Mixed Precision Randomized Preconditioners

Erin C. Carson

Charles University

SIAM CSE 2023, Amsterdam, NL

March 2, 2023

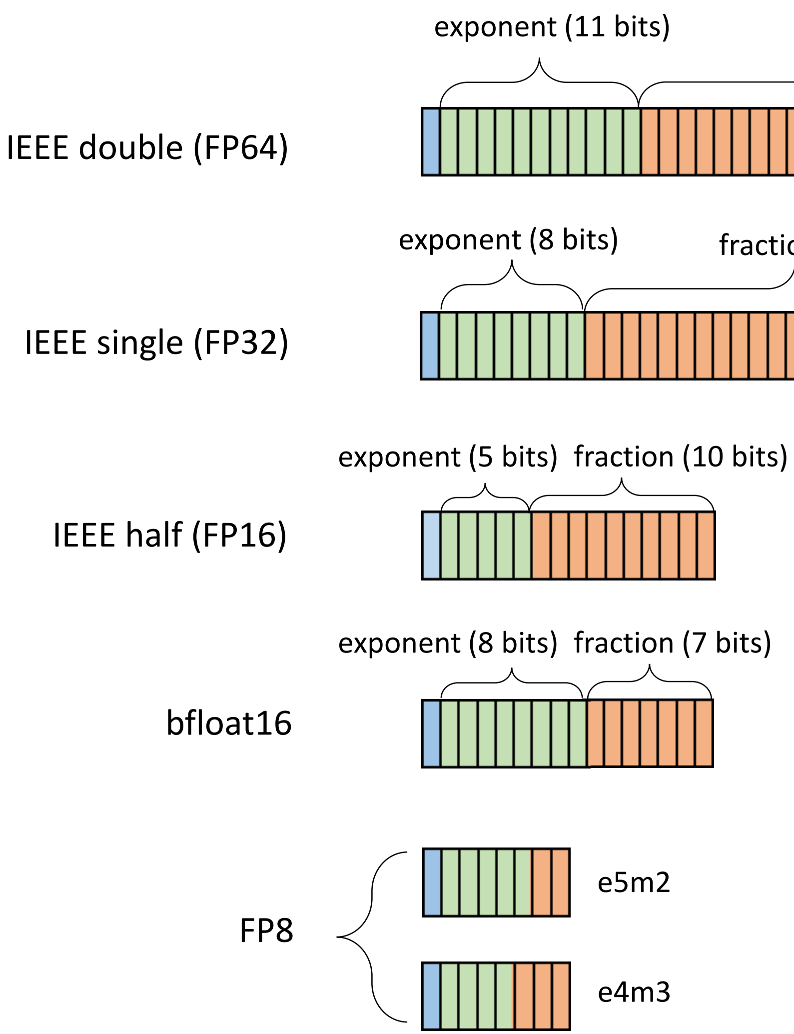
FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

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.

Floating Point Formats

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



	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}	

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], [Higham and Mary, 2022]

HPL-MxP Benchmark

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

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

HPL-MxP Benchmark

November 2022

Rank	Site	Computer	Cores	HPL-AI (Eflop/s)	TOP500 Rank	HPL Rmax (Eflop/s)	Speedup
1	DOE/SC/ORNL	Frontier	8,730,112	7.942	1	1.1020	7.2
2	EuroHPC/CSC	LUMI	2,174,976	2.168	3	0.3091	7.0
3	RIKEN	Fugaku	7,630,848	2.000	1	0.4420	4.5
4	EuroHPC/CINECA	Leonardo	1,463,616	1.842	4	0.1682	11.0
5	DOE/SC/ORNL	Summit	2,414,592	1.411	2	0.1486	9.5
6	NVIDIA	Selene	555,520	0.630	6	0.0630	9.9
7	DOE/SC/LBNL	Perlmutter	761,856	0.590	5	0.0709	8.3
8	FZJ	JUWELS BM	449,280	0.470	8	0.0440	10.0
9	GENCI-CINES	Adastra	319,072	0.303	11	0.0461	6.6
10	Pawsey Supercomputing Centre	Setonix - GPU	181,248	0.175	15	0.0272	6.4

HPL-MxP Benchmark

November 2022

Rank	Site	Computer	Cores	HPL-AI (Eflop/s)	TOP500 Rank	HPL Rmax (Eflop/s)	Speedup
1	DOE/SC/ORNL	Frontier	8,730,112	7.942	1	1.1020	7.2
2	EuroHPC/CSC	LUMI	2,174,976	2.168	3	0.3091	7.0
3	RIKEN	Fugaku	7,630,848	2.000	1	0.4420	4.5
4	EuroHPC/CINECA	Leonardo	1,463,616	1.842	4	0.1682	11.0
5	DOE/SC/ORNL	Summit	2,414,592	1.411	2	0.1486	9.5
6	NVIDIA	Selene	555,520	0.630	6	0.0630	9.9
7	DOE/SC/LBNL	Perlmutter	761,856	0.590	5	0.0709	8.3
8	FZJ	JUWELS BM	449,280	0.470	8	0.0440	10.0
9	GENCI-CINES	Adastra	319,072	0.303	11	0.0461	6.6
10	Pawsey Supercomputing Centre	Setonix - GPU	181,248	0.175	15	0.0272	6.4

HPL-MxP Benchmark

November 2022

Rank	Site	Computer	Cores	HPL-AI (Eflop/s)	TOP500 Rank	HPL Rmax (Eflop/s)	Speedup
1	DOE/SC/ORNL	Frontier	8,730,112	7.942	1	1.1020	7.2
2	EuroHPC/CSC	LUMI	2,174,976	2.168	3	0.3091	7.0
3	RIKEN	Fugaku	7,630,848	2.000	1	0.4420	4.5
4	EuroHPC/CINECA	Leonardo	1,463,616	1.842	4	0.1682	11.0
5	DOE/SC/ORNL	Summit	2,414,592	1.411	2	0.1486	9.5
6	NVIDIA	Selene	555,520	0.630	6	0.0630	9.9
7	DOE/SC/LBNL	Perlmutter	761,856	0.590	5	0.0709	8.3
8	FZJ	JUWELS BM	449,280	0.470	8	0.0440	10.0
9	GENCI-CINES	Adastra	319,072	0.303	11	0.0461	6.6
10	Pawsey Supercomputing Centre	Setonix - GPU	181,248	0.175	15	0.0272	6.4

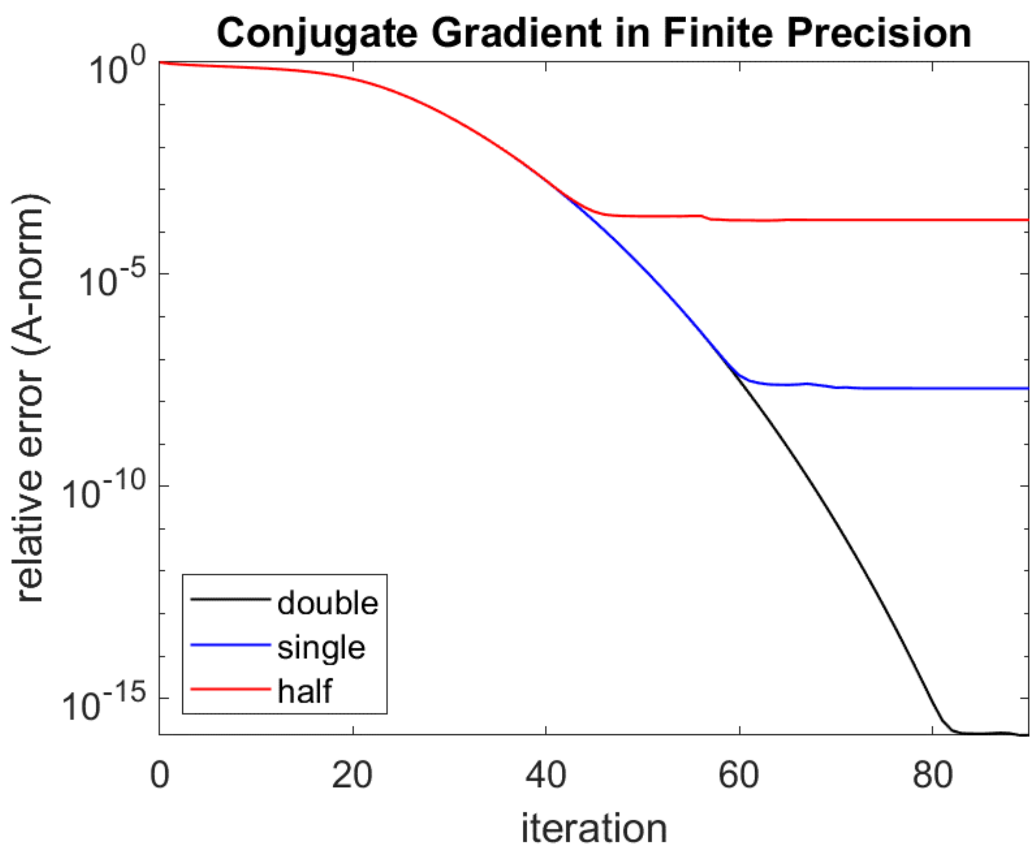
When Can I Use Low Precision?

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);
```



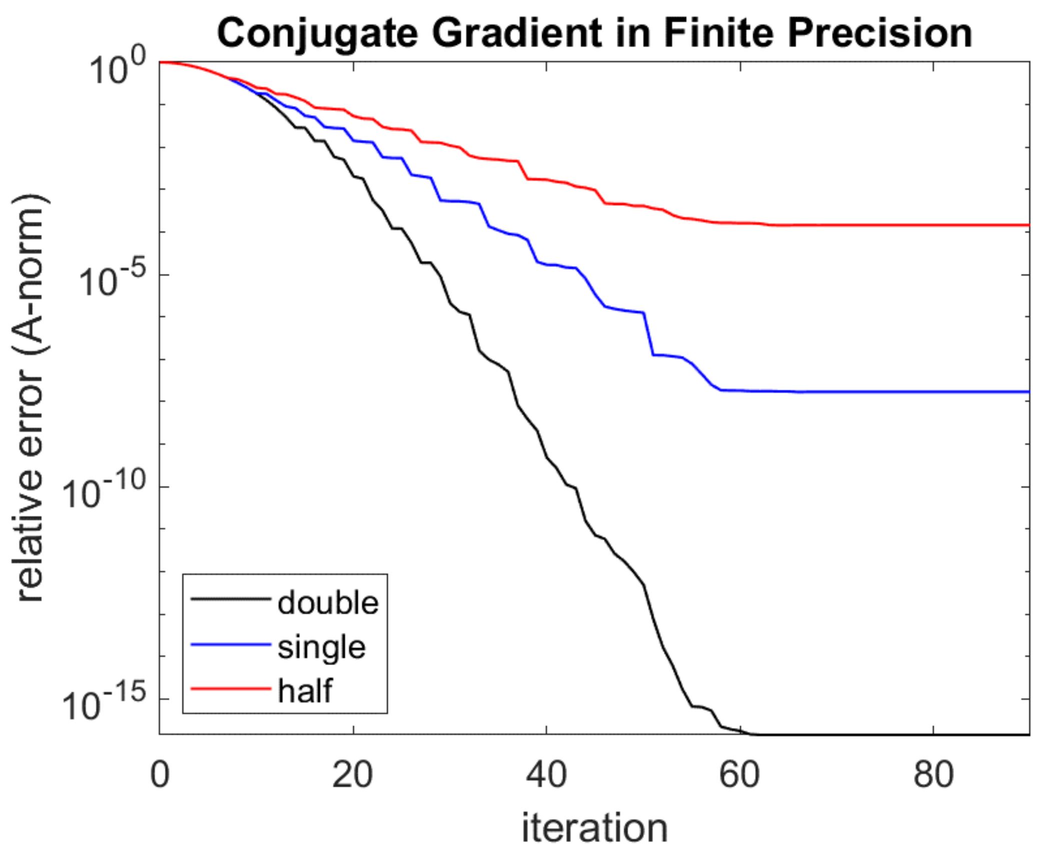
When Can I Use Low Precision?

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 = 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 $\boldsymbol{u_f}$)

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

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

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

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

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?

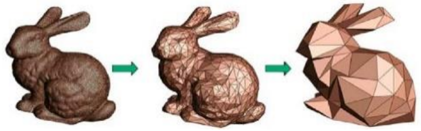
1. When low accuracy is needed
2. When a self-correction mechanism is available
3. When other approximations are being used

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

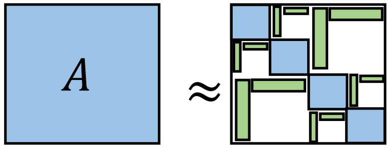
- E.g., reduced models, sparsification, low-rank approximations, randomization

Model Reduction

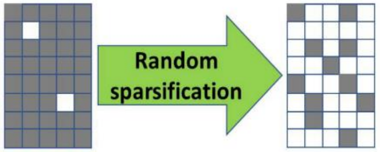


[Schilders, van der Vorst, Rommes, 2008]

Low-rank approximation



Sparsification, randomization



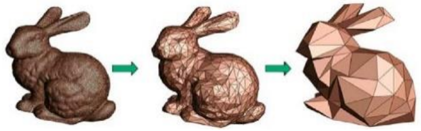
[Sinha, 2018]

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

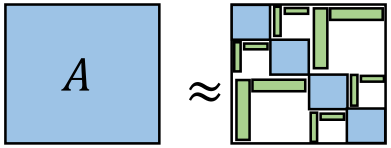
- E.g., reduced models, sparsification, low-rank approximations, randomization

Model Reduction

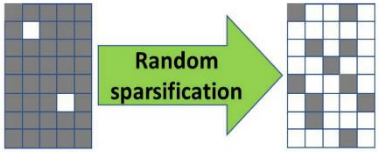


[Schilders, van der Vorst, Rommes, 2008]

Low-rank approximation



Sparsification, randomization



[Sinha, 2018]

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 $\boldsymbol{U} \in \mathbb{R}^{n\times k}$ **are $\boldsymbol{k}$ approximate eigenvectors of $\boldsymbol{A}$** and $U^T U = I$, **$\boldsymbol{\Theta}$ is diagonal with approximations to eigenvalues of $\boldsymbol{A}$**, 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:

$$A_N = (AQ)(Q^T AQ)^{+}(AQ)^T$$

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.

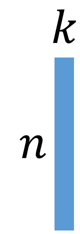
Randomized Nyström Approximation

[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

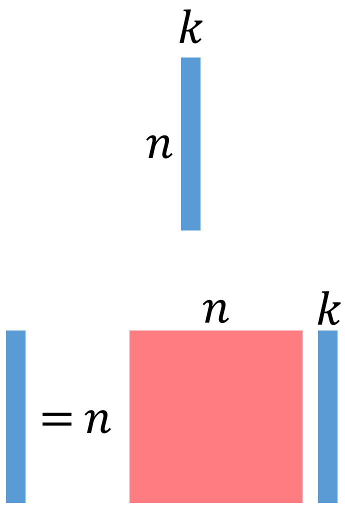
[Tropp et al., 2017]

Given sym. PSD matrix A, target rank k

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

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

$\boldsymbol{Y = AQ}$



Randomized Nyström Approximation

[Tropp et al., 2017]

Given sym. PSD matrix A, target rank k

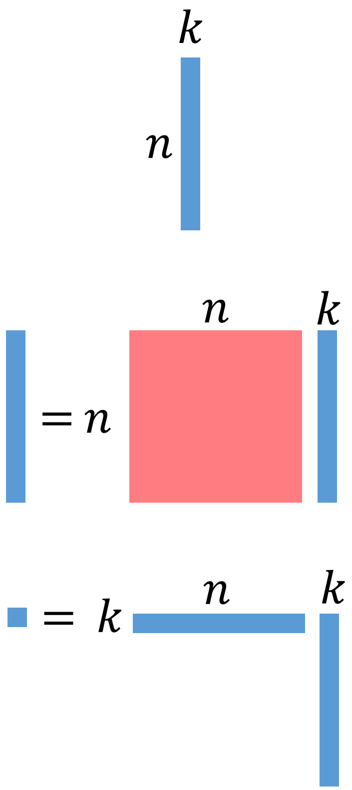
$G = \text{randn}(n, k)$

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

$\boldsymbol{Y} = \boldsymbol{AQ}$

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

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

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

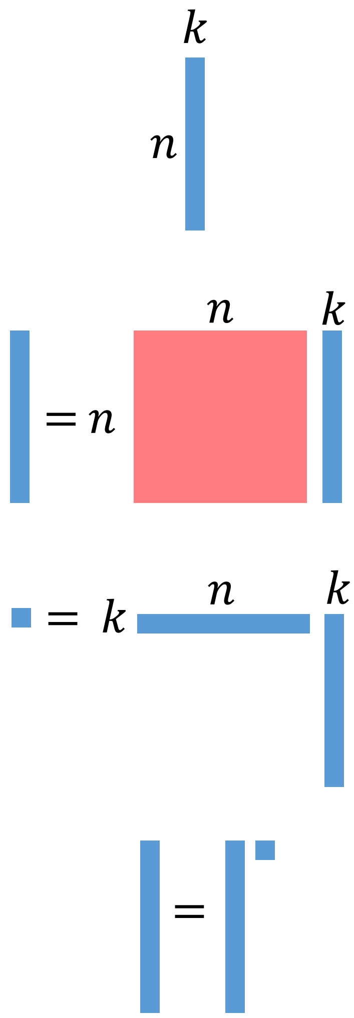
$\boldsymbol{Y = AQ}$

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

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

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

$\boldsymbol{Y = AQ}$

Compute shift ν; $Y_\nu = Y + \nu Q$

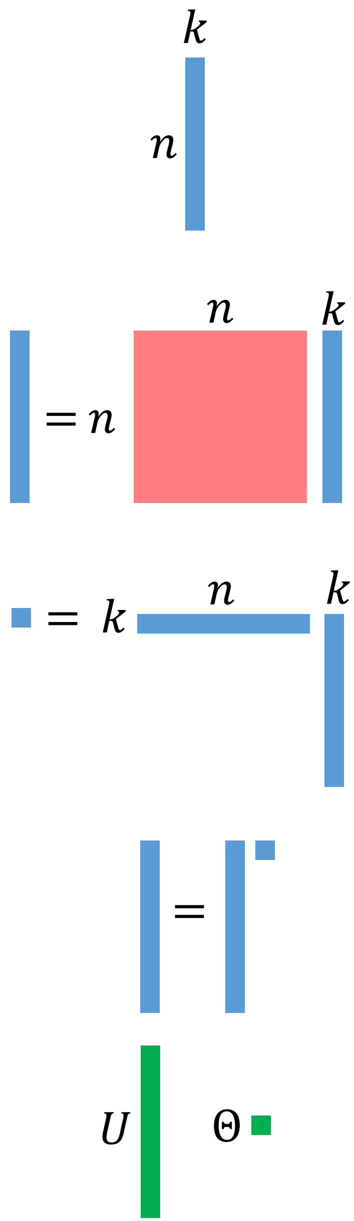
$B = Q^T Y_\nu$

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

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

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

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

$\boldsymbol{Y = AQ}$

Compute shift ν; $Y_\nu = Y + \nu Q$

$B = Q^T Y_\nu$

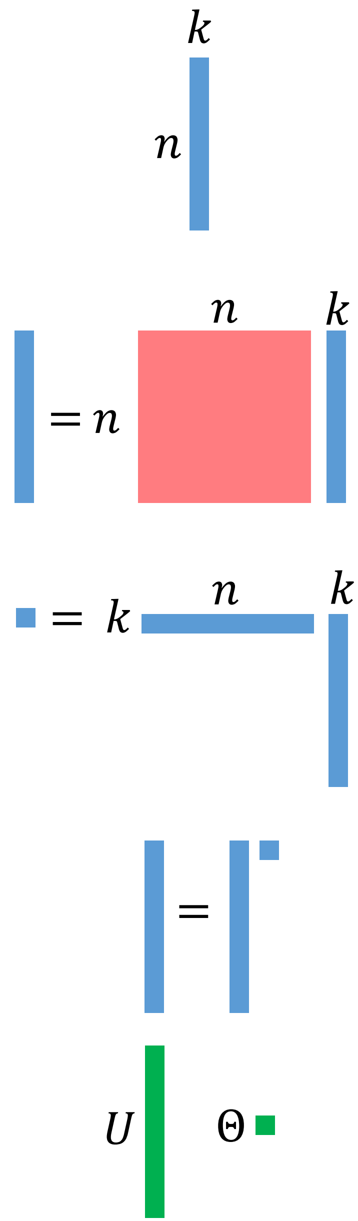
$C = \text{chol}((B + B^T)/2)$

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

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

exact Nyström approximation

Nyström approximation computed in finite precision

Error Bounds

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

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

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

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

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 $e^{-t^2/2} + c_1\alpha$,

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

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 $e^{-t^2/2} + c_1\alpha$,

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

where A_k is the best rank-k approximation of A

Interpretation: Likely that $\left\|A_N - \hat{A}_N\right\|_2 \gtrsim \|A - A_N\|_2$ 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]: With failure probability at most $e^{-t^2/2} + c_1\alpha$,

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

where A_k is the best rank-k approximation of A

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

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

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

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

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

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

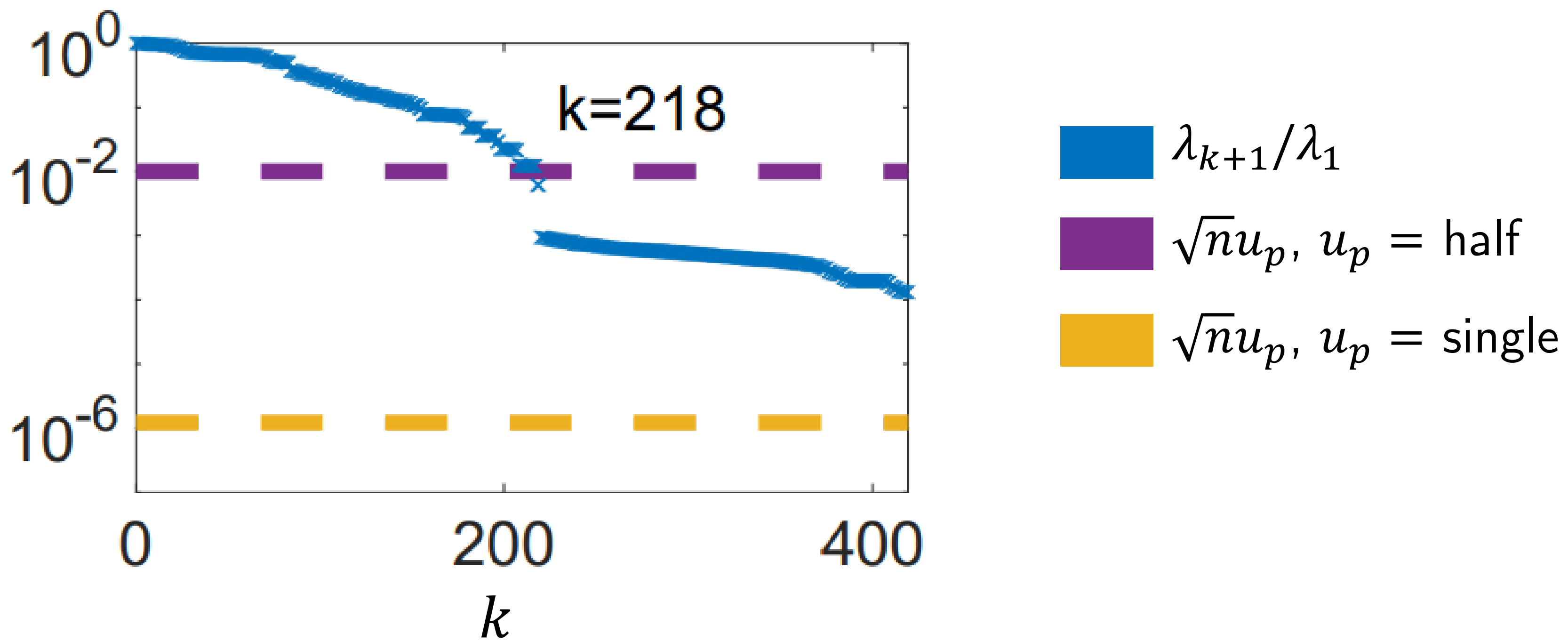
where the upper bound holds if $\mu > \|\mathcal{E}\|_2$.

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

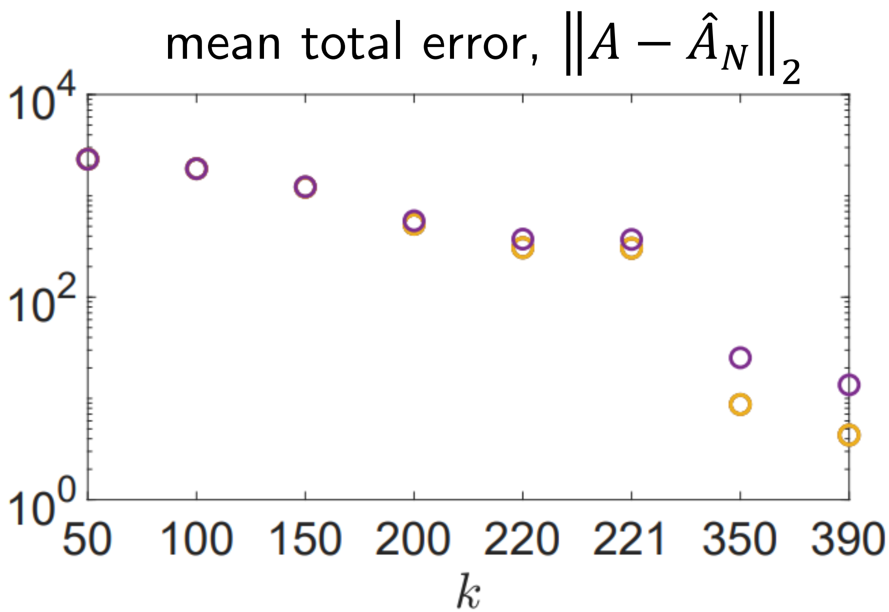
Matrix: bcsstm07, $n = 420$

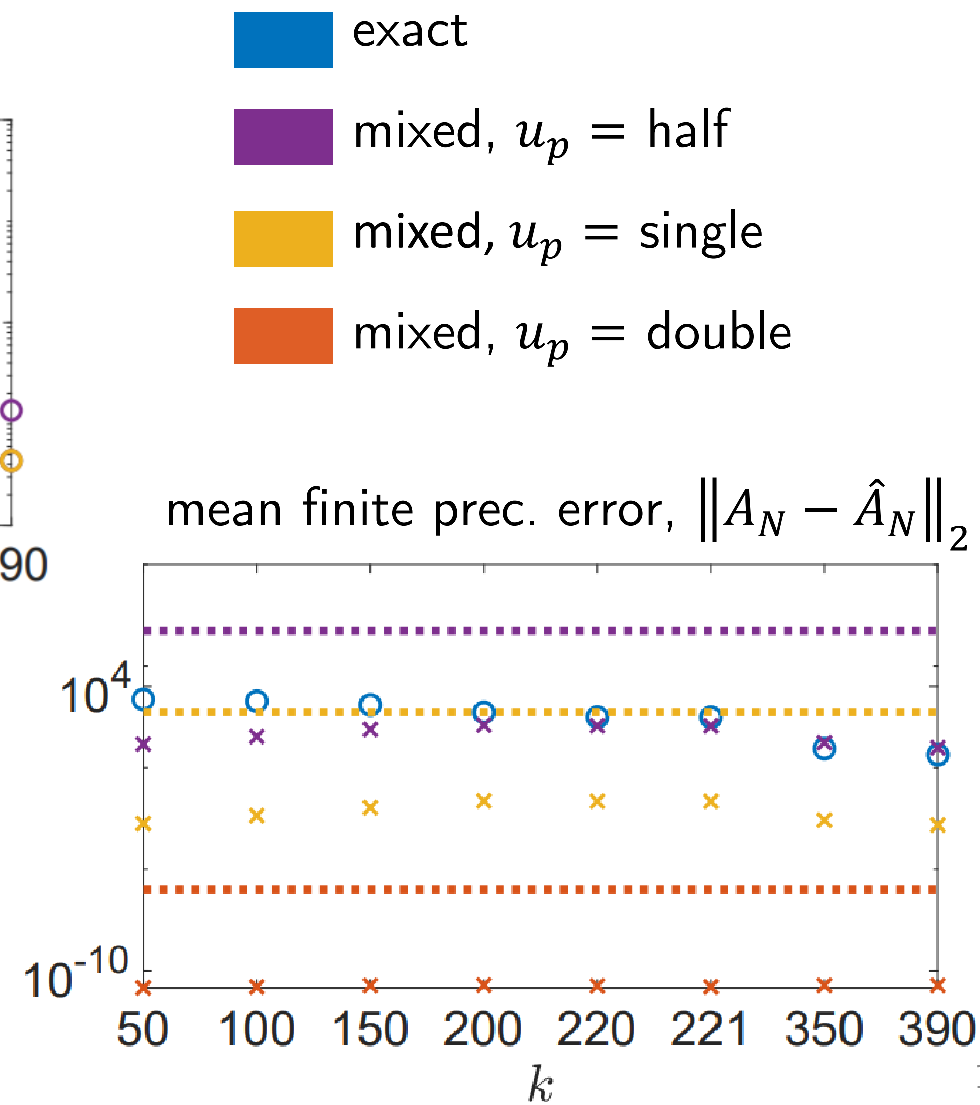


https://github.com/dauzickaite/mpNystrom

Numerical Experiment

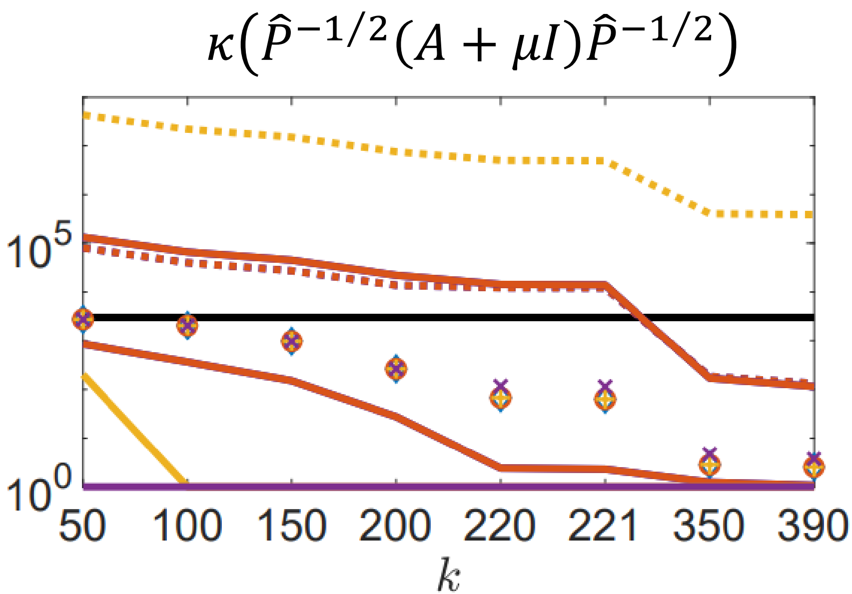
Matrix: bcsstm07, $n = 420$

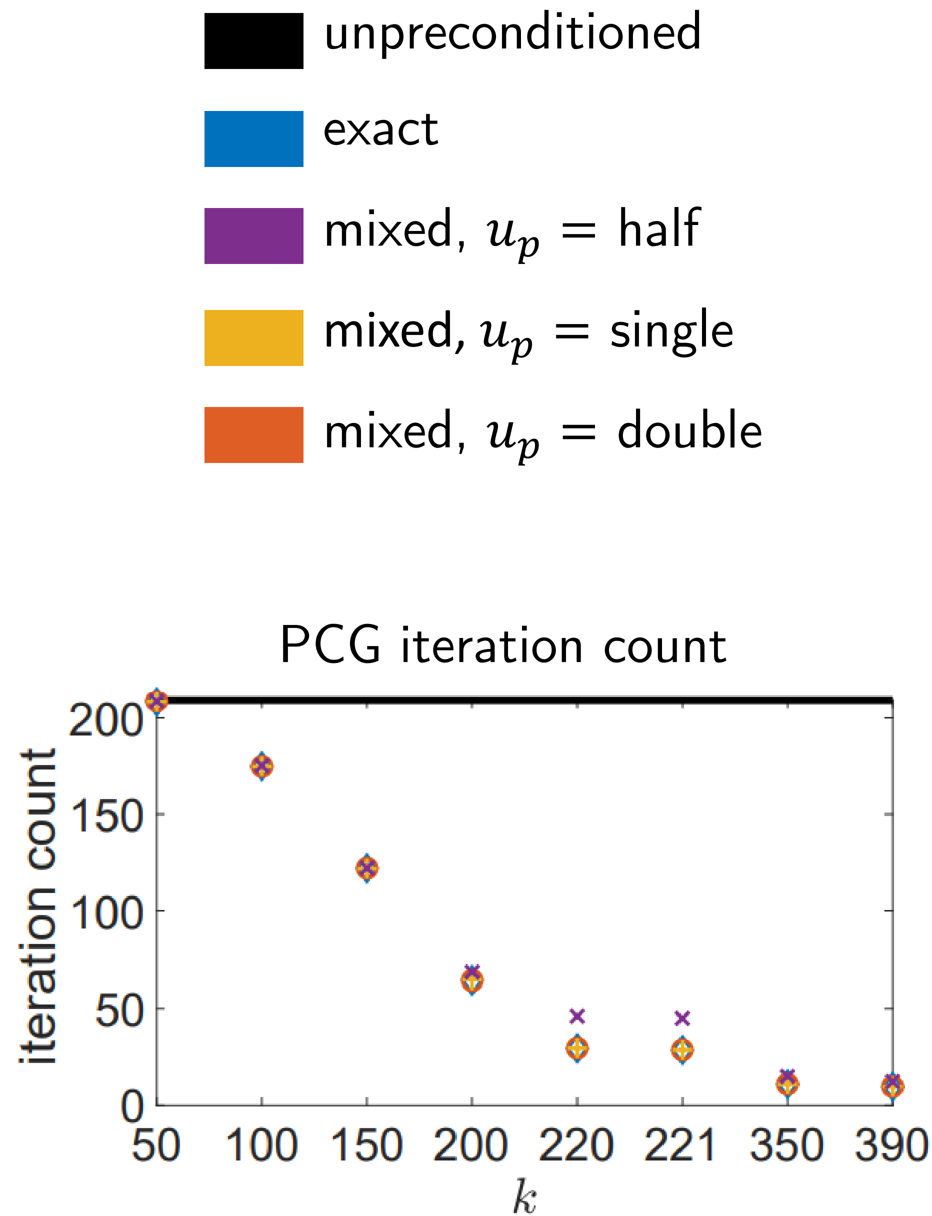




https://github.com/dauzickaite/mpNystrom

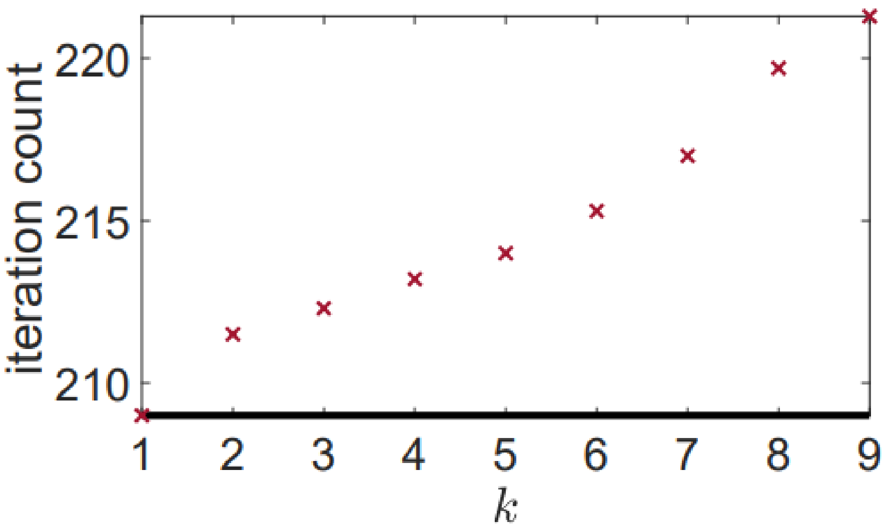
Numerical Experiment



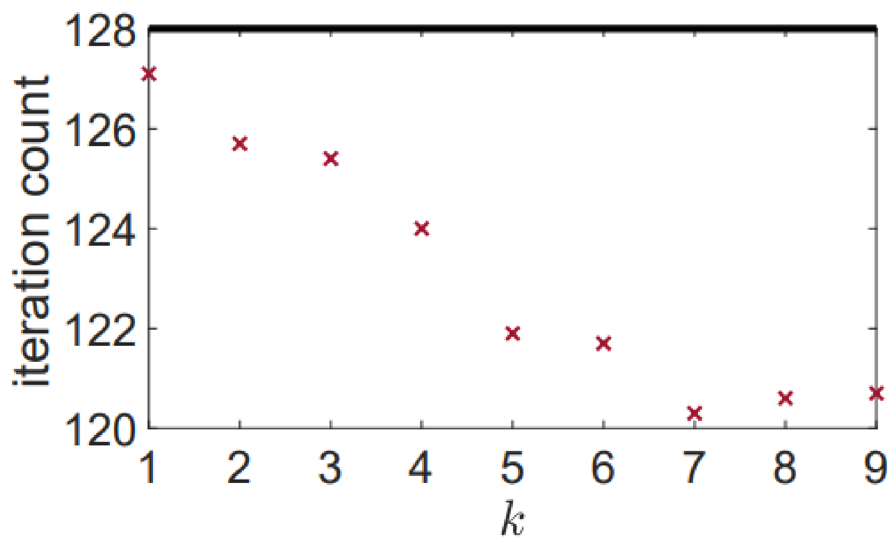


https://github.com/dauzickaite/mpNystrom

Quarter precision?



bcsstm07, iteration count



Journals, iteration count

https://github.com/dauzickaite/mpNystrom

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/