within GMRES-IR

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

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

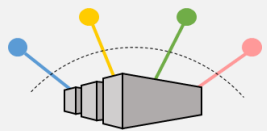




Low Precision SPAI within GMRES-IR

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

$$n\mathbf{u}_f \text{cond}_2(A^T) \lesssim n\boldsymbol{\varepsilon} \lesssim \mathbf{u}^{-1/2}.$$

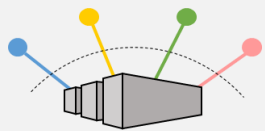


Low Precision SPAI within GMRES-IR

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

$$n u_f \text{cond}_2(A^T) \lesssim n \epsilon \lesssim u^{-1/2}.$$

\hat{M} can be
constructed



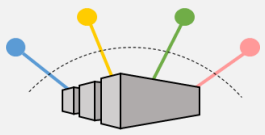
Low Precision SPAI within GMRES-IR

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

$$n u_f \text{cond}_2(A^T) \lesssim n \epsilon \lesssim u^{-1/2}.$$

\hat{M} can be
constructed

\hat{M} is a good enough
preconditioner



Low Precision SPAI within GMRES-IR

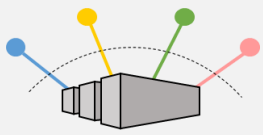
To guarantee that both SPAI construction will complete and the GMRES-based iterative refinement scheme will converge, we must have roughly

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

\hat{M} can be
constructed

\hat{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

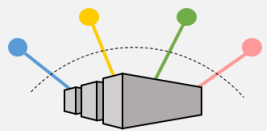
$$n u_f \text{cond}_2(A^T) \lesssim n \varepsilon \lesssim u^{-1/2}.$$

\hat{M} can be
constructed

\hat{M} is a good enough
preconditioner

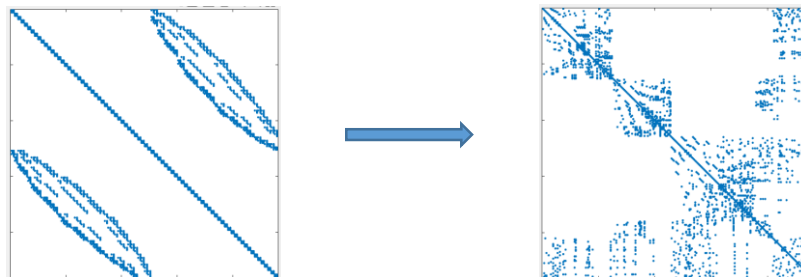
If ε satisfies these constraints, then the **constraints on condition number** for forward and backward errors to converge are the **same as for GMRES-IR with full LU factorization**.

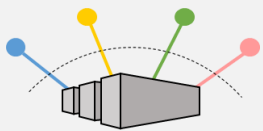
Compared to GMRES-IR with full LU factorization, in general expect **slower convergence, but much sparser preconditioner**.



SPAI-GMRES-IR Example

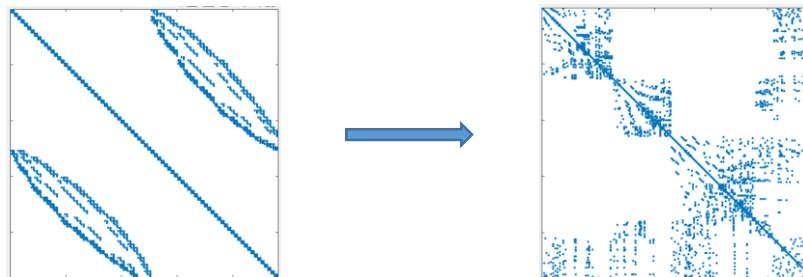
Matrix: steam1, $n = 240$, $\text{nnz} = 2,248$, $\kappa_{\infty}(A) = 3 \cdot 10^7$, $\text{cond}(A^T) = 3 \cdot 10^3$



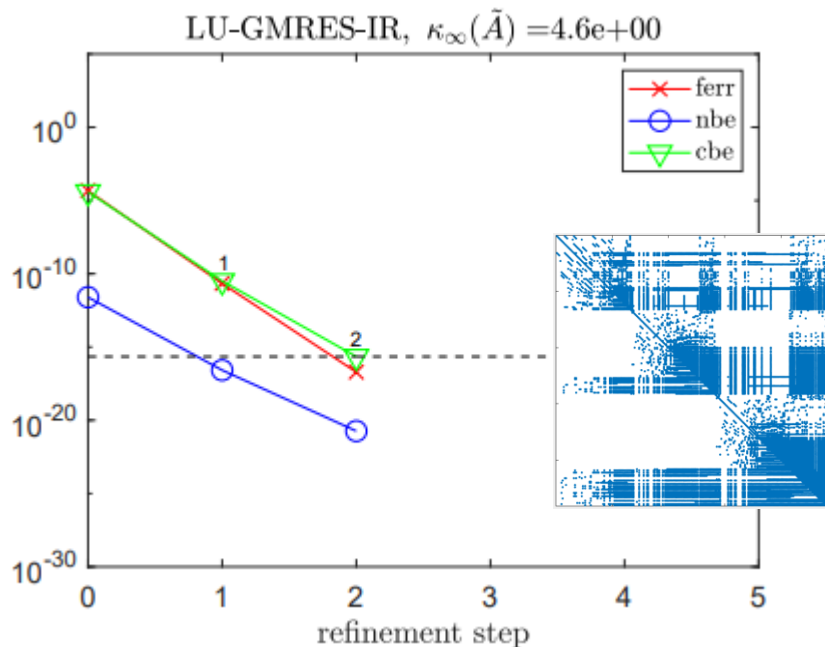


SPAI-GMRES-IR Example

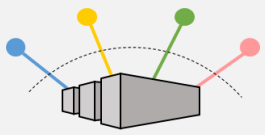
Matrix: steam1, $n = 240$, $\text{nnz} = 2,248$, $\kappa_\infty(A) = 3 \cdot 10^7$, $\text{cond}(A^T) = 3 \cdot 10^3$



$(\mathbf{u}_f, \mathbf{u}, \mathbf{u}_r) = (\text{single}, \text{double}, \text{quad})$

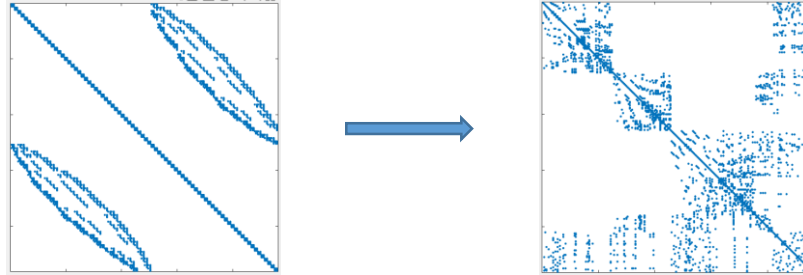


$\text{nnz}(L + U) = 13,765$

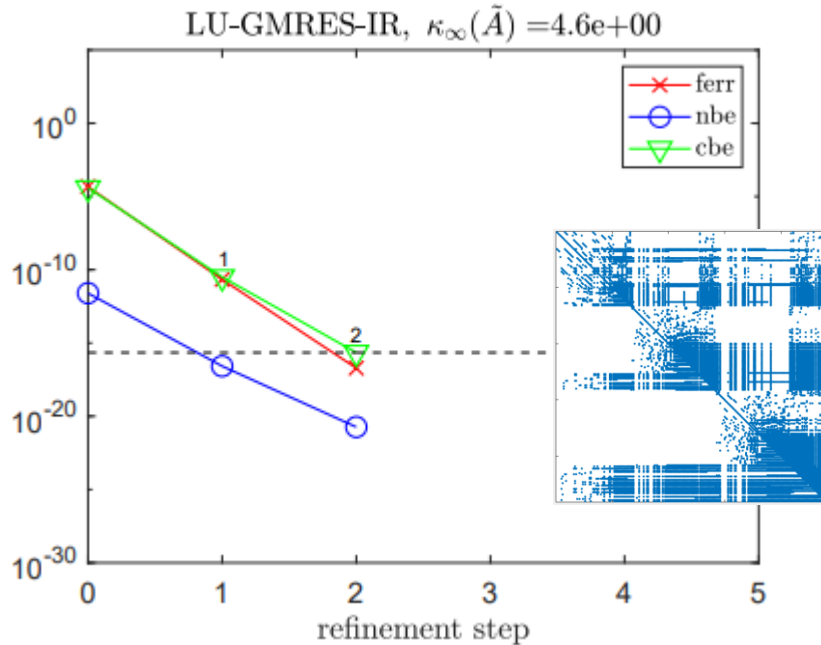


SPAI-GMRES-IR Example

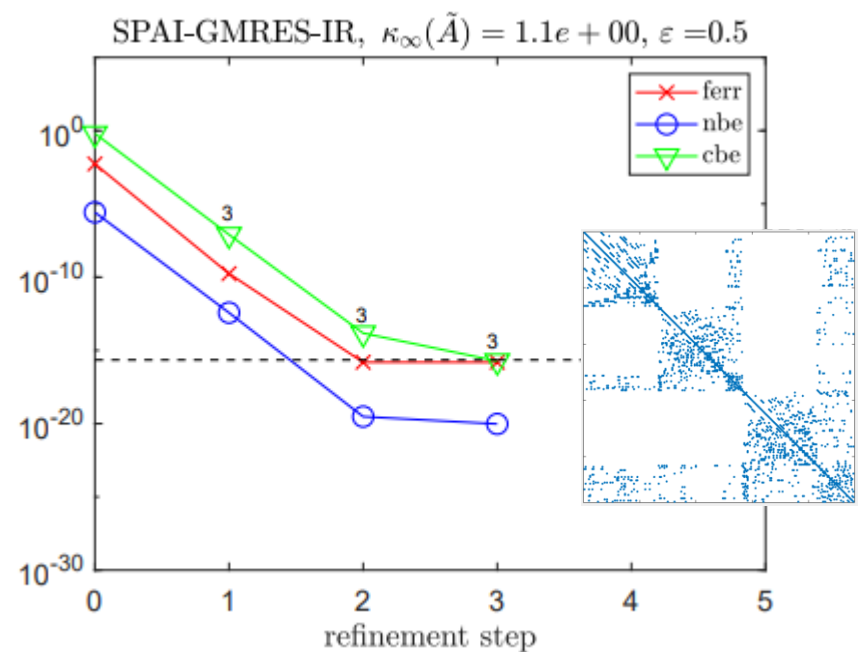
Matrix: steam1, $n = 240$, $\text{nnz} = 2,248$, $\kappa_\infty(A) = 3 \cdot 10^7$, $\text{cond}(A^T) = 3 \cdot 10^3$



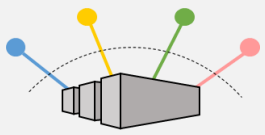
$(\mathbf{u}_f, \mathbf{u}, \mathbf{u}_r) = (\text{single}, \text{double}, \text{quad})$



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



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



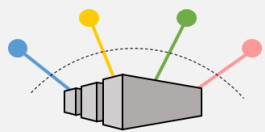
A Question

Is there a point in using precision higher than that dictated by $\mathbf{u}_f \text{cond}_2(A^T) \leq \epsilon$?

Matrix: bfw782, $n = 782$, $\text{nnz} = 7514$, $\kappa_\infty(A) = 7 \cdot 10^3$, $\text{cond}(A^T) = 1 \cdot 10^3$

$(\mathbf{u}_f, \mathbf{u}, \mathbf{u}_r) = (\text{half}, \text{single}, \text{double})$

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



A Question

Is there a point in using precision higher than that dictated by $\mathbf{u}_f \text{cond}_2(A^T) \leq \epsilon$?

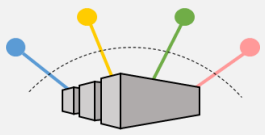
Matrix: bfw782, $n = 782$, $\text{nnz} = 7514$, $\kappa_\infty(A) = 7 \cdot 10^3$, $\text{cond}(A^T) = 1 \cdot 10^3$

$$(\mathbf{u}_f, \mathbf{u}, \mathbf{u}_r) = (\text{half}, \text{single}, \text{double})$$

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

$$(\mathbf{u}_f, \mathbf{u}, \mathbf{u}_r) = (\text{single}, \text{single}, \text{double})$$

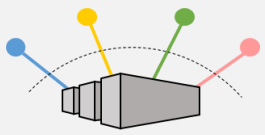
Preconditioner	$\kappa_\infty(\tilde{A})$	Precond. nnz	GMRES-IR steps/iteration
SPAI ($\epsilon = 0.2$)	$2.2e + 02$	26801	69 (32, 37)
SPAI ($\epsilon = 0.5$)	$9.7e + 02$	7529	153 (71, 82)



Ongoing and Future Work

- Incorporate mixed-precision storage of \hat{M} and **adaptive-precision SpMV** to apply \hat{M} using the work of [Graillat et al., 2002]
- Theoretical analysis of **incomplete factorization preconditioners** in mixed precision
 - Experimental work shows that half precision works well in practice [Scott, Tuma, 2023]

Mixed Precision Randomized Preconditioners



Our setting

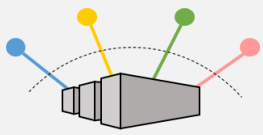
Let $A \in \mathbb{R}^{n \times n}$ be a symmetric positive semidefinite matrix. Want to solve

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

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.

Many applications, e.g., ridge regression.



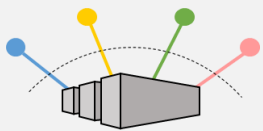
Limited Memory Preconditioners

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

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

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.

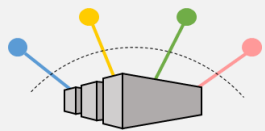
Nyström approximation:

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

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

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.



Randomized Nyström Approximation

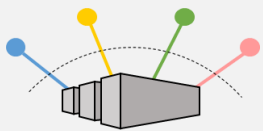
[Tropp et al., 2017]

Given sym. PSD matrix A , target rank k

$$G = \text{randn}(n, k)$$

$$[Q, \sim] = \text{qr}(G, 0)$$





Randomized Nyström Approximation

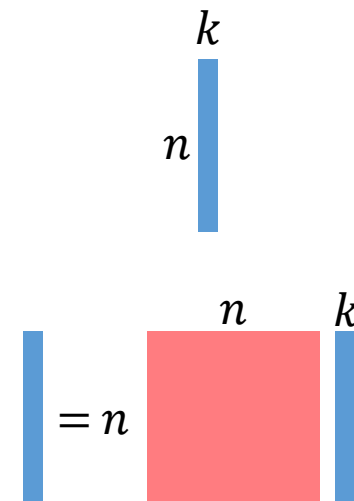
[Tropp et al., 2017]

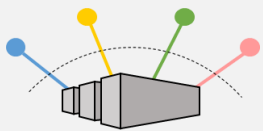
Given sym. PSD matrix A , target rank k

$$G = \text{randn}(n, k)$$

$$[Q, \sim] = \text{qr}(G, 0)$$

$$Y = AQ$$





Randomized Nyström Approximation

[Tropp et al., 2017]

Given sym. PSD matrix A , target rank k

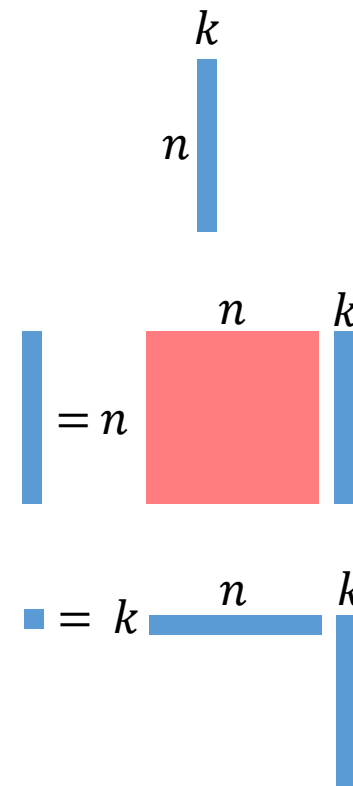
$$G = \text{randn}(n, k)$$

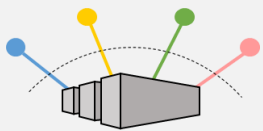
$$[Q, \sim] = \text{qr}(G, 0)$$

$$Y = AQ$$

Compute shift ν ; $Y_\nu = Y + \nu Q$

$$B = Q^T Y_\nu$$





Randomized Nyström Approximation

[Tropp et al., 2017]

Given sym. PSD matrix A , target rank k

$$G = \text{randn}(n, k)$$

$$[Q, \sim] = \text{qr}(G, 0)$$

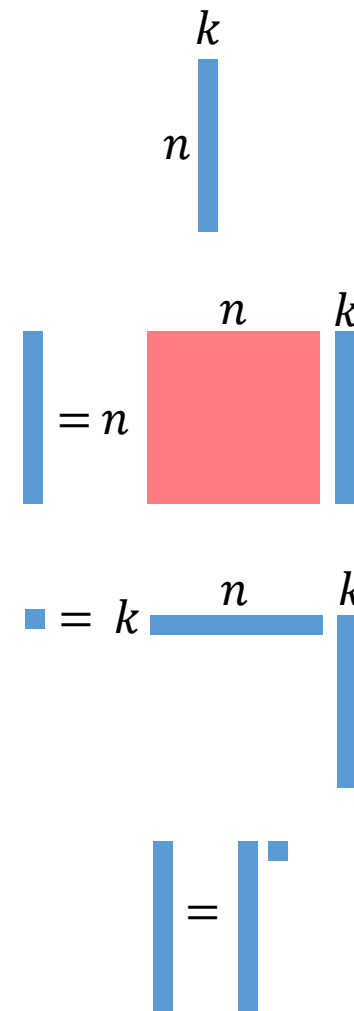
$$Y = AQ$$

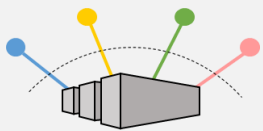
Compute shift ν ; $Y_\nu = Y + \nu Q$

$$B = Q^T Y_\nu$$

$$C = \text{chol}((B + B^T)/2)$$

Solve $F = Y_\nu / C$





Randomized Nyström Approximation

[Tropp et al., 2017]

Given sym. PSD matrix A , target rank k

$$G = \text{randn}(n, k)$$

$$[Q, \sim] = \text{qr}(G, 0)$$

$$Y = AQ$$

Compute shift ν ; $Y_\nu = Y + \nu Q$

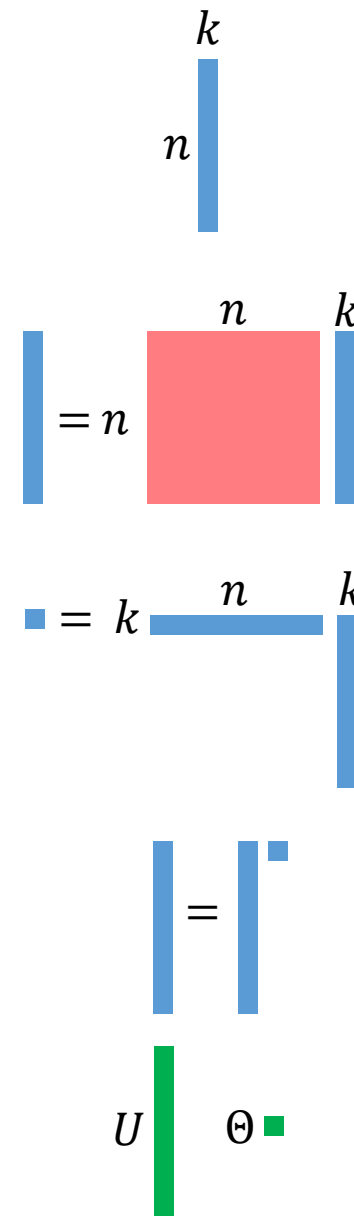
$$B = Q^T Y_\nu$$

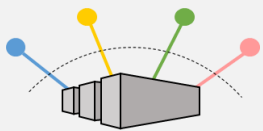
$$C = \text{chol}((B + B^T)/2)$$

Solve $F = Y_\nu / C$

$$[U, \Sigma, \sim] = \text{svd}(F, 0)$$

$$\Theta = \max(0, \Sigma^2 - \nu I)$$





Randomized Nyström Approximation

[Tropp et al., 2017]

Given sym. PSD matrix A , target rank k

$$G = \text{randn}(n, k)$$

$$[Q, \sim] = \text{qr}(G, 0)$$

$$Y = AQ$$

Compute shift ν ; $Y_\nu = Y + \nu Q$

$$B = Q^T Y_\nu$$

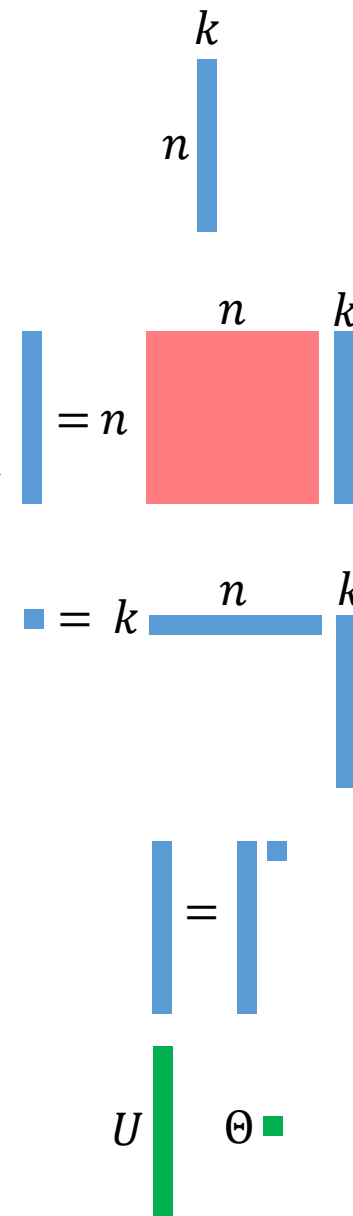
$$C = \text{chol}((B + B^T)/2)$$

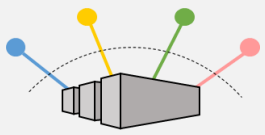
Solve $F = Y_\nu / C$

$$[U, \Sigma, \sim] = \text{svd}(F, 0)$$

$$\Theta = \max(0, \Sigma^2 - \nu I)$$

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



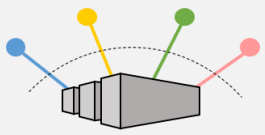


Error Bounds

$$\|A - \hat{A}_N\|_2 = \|A - A_N + A_N - \hat{A}_N\|_2 \leq \|A - A_N\|_2 + \|A_N - \hat{A}_N\|_2$$

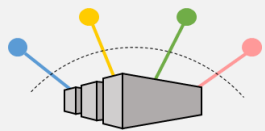
exact Nyström
approximation

Nyström approximation
computed in
finite precision



Error Bounds

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



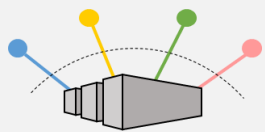
Error Bounds

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

Deterministic bound [Gittens, Mahoney, 2016]:

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

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



Error Bounds

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

Deterministic bound [Gittens, Mahoney, 2016]:

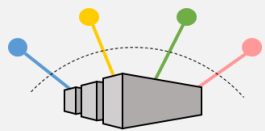
$$\|A - A_N\|_2 \leq \lambda_{k+1} + \left\| \Sigma_2^{1/2} U_2^T Q (U_1 Q)^+ \right\|_2^2$$

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

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

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

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

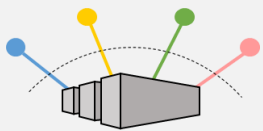


Finite Precision Error Bound

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: $A_N - \hat{A}_N$

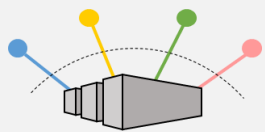
Assumptions:

- 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]: With failure probability at most $e^{-t^2/2} + c_1\alpha$,

$$\|A_N - \hat{A}_N\|_2 \lesssim \alpha^{-1} n^{1/2} k (n^{1/2} + k^{1/2} + t)^2 u_p \|A\|_2 \kappa(A_k)$$

where A_k is the best rank- k approximation 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 $u \ll u_p$

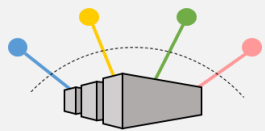
[C., Daužickaitė, 2022]: With failure probability at most $e^{-t^2/2} + c_1\alpha$,

$$\|A_N - \hat{A}_N\|_2 \lesssim \alpha^{-1} n^{1/2} k (n^{1/2} + k^{1/2} + t)^2 u_p \|A\|_2 \kappa(A_k)$$

where A_k is the best rank- k approximation of A

Interpretation: Likely that $\|A_N - \hat{A}_N\|_2 \gtrsim \|A - A_N\|_2$ when

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



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]: With failure probability at most $e^{-t^2/2} + c_1\alpha$,

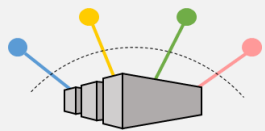
$$\|A_N - \hat{A}_N\|_2 \lesssim \alpha^{-1} n^{1/2} k (n^{1/2} + k^{1/2} + t)^2 u_p \|A\|_2 \kappa(A_k)$$

where A_k is the best rank- k approximation of A

Interpretation: Likely that $\|A_N - \hat{A}_N\|_2 \gtrsim \|A - A_N\|_2$ when

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

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



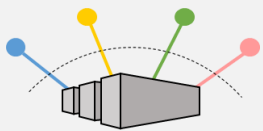
Condition Number Bounds

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

Let

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

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



Condition Number Bounds

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

Let

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

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

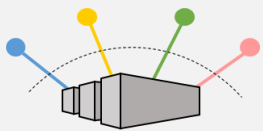
Then

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

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

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

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



Condition Number Bounds

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

Let

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

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

Then

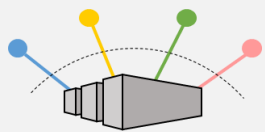
If $\mathcal{E} = 0$, reduces to bounds of [Frangella, Tropp, Udell, 2021] for exact case.

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

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

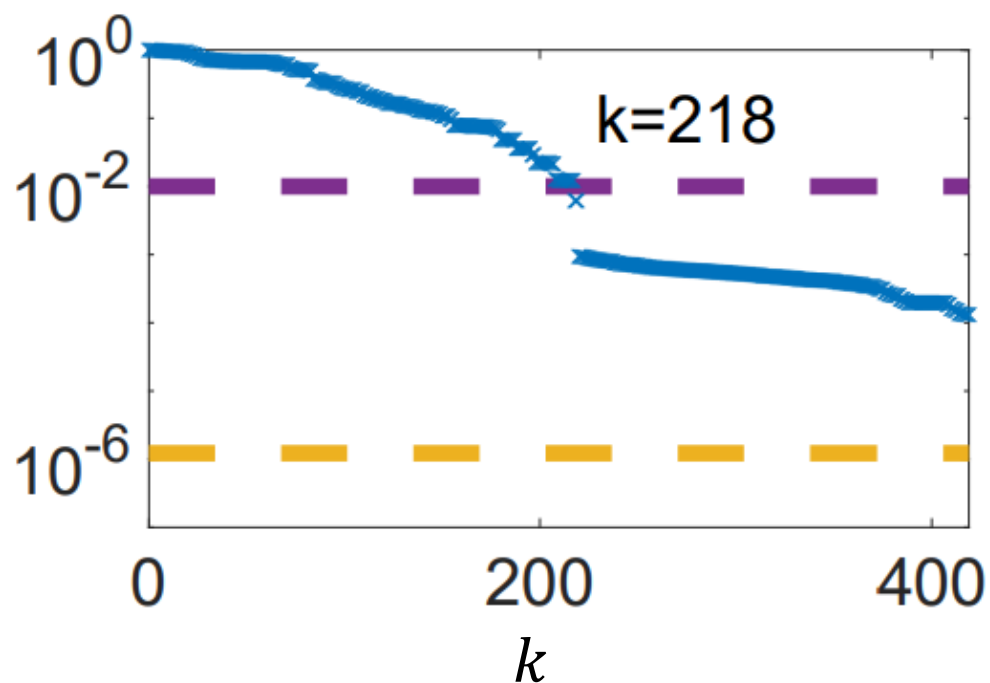
Regardless of this constraint, if A is positive definite, then




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

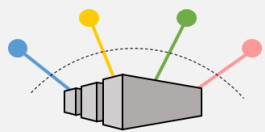


Numerical Experiment

Matrix: bcsstm07, $n = 420$

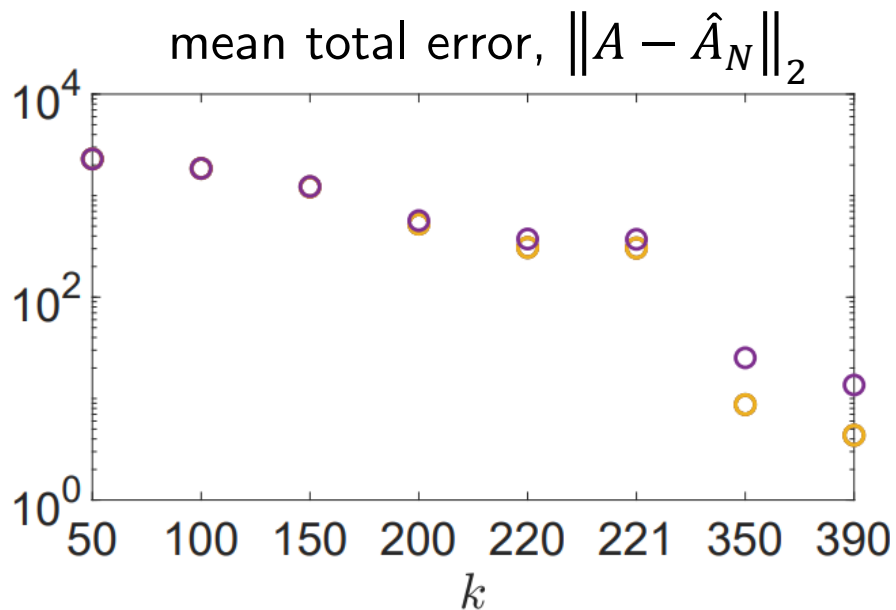


-  λ_{k+1}/λ_1
-  $\sqrt{n}u_p, u_p = \text{half}$
-  $\sqrt{n}u_p, u_p = \text{single}$

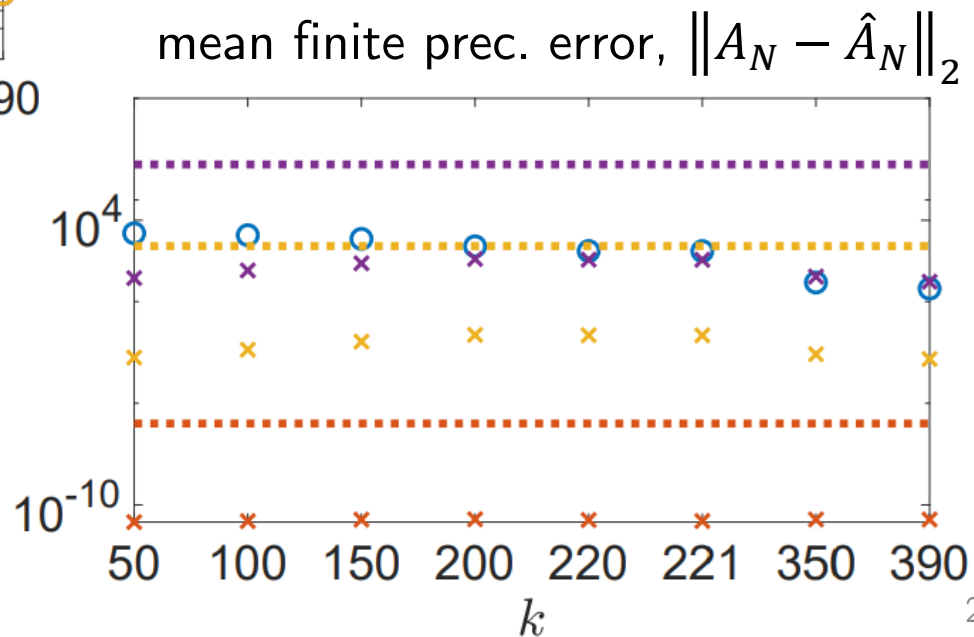


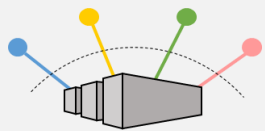
Numerical Experiment

Matrix: bcsstm07, $n = 420$



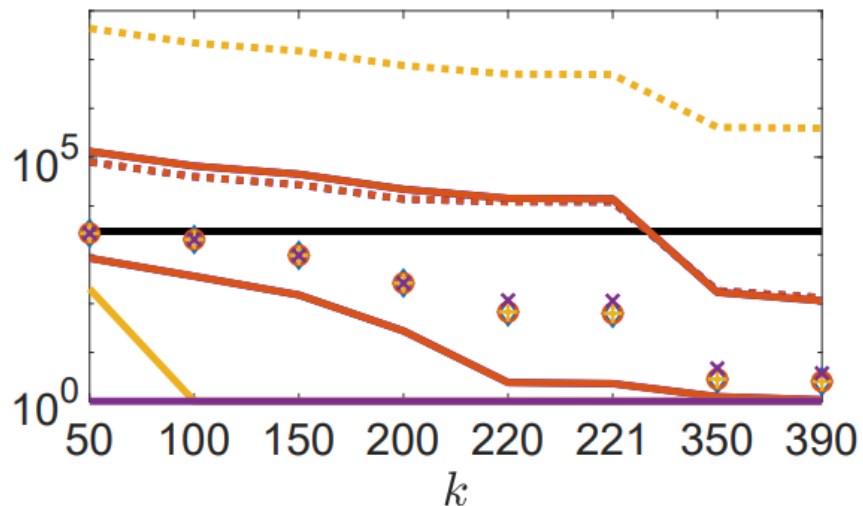
- exact
- mixed, $u_p = \text{half}$
- mixed, $u_p = \text{single}$
- mixed, $u_p = \text{double}$





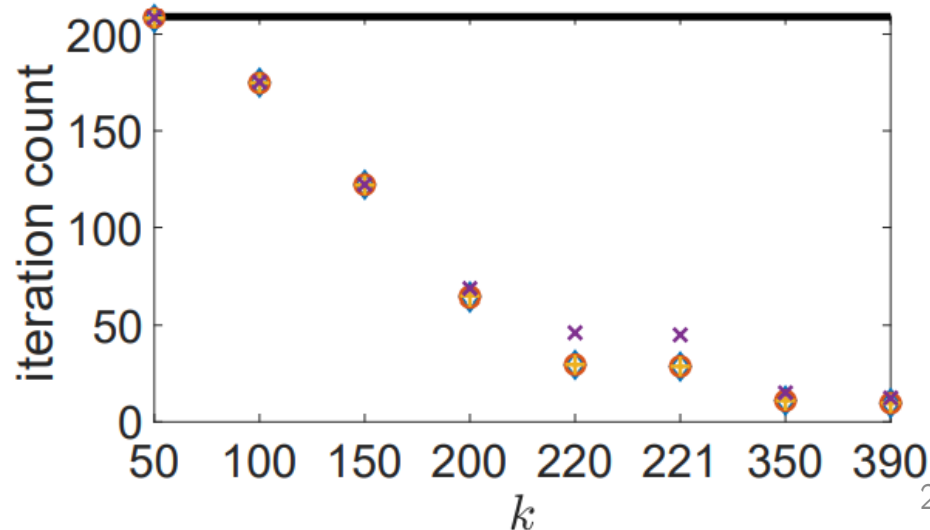
Numerical Experiment

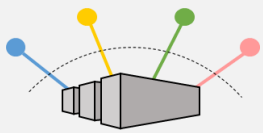
$$\kappa(\hat{P}^{-1/2}(A + \mu I)\hat{P}^{-1/2})$$



- unpreconditioned
- exact
- mixed, $u_p = \text{half}$
- mixed, $u_p = \text{single}$
- mixed, $u_p = \text{double}$

PCG iteration count





Ongoing Work

- Mixed-precision randomized preconditioners for Krylov subspace method-based iterative refinement of least squares problems $\min_x \|b - Ax\|_2$

Compute \hat{R} factor of QR decomposition of randomly sketched A using precision u_s (sketching step) and u_o (QR step).

Solve $\min_x \|b - Ax\|_2$ via LSQR preconditioned with \hat{R} in precision u to get initial solution x_0 and residual r_0 .

for $i = 0, \dots$, until convergence

Compute residual $\begin{bmatrix} f_i \\ g_i \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r_i \\ x_i \end{bmatrix}$ and $h_i = \hat{R}^{-T} g_i$ in precision u_r .

Solve via KSM in (effective) precision u_s :

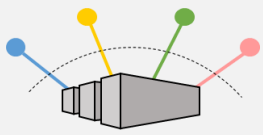
$$\begin{bmatrix} I & 0 \\ 0 & \hat{R}^{-T} \end{bmatrix} \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \hat{R}^{-1} \end{bmatrix} \begin{bmatrix} \delta r_i \\ \delta z_i \end{bmatrix} = \begin{bmatrix} f_i \\ h_i \end{bmatrix},$$

where $\hat{R} \delta x_i = \delta z_i$.

Update in precision u :

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

- Collaboration with Hartwig Anzt and Vasileios Georgiou



Ongoing Work

- Mixed-precision randomized preconditioners for Krylov subspace method-based iterative refinement of least squares problems $\min_x \|b - Ax\|_2$

Compute \hat{R} factor of QR decomposition of randomly sketched A using precision u_s (sketching step) and u_o (QR step).

Solve $\min_x \|b - Ax\|_2$ via LSQR preconditioned with \hat{R} in precision u to get initial solution x_0 and residual r_0 .

for $i = 0, \dots$, until convergence

Compute residual $\begin{bmatrix} f_i \\ g_i \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r_i \\ x_i \end{bmatrix}$ and $h_i = \hat{R}^{-T} g_i$ in precision u_r .

Solve via KSM in (effective) precision u_s :

$$\begin{bmatrix} I & 0 \\ 0 & \hat{R}^{-T} \end{bmatrix} \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \hat{R}^{-1} \end{bmatrix} \begin{bmatrix} \delta r_i \\ \delta z_i \end{bmatrix} = \begin{bmatrix} f_i \\ h_i \end{bmatrix},$$

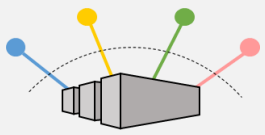
where $\hat{R} \delta x_i = \delta z_i$.

Update in precision u :

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

can use sketch-and-apply approach of [Meier et al., 2023]

- Collaboration with Hartwig Anzt and Vasileios Georgiou



Summary and Takeaway

- To efficiently use modern exascale machines, we **need to use mixed precision hardware**
- **Understanding the interaction and balance of errors** from finite precision and sources of algorithmic approximation is thus crucial
- Careful analysis will reveal **not only limitations, but opportunities!**

Thank You!

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