The s-Step Conjugate Gradient Method in Finite Precision

Erin C. Carson

Charles University

Prague, Czech Republic

SIAM CSE '19

February 25, 2019

This research was partially supported by OP RDE project No. CZ.02.2.69/0.0/0.0/16_027/0008495

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

Conjugate Gradient on the World's Fastest Computer

Summit - IBM Power System AC922

Site:	Oak Ridge National Laboratory
Manufacturer:	IBM
Cores:	2,282,544
Memory:	2,801,664 GB
Processor:	IBM POWER9 22C 3.07GHz
Interconnect:	Dual-rail Mellanox EDR Infiniband
Performance	
Theoretical peak:	187,659 TFlops/s
LINPACK benchmark:	122,300 Tflops/s
HPCG benchmark:	2,926 Tflops/s

Conjugate Gradient on the World's Fastest Computer

Summit - IBM Power System AC922

current #1 on top500

Site:	Oak Ridge National Laboratory
Manufacturer:	IBM
Cores:	2,282,544
Memory:	2,801,664 GB
Processor:	IBM POWER9 22C 3.07GHz
Interconnect:	Dual-rail Mellanox EDR Infiniband
Performance	
Theoretical peak:	187,659 TFlops/s
LINPACK benchmark:	122,300 Tflops/s
HPCG benchmark:	2,926 Tflops/s

Conjugate Gradient on the World's Fastest Computer

Summit - IBM Power System AC922

current #1 on top500

Site:	Oak Ridge National Laboratory
Manufacturer:	IBM
Cores:	2,282,544
Memory:	2,801,664 GB
Processor:	IBM POWER9 22C 3.07GHz
Interconnect:	Dual-rail Mellanox EDR Infiniband
Performance	
Theoretical peak:	187,659 TFlops/s
LINPACK benchmark:	122,300 Tflops/s
HPCG benchmark:	2,926 Tflops/s

LINPACK benchmark
(dense $Ax = b$, direct)
65% efficiency

Conjugate Gradient on the World's Fastest Computer

Summit - IBM Power System AC922

current #1 on top500

Site:	Oak Ridge National Laboratory
Manufacturer:	IBM
Cores:	2,282,544
Memory:	2,801,664 GB
Processor:	IBM POWER9 22C 3.07GHz
Interconnect:	Dual-rail Mellanox EDR Infiniband
Performance	
Theoretical peak:	187,659 TFlops/s
LINPACK benchmark:	122,300 Tflops/s
HPCG benchmark:	2,926 Tflops/s

LINPACK benchmark
(dense $Ax = b$, direct)
65% efficiency

HPCG benchmark
(sparse $Ax = b$, iterative)
1.5% efficiency

The Conjugate Gradient (CG) Method

$r_0 = b - Ax_0, \quad p_0 = r_0$

for $i = 1$:nmax

$$\alpha_{i-1} = \frac{r_{i-1}^T r_{i-1}}{p_{i-1}^T A p_{i-1}}$$

$$x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$$

$$r_i = r_{i-1} - \alpha_{i-1} A p_{i-1}$$

$$\beta_i = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$$

$$p_i = r_i + \beta_i p_{i-1}$$

end

The Conjugate Gradient (HSCG) Method

$r_0 = b - Ax_0, \quad p_0 = r_0$

for $i = 1$:nmax

$$\alpha_{i-1} = \frac{r_{i-1}^T r_{i-1}}{p_{i-1}^T A p_{i-1}}$$

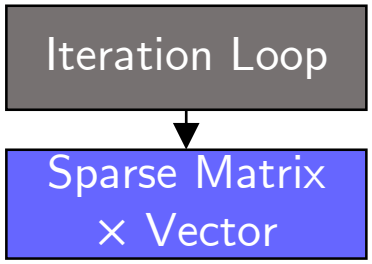
$$x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$$

$$r_i = r_{i-1} - \alpha_{i-1} A p_{i-1}$$

$$\beta_i = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$$

$$p_i = r_i + \beta_i p_{i-1}$$

end



The Conjugate Gradient (HSCG) Method

$r_0 = b - Ax_0, \quad p_0 = r_0$

for $i = 1$:nmax

$$\alpha_{i-1} = \frac{r_{i-1}^T r_{i-1}}{p_{i-1}^T A p_{i-1}}$$

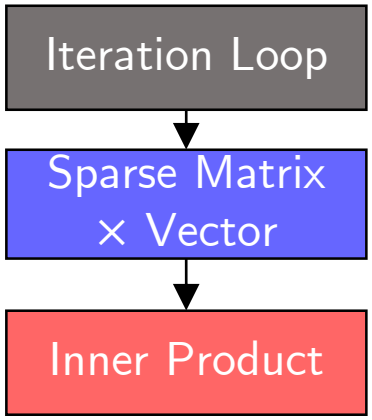
$$x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$$

$$r_i = r_{i-1} - \alpha_{i-1} A p_{i-1}$$

$$\beta_i = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$$

$$p_i = r_i + \beta_i p_{i-1}$$

end



The Conjugate Gradient (HSCG) Method

$r_0 = b - Ax_0, \quad p_0 = r_0$

for $i = 1$:nmax

$$\alpha_{i-1} = \frac{r_{i-1}^T r_{i-1}}{p_{i-1}^T A p_{i-1}}$$

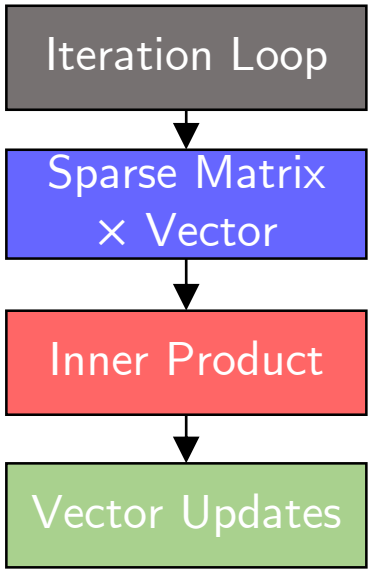
$$x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$$

$$r_i = r_{i-1} - \alpha_{i-1} A p_{i-1}$$

$$\beta_i = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$$

$$p_i = r_i + \beta_i p_{i-1}$$

end



The Conjugate Gradient (HSCG) Method

$r_0 = b - Ax_0, \quad p_0 = r_0$

for $i = 1$:nmax

$$\alpha_{i-1} = \frac{r_{i-1}^T r_{i-1}}{p_{i-1}^T A p_{i-1}}$$

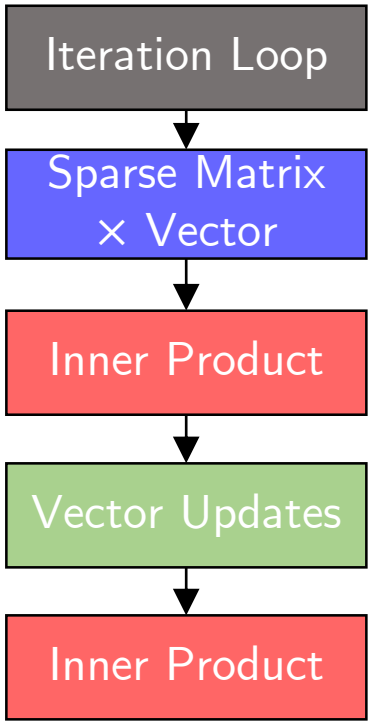
$$x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$$

$$r_i = r_{i-1} - \alpha_{i-1} A p_{i-1}$$

$$\beta_i = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$$

$$p_i = r_i + \beta_i p_{i-1}$$

end



The Conjugate Gradient (HSCG) Method

$r_0 = b - Ax_0, \quad p_0 = r_0$

for $i = 1$:nmax

$$\alpha_{i-1} = \frac{r_{i-1}^T r_{i-1}}{p_{i-1}^T A p_{i-1}}$$

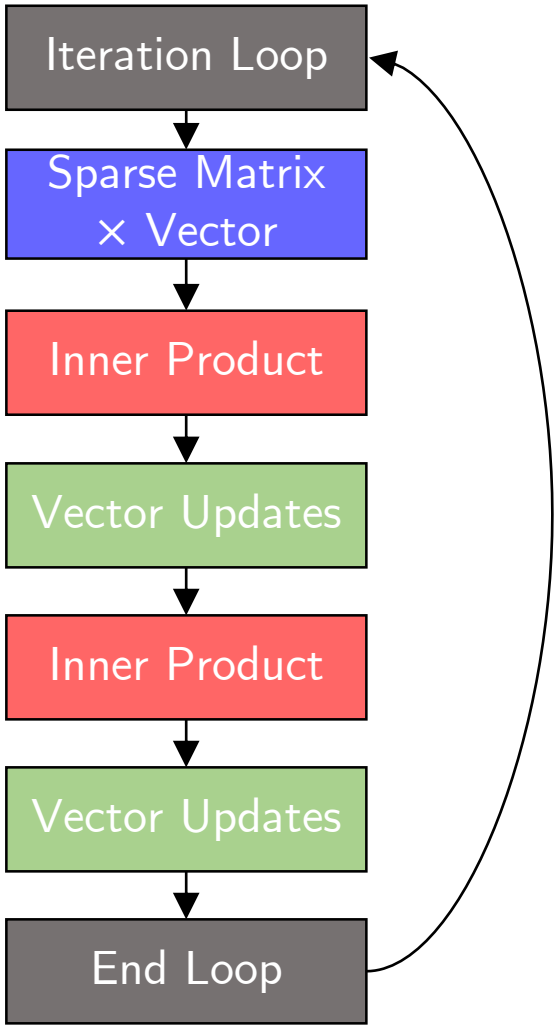
$$x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$$

$$r_i = r_{i-1} - \alpha_{i-1} A p_{i-1}$$

$$\beta_i = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$$

$$p_i = r_i + \beta_i p_{i-1}$$

end



The Conjugate Gradient (HSCG) Method

$r_0 = b - Ax_0, \quad p_0 = r_0$

for $i = 1$:nmax

$$\alpha_{i-1} = \frac{r_{i-1}^T r_{i-1}}{p_{i-1}^T A p_{i-1}}$$

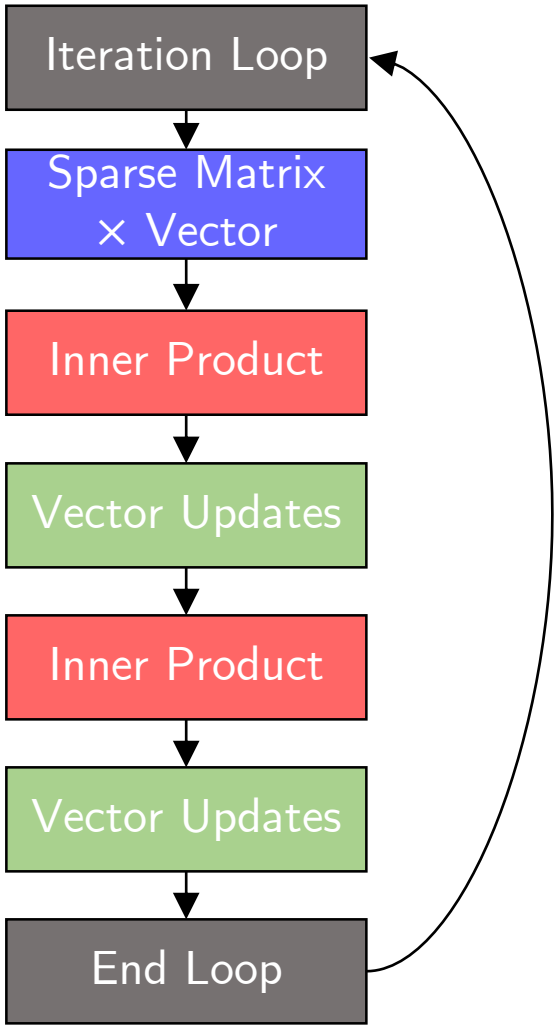
$$x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$$

$$r_i = r_{i-1} - \alpha_{i-1} A p_{i-1}$$

$$\beta_i = \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$$

$$p_i = r_i + \beta_i p_{i-1}$$

end



⇒ Communication bottleneck!

s-step Krylov subspace methods

- Idea: Compute blocks of s iterations at once
  - Compute updates in a different basis
  - Communicate every s iterations instead of every iteration
  - Reduces number of synchronizations per iteration by a factor of s

- An idea rediscovered many times...
- First related work: s-dimensional steepest descent, least squares
  - Khabaza ('63), Forsythe ('68), Marchuk and Kuznecov ('68)
- Flurry of work on s-step Krylov methods in '80s/early '90s: see, e.g., Van Rosendale (1983); Chronopoulos and Gear (1989)

- Resurgence of interest in recent years due to growing problem sizes; growing relative cost of communication

s-step CG

$r_0 = b - Ax_0, p_0 = r_0$

for $k = 0$:nmax$/s$

Compute $\mathcal{Y}_k$ and $\mathcal{B}_k$ such that $A\underline{\mathcal{Y}_k} = \mathcal{Y}_k\mathcal{B}_k$ and

$$\text{span}(\mathcal{Y}_k) = \mathcal{K}_{s+1}(A, p_{sk}) + \mathcal{K}_s(A, r_{sk})$$

$\mathcal{G}_k = \mathcal{Y}_k^T\mathcal{Y}_k$

$x'_0 = 0, r'_0 = e_{s+2}, p'_0 = e_1$

for $j = 1:s$

$$\alpha_{sk+j-1} = \frac{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}{p'^T_{j-1}\mathcal{G}_k\mathcal{B}_k p'_{j-1}}$$

$$x'_j = x'_{j-1} + \alpha_{sk+j-1}p'_{j-1}$$

$$r'_j = r'_{j-1} - \alpha_{sk+j-1}\mathcal{B}_k p'_{j-1}$$

$$\beta_{sk+j} = \frac{r'^T_j\mathcal{G}_k r'_j}{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}$$

$$p'_j = r'_j + \beta_{sk+j}p'_{j-1}$$

end

$[x_{s(k+1)} - x_{sk}, r_{s(k+1)}, p_{s(k+1)}] = \mathcal{Y}_k[x'_s, r'_s, p'_s]$

end

s-step CG

$r_0 = b - Ax_0, p_0 = r_0$

for $k = 0$:nmax$/s$

Compute $\mathcal{Y}_k$ and $\mathcal{B}_k$ such that $A\underline{\mathcal{Y}_k} = \mathcal{Y}_k\mathcal{B}_k$ and

$$\text{span}(\mathcal{Y}_k) = \mathcal{K}_{s+1}(A, p_{sk}) + \mathcal{K}_s(A, r_{sk})$$

$\mathcal{G}_k = \mathcal{Y}_k^T\mathcal{Y}_k$

$x'_0 = 0, r'_0 = e_{s+2}, p'_0 = e_1$

for $j = 1:s$

$$\alpha_{sk+j-1} = \frac{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}{p'^T_{j-1}\mathcal{G}_k\mathcal{B}_k p'_{j-1}}$$

$$x'_j = x'_{j-1} + \alpha_{sk+j-1}p'_{j-1}$$

$$r'_j = r'_{j-1} - \alpha_{sk+j-1}\mathcal{B}_k p'_{j-1}$$

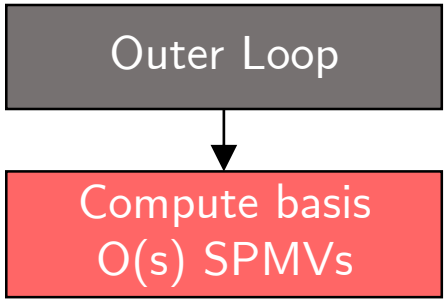
$$\beta_{sk+j} = \frac{r'^T_j\mathcal{G}_k r'_j}{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}$$

$$p'_j = r'_j + \beta_{sk+j}p'_{j-1}$$

end

$[x_{s(k+1)} - x_{sk}, r_{s(k+1)}, p_{s(k+1)}] = \mathcal{Y}_k[x'_s, r'_s, p'_s]$

end



s-step CG

$r_0 = b - Ax_0, p_0 = r_0$

for $k = 0$:nmax$/s$

Compute $\mathcal{Y}_k$ and $\mathcal{B}_k$ such that $A\underline{\mathcal{Y}_k} = \mathcal{Y}_k\mathcal{B}_k$ and

$$\text{span}(\mathcal{Y}_k) = \mathcal{K}_{s+1}(A, p_{sk}) + \mathcal{K}_s(A, r_{sk})$$

$\mathcal{G}_k = \mathcal{Y}_k^T\mathcal{Y}_k$

$x'_0 = 0, r'_0 = e_{s+2}, p'_0 = e_1$

for $j = 1:s$

$$\alpha_{sk+j-1} = \frac{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}{p'^T_{j-1}\mathcal{G}_k\mathcal{B}_k p'_{j-1}}$$

$$x'_j = x'_{j-1} + \alpha_{sk+j-1}p'_{j-1}$$

$$r'_j = r'_{j-1} - \alpha_{sk+j-1}\mathcal{B}_k p'_{j-1}$$

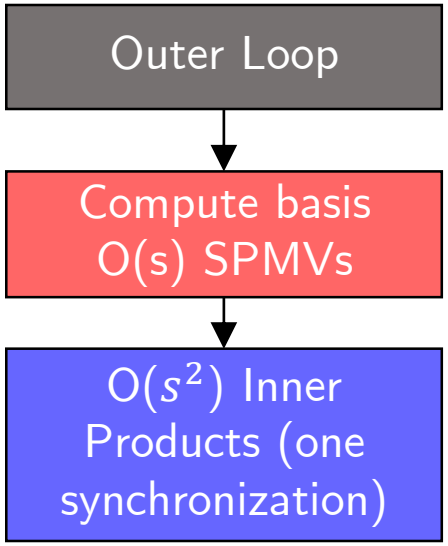
$$\beta_{sk+j} = \frac{r'^T_j\mathcal{G}_k r'_j}{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}$$

$$p'_j = r'_j + \beta_{sk+j}p'_{j-1}$$

end

$[x_{s(k+1)} - x_{sk}, r_{s(k+1)}, p_{s(k+1)}] = \mathcal{Y}_k[x'_s, r'_s, p'_s]$

end



s-step CG

$r_0 = b - Ax_0, p_0 = r_0$

for $k = 0$:nmax$/s$

Compute $\mathcal{Y}_k$ and $\mathcal{B}_k$ such that $A\underline{\mathcal{Y}_k} = \mathcal{Y}_k\mathcal{B}_k$ and

$$\text{span}(\mathcal{Y}_k) = \mathcal{K}_{s+1}(A, p_{sk}) + \mathcal{K}_s(A, r_{sk})$$

$\mathcal{G}_k = \mathcal{Y}_k^T\mathcal{Y}_k$

$x'_0 = 0, r'_0 = e_{s+2}, p'_0 = e_1$

for $j = 1:s$

$$\alpha_{sk+j-1} = \frac{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}{p'^T_{j-1}\mathcal{G}_k\mathcal{B}_k p'_{j-1}}$$

$$x'_j = x'_{j-1} + \alpha_{sk+j-1}p'_{j-1}$$

$$r'_j = r'_{j-1} - \alpha_{sk+j-1}\mathcal{B}_k p'_{j-1}$$

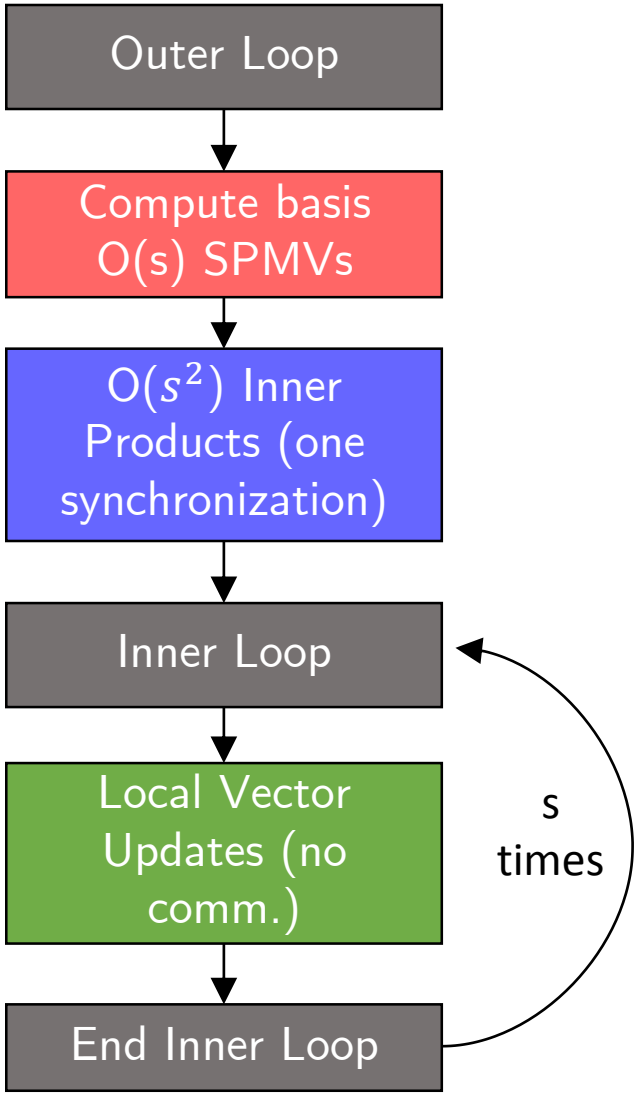
$$\beta_{sk+j} = \frac{r'^T_j\mathcal{G}_k r'_j}{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}$$

$$p'_j = r'_j + \beta_{sk+j}p'_{j-1}$$

end

$[x_{s(k+1)} - x_{sk}, r_{s(k+1)}, p_{s(k+1)}] = \mathcal{Y}_k[x'_s, r'_s, p'_s]$

end



s-step CG

$r_0 = b - Ax_0, p_0 = r_0$

for $k = 0$:nmax/s

Compute $\mathcal{Y}_k$ and $\mathcal{B}_k$ such that $A\underline{\mathcal{Y}_k} = \mathcal{Y}_k\mathcal{B}_k$ and

$$\text{span}(\mathcal{Y}_k) = \mathcal{K}_{s+1}(A, p_{sk}) + \mathcal{K}_s(A, r_{sk})$$

$\mathcal{G}_k = \mathcal{Y}_k^T\mathcal{Y}_k$

$x'_0 = 0, r'_0 = e_{s+2}, p'_0 = e_1$

for $j = 1:s$

$$\alpha_{sk+j-1} = \frac{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}{p'^T_{j-1}\mathcal{G}_k\mathcal{B}_k p'_{j-1}}$$

$$x'_j = x'_{j-1} + \alpha_{sk+j-1}p'_{j-1}$$

$$r'_j = r'_{j-1} - \alpha_{sk+j-1}\mathcal{B}_k p'_{j-1}$$

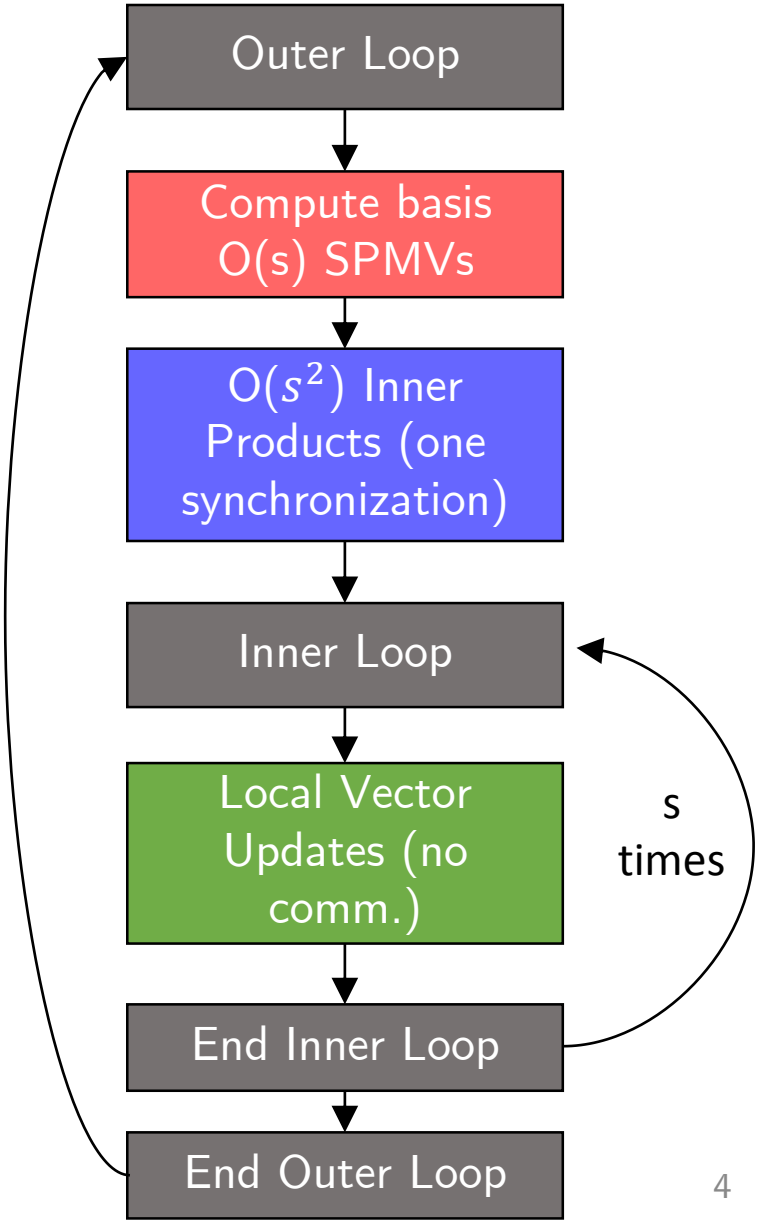
$$\beta_{sk+j} = \frac{r'^T_j\mathcal{G}_k r'_j}{r'^T_{j-1}\mathcal{G}_k r'_{j-1}}$$

$$p'_j = r'_j + \beta_{sk+j}p'_{j-1}$$

end

$[x_{s(k+1)} - x_{sk}, r_{s(k+1)}, p_{s(k+1)}] = \mathcal{Y}_k[x'_s, r'_s, p'_s]$

end

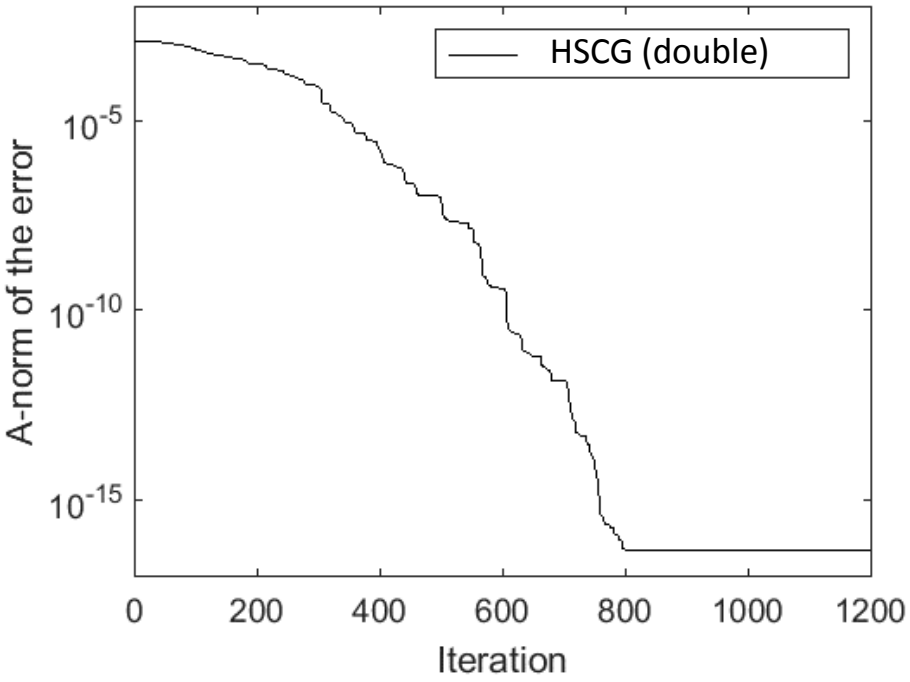


The effects of finite precision

Well-known that roundoff error has two effects:

1. Delay of convergence
   - No longer have exact Krylov subspace
   - Can lose numerical rank deficiency
   - Residuals no longer orthogonal - Minimization of $\|x - x_i\|_A$ no longer exact

2. Loss of attainable accuracy
   - Rounding errors cause true residual $b - Ax_i$ and updated residual r_i deviate!

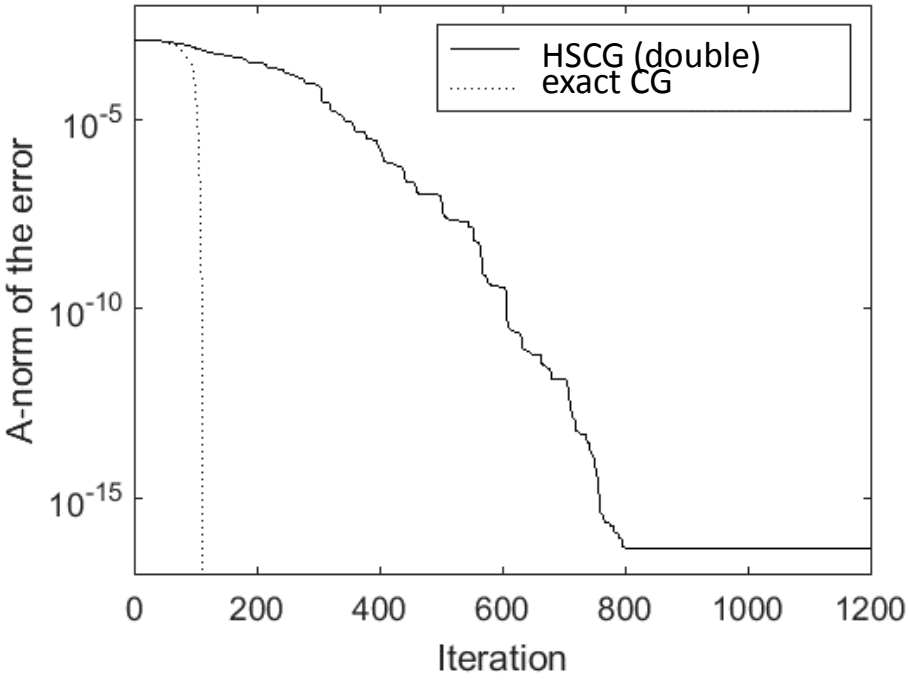


A: bcsstk03 from SuiteSparse,
b: equal components in the eigenbasis of $A, \|b\| = 1$
$N = 112, \kappa(A) \approx 7e6$

The effects of finite precision

Well-known that roundoff error has two effects:

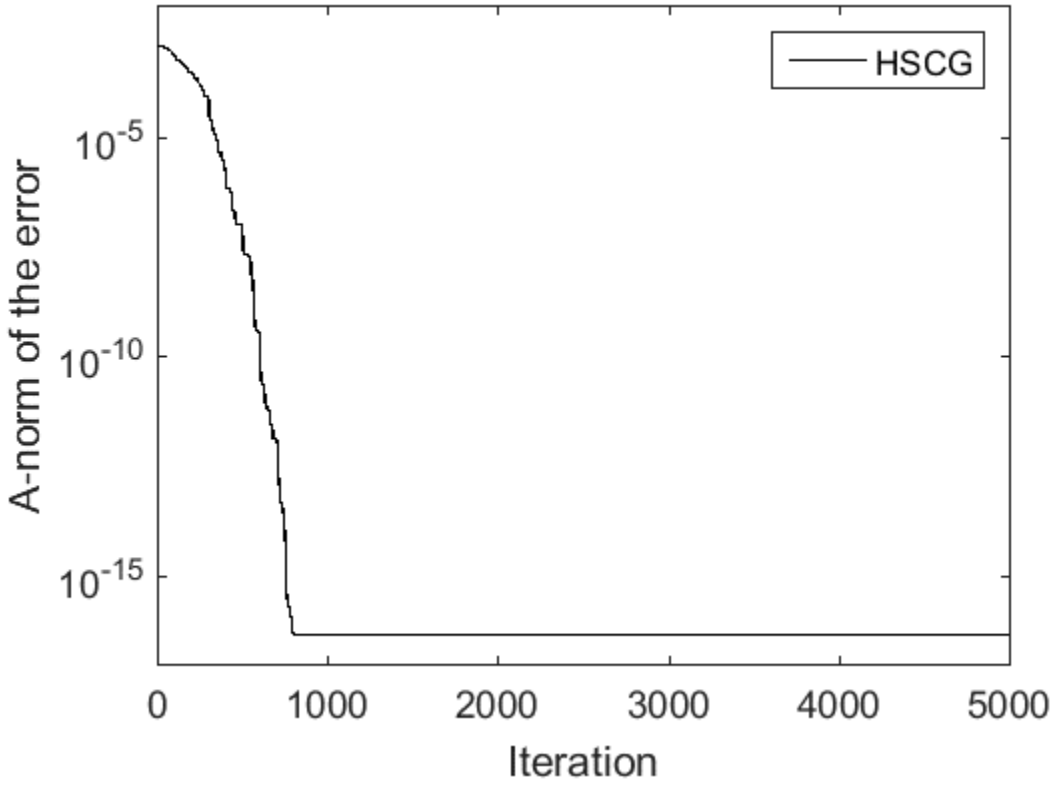
1. Delay of convergence
   - No longer have exact Krylov subspace
   - Can lose numerical rank deficiency
   - Residuals no longer orthogonal - Minimization of $\|x - x_i\|_A$ no longer exact
2. Loss of attainable accuracy
   - Rounding errors cause true residual $b - Ax_i$ and updated residual r_i deviate!



A: bcsstk03 from SuiteSparse,
b: equal components in the eigenbasis of A, $\|b\| = 1$
$N = 112, \kappa(A) \approx 7e6$

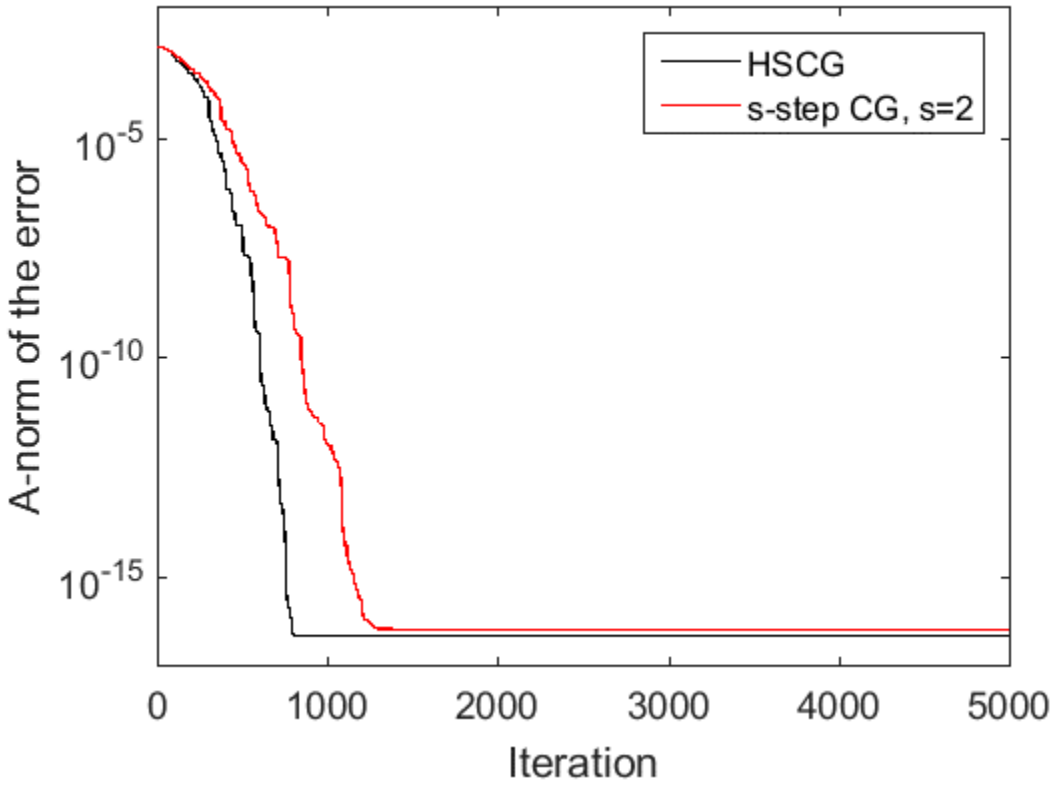
Much work on these results for CG; See Meurant and Strakoš (2006) for a thorough summary of early developments in finite precision analysis of Lanczos and CG

s-step CG


HSCG
A-norm of the error
10^-5
10^-10
10^-15
0
1000
2000
3000
4000
5000
Iteration

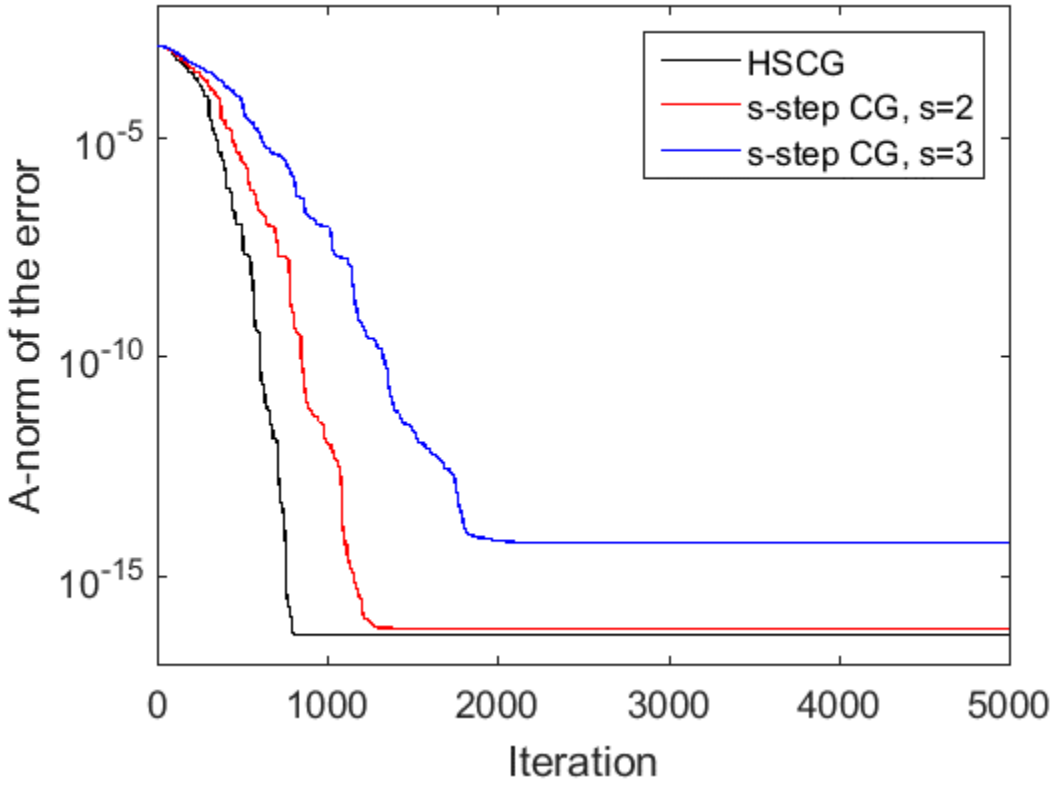
s-step CG

s-step CG with monomial basis ($\mathcal{Y} = [p_i, Ap_i, \ldots, A^s p_i, r_i, Ar_i, \ldots A^{s-1} r_i]$)



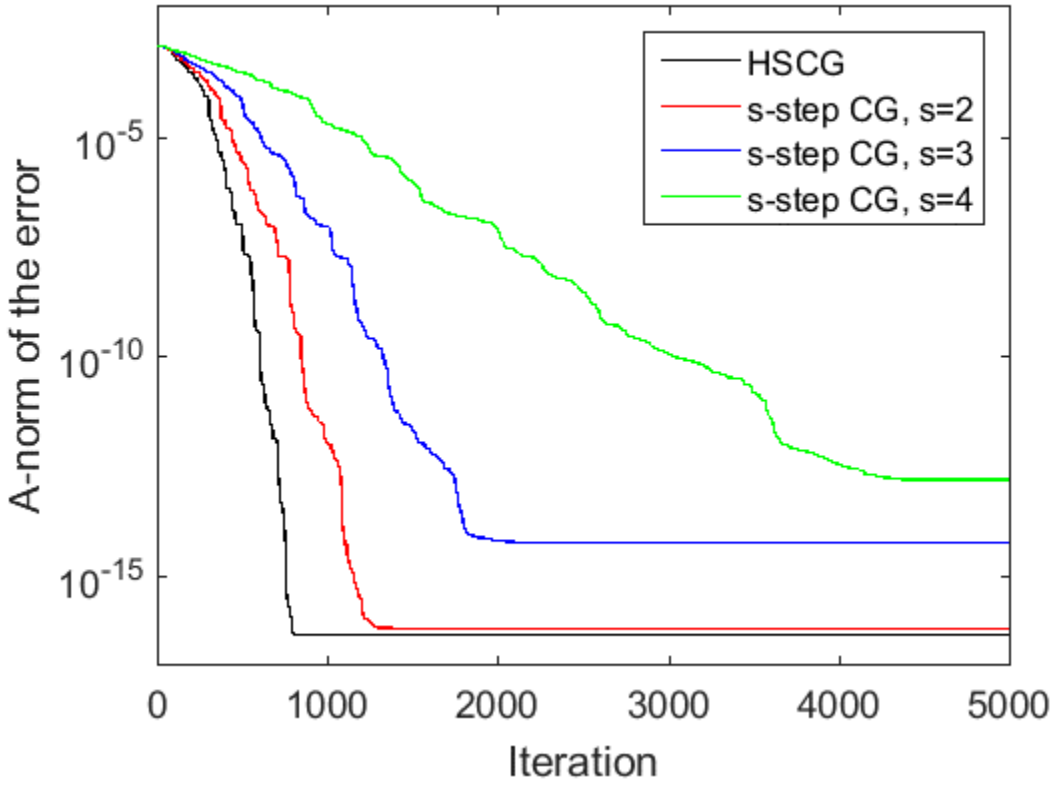
s-step CG

s-step CG with monomial basis ($\mathcal{Y} = [p_i, Ap_i, \ldots, A^s p_i, r_i, Ar_i, \ldots A^{s-1} r_i]$)



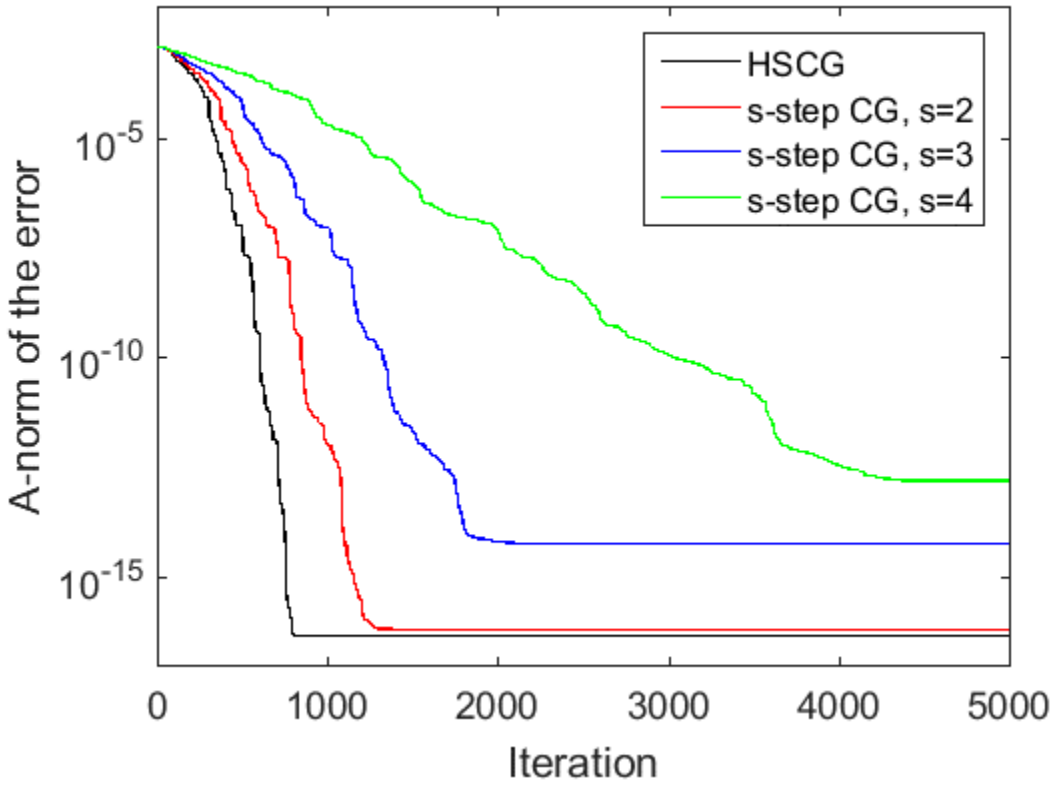
s-step CG

s-step CG with monomial basis ($\mathcal{Y} = [p_i, Ap_i, \ldots, A^s p_i, r_i, Ar_i, \ldots A^{s-1} r_i]$)



s-step CG

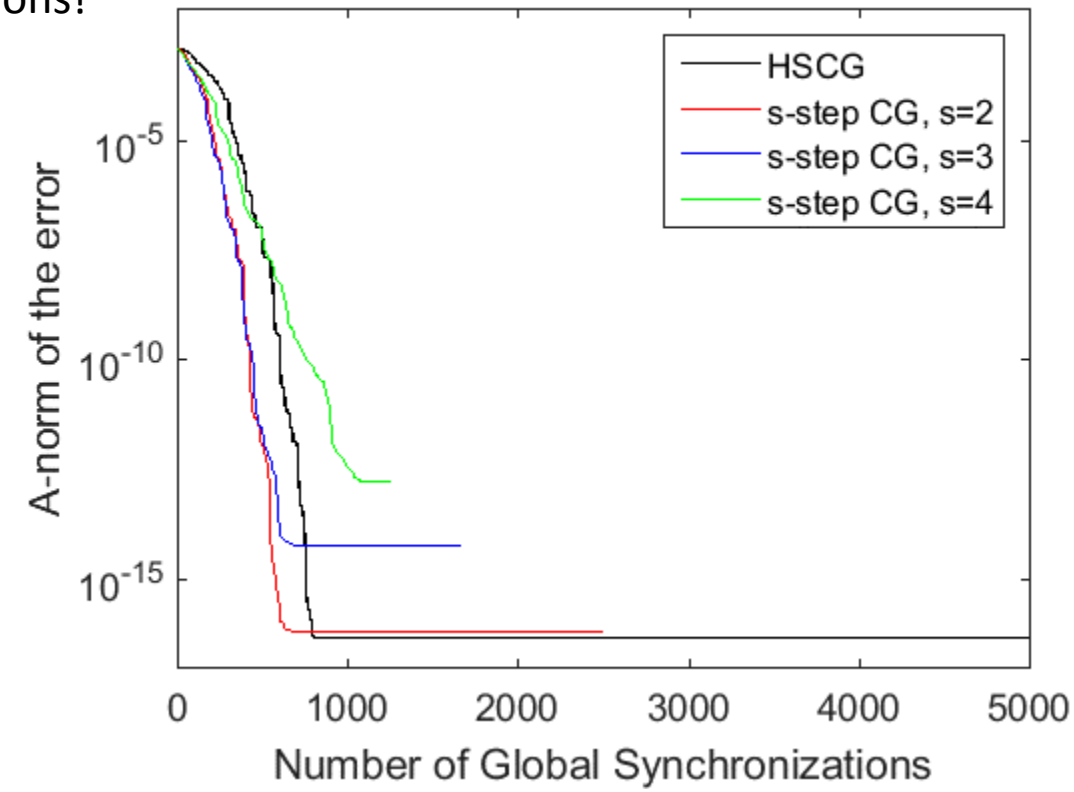
s-step CG with monomial basis ($\mathcal{Y} = [p_i, Ap_i, \ldots, A^s p_i, r_i, Ar_i, \ldots A^{s-1} r_i]$)



Convergence delay and attainable accuracy worse with increasing s!

s-step CG

Even assuming perfect parallel scalability with s (which is usually not the case due to extra SpMVs and inner products), already at $s = 4$ we are worse than HSCG in terms of number of synchronizations!



Maximum attainable accuracy

- Accuracy $\|x - \hat{x}_i\|$ generally not computable, *but* $x - \hat{x}_i = A^{-1}(b - A\hat{x}_i)$
- Size of the true residual, $\|b - A\hat{x}_i\|$, used as computable measure of accuracy

Maximum attainable accuracy

- Accuracy $\|x - \hat{x}_i\|$ generally not computable, *but* $x - \hat{x}_i = A^{-1}(b - A\hat{x}_i)$
- Size of the true residual, $\|b - A\hat{x}_i\|$, used as computable measure of accuracy
- Rounding errors cause the **true residual, $\boldsymbol{b - A\hat{x}_i}$**, and the **updated residual, $\boldsymbol{\hat{r}_i}$**, to deviate

Maximum attainable accuracy

- Accuracy $\|x - \hat{x}_i\|$ generally not computable, *but* $x - \hat{x}_i = A^{-1}(b - A\hat{x}_i)$
- Size of the true residual, $\|b - A\hat{x}_i\|$, used as computable measure of accuracy
- Rounding errors cause the **true residual, $\boldsymbol{b - A\hat{x}_i}$**, and the **updated residual, $\boldsymbol{\hat{r}_i}$**, to deviate

- Writing $b - A\hat{x}_i = \hat{r}_i + b - A\hat{x}_i - \hat{r}_i$,

$$\|b - A\hat{x}_i\| \leq \|\hat{r}_i\| + \|b - A\hat{x}_i - \hat{r}_i\|$$

Maximum attainable accuracy

- Accuracy $\|x - \hat{x}_i\|$ generally not computable, *but* $x - \hat{x}_i = A^{-1}(b - A\hat{x}_i)$
- Size of the true residual, $\|b - A\hat{x}_i\|$, used as computable measure of accuracy
- Rounding errors cause the **true residual, $\boldsymbol{b - A\hat{x}_i}$**, and the **updated residual, $\boldsymbol{\hat{r}_i}$**, to deviate

- Writing $b - A\hat{x}_i = \hat{r}_i + b - A\hat{x}_i - \hat{r}_i$,

$$\|b - A\hat{x}_i\| \leq \|\hat{r}_i\| + \|b - A\hat{x}_i - \hat{r}_i\|$$

  - As $\|\hat{r}_i\| \to 0$, $\|b - A\hat{x}_i\|$ depends on $\|b - A\hat{x}_i - \hat{r}_i\|$

Maximum attainable accuracy

- Accuracy $\|x - \hat{x}_i\|$ generally not computable, *but* $x - \hat{x}_i = A^{-1}(b - A\hat{x}_i)$
- Size of the true residual, $\|b - A\hat{x}_i\|$, used as computable measure of accuracy
- Rounding errors cause the **true residual, $\boldsymbol{b - A\hat{x}_i}$**, and the **updated residual, $\boldsymbol{\hat{r}_i}$**, to deviate

- Writing $b - A\hat{x}_i = \hat{r}_i + b - A\hat{x}_i - \hat{r}_i$,

$$\|b - A\hat{x}_i\| \leq \|\hat{r}_i\| + \|b - A\hat{x}_i - \hat{r}_i\|$$

  - As $\|\hat{r}_i\| \to 0$, $\|b - A\hat{x}_i\|$ depends on $\|b - A\hat{x}_i - \hat{r}_i\|$

- Many results on bounding attainable accuracy, e.g.: Greenbaum (1989, 1994, 1997), Sleijpen, van der Vorst and Fokkema (1994), Sleijpen, van der Vorst and Modersitzki (2001), Björck, Elfving and Strakoš (1998) and Gutknecht and Strakoš (2000).

Maximum attainable accuracy of HSCG

- In finite precision HSCG, iterates are updated by

$$\hat{x}_i = \hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} - \boldsymbol{\delta x_i}$$ and $$\hat{r}_i = \hat{r}_{i-1} - \hat{\alpha}_{i-1}A\hat{p}_{i-1} - \boldsymbol{\delta r_i}$$

Maximum attainable accuracy of HSCG

- In finite precision HSCG, iterates are updated by

$$\hat{x}_i = \hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} - \boldsymbol{\delta x_i} \quad \text{and} \quad \hat{r}_i = \hat{r}_{i-1} - \hat{\alpha}_{i-1}A\hat{p}_{i-1} - \boldsymbol{\delta r_i}$$

- Let $f_i \equiv b - A\hat{x}_i - \hat{r}_i$

Maximum attainable accuracy of HSCG

- In finite precision HSCG, iterates are updated by

$$\hat{x}_i = \hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} - \boldsymbol{\delta x_i} \quad \text{and} \quad \hat{r}_i = \hat{r}_{i-1} - \hat{\alpha}_{i-1}A\hat{p}_{i-1} - \boldsymbol{\delta r_i}$$

- Let $f_i \equiv b - A\hat{x}_i - \hat{r}_i$

$$f_i = b - A(\hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} - \delta x_i) - (\hat{r}_{i-1} - \hat{\alpha}_{i-1}A\hat{p}_{i-1} - \delta r_i)$$

Maximum attainable accuracy of HSCG

- In finite precision HSCG, iterates are updated by

$$\hat{x}_i = \hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} - \boldsymbol{\delta x_i} \quad \text{and} \quad \hat{r}_i = \hat{r}_{i-1} - \hat{\alpha}_{i-1}A\hat{p}_{i-1} - \boldsymbol{\delta r_i}$$

- Let $f_i \equiv b - A\hat{x}_i - \hat{r}_i$

$$\begin{aligned} f_i &= b - A(\hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} - \delta x_i) - (\hat{r}_{i-1} - \hat{\alpha}_{i-1}A\hat{p}_{i-1} - \delta r_i) \\ &= f_{i-1} + A\delta x_i + \delta r_i \end{aligned}$$

Maximum attainable accuracy of HSCG

- In finite precision HSCG, iterates are updated by

$$\hat{x}_i = \hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} - \boldsymbol{\delta x_i} \quad \text{and} \quad \hat{r}_i = \hat{r}_{i-1} - \hat{\alpha}_{i-1}A\hat{p}_{i-1} - \boldsymbol{\delta r_i}$$

- Let $f_i \equiv b - A\hat{x}_i - \hat{r}_i$

$$\begin{aligned} f_i &= b - A(\hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} - \delta x_i) - (\hat{r}_{i-1} - \hat{\alpha}_{i-1}A\hat{p}_{i-1} - \delta r_i) \\ &= f_{i-1} + A\delta x_i + \delta r_i \\ &= f_0 + \sum_{m=1}^{i}(A\delta x_m + \delta r_m) \end{aligned}$$

Maximum attainable accuracy of HSCG

- In finite precision HSCG, iterates are updated by

$$\hat{x}_i = \hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} - \boldsymbol{\delta x_i} \quad \text{and} \quad \hat{r}_i = \hat{r}_{i-1} - \hat{\alpha}_{i-1}A\hat{p}_{i-1} - \boldsymbol{\delta r_i}$$

- Let $f_i \equiv b - A\hat{x}_i - \hat{r}_i$

$$\begin{aligned} f_i &= b - A(\hat{x}_{i-1} + \hat{\alpha}_{i-1}\hat{p}_{i-1} - \delta x_i) - (\hat{r}_{i-1} - \hat{\alpha}_{i-1}A\hat{p}_{i-1} - \delta r_i) \\ &= f_{i-1} + A\delta x_i + \delta r_i \\ &= f_0 + \textstyle\sum_{m=1}^{i}(A\delta x_m + \delta r_m) \end{aligned}$$

$\|f_i\| \le O(\varepsilon)\sum_{m=0}^{i} N_A\|A\|\|\hat{x}_m\| + \|\hat{r}_m\|$ van der Vorst and Ye, 2000

$\|f_i\| \le O(\varepsilon)\|A\|\left(\|x\| + \max_{m=0,\ldots,i}\|\hat{x}_m\|\right)$ Greenbaum, 1997

$\|f_i\| \le O(\varepsilon)N_A\||A|\|\|A^{-1}\|\sum_{m=0}^{i}\|\hat{r}_m\|$ Sleijpen and van der Vorst, 1995

Sources of local roundoff error in s-step CG

Computing the s-step Krylov subspace basis:

$$A\underline{\hat{\mathcal{Y}}_k} = \hat{\mathcal{Y}}_k \mathcal{B}_k + \Delta\mathcal{Y}_k$$

Updating coordinate vectors in the inner loop:

$$\hat{x}'_{k,j} = \hat{x}'_{k,j-1} + \hat{q}'_{k,j-1} + \xi_{k,j}$$
$$\hat{r}'_{k,j} = \hat{r}'_{k,j-1} - \mathcal{B}_k\, \hat{q}'_{k,j-1} + \eta_{k,j}$$
$$\text{with} \quad \hat{q}'_{k,j-1} = \mathrm{fl}(\hat{\alpha}_{sk+j-1}\hat{p}'_{k,j-1})$$

Recovering CG vectors for use in next outer loop:

$$\hat{x}_{sk+j} = \hat{\mathcal{Y}}_k \hat{x}'_{k,j} + \hat{x}_{sk} + \phi_{sk+j}$$
$$\hat{r}_{sk+j} = \hat{\mathcal{Y}}_k \hat{r}'_{k,j} + \psi_{sk+j}$$

Sources of local roundoff error in s-step CG

Computing the s-step Krylov subspace basis:

$$A\underline{\hat{\mathcal{Y}}_k} = \hat{\mathcal{Y}}_k \mathcal{B}_k + \Delta\mathcal{Y}_k$$

Error in computing s-step basis

Updating coordinate vectors in the inner loop:

$$\hat{x}'_{k,j} = \hat{x}'_{k,j-1} + \hat{q}'_{k,j-1} + \xi_{k,j}$$
$$\hat{r}'_{k,j} = \hat{r}'_{k,j-1} - \mathcal{B}_k\, \hat{q}'_{k,j-1} + \eta_{k,j}$$
$$\text{with} \quad \hat{q}'_{k,j-1} = \mathrm{fl}(\hat{\alpha}_{sk+j-1}\hat{p}'_{k,j-1})$$

Recovering CG vectors for use in next outer loop:

$$\hat{x}_{sk+j} = \hat{\mathcal{Y}}_k \hat{x}'_{k,j} + \hat{x}_{sk} + \phi_{sk+j}$$
$$\hat{r}_{sk+j} = \hat{\mathcal{Y}}_k \hat{r}'_{k,j} + \psi_{sk+j}$$

Sources of local roundoff error in s-step CG

Computing the s-step Krylov subspace basis:

$$A\underline{\hat{\mathcal{Y}}_k} = \hat{\mathcal{Y}}_k \mathcal{B}_k + \Delta\mathcal{Y}_k$$

Error in computing s-step basis

Updating coordinate vectors in the inner loop:

$$\hat{x}'_{k,j} = \hat{x}'_{k,j-1} + \hat{q}'_{k,j-1} + \xi_{k,j}$$
$$\hat{r}'_{k,j} = \hat{r}'_{k,j-1} - \mathcal{B}_k\, \hat{q}'_{k,j-1} + \eta_{k,j}$$
$$\text{with} \quad \hat{q}'_{k,j-1} = \mathrm{fl}(\hat{\alpha}_{sk+j-1}\hat{p}'_{k,j-1})$$

Error in updating coordinate vectors

Recovering CG vectors for use in next outer loop:

$$\hat{x}_{sk+j} = \hat{\mathcal{Y}}_k \hat{x}'_{k,j} + \hat{x}_{sk} + \phi_{sk+j}$$
$$\hat{r}_{sk+j} = \hat{\mathcal{Y}}_k \hat{r}'_{k,j} + \psi_{sk+j}$$

Sources of local roundoff error in s-step CG

Computing the s-step Krylov subspace basis:

$$A\underline{\hat{\mathcal{Y}}_k} = \hat{\mathcal{Y}}_k \mathcal{B}_k + \Delta\mathcal{Y}_k$$

Error in computing s-step basis

Updating coordinate vectors in the inner loop:

$$\hat{x}'_{k,j} = \hat{x}'_{k,j-1} + \hat{q}'_{k,j-1} + \xi_{k,j}$$
$$\hat{r}'_{k,j} = \hat{r}'_{k,j-1} - \mathcal{B}_k\, \hat{q}'_{k,j-1} + \eta_{k,j}$$
$$\text{with} \quad \hat{q}'_{k,j-1} = \mathrm{fl}(\hat{\alpha}_{sk+j-1}\hat{p}'_{k,j-1})$$

Error in updating coordinate vectors

Recovering CG vectors for use in next outer loop:

$$\hat{x}_{sk+j} = \hat{\mathcal{Y}}_k \hat{x}'_{k,j} + \hat{x}_{sk} + \phi_{sk+j}$$
$$\hat{r}_{sk+j} = \hat{\mathcal{Y}}_k \hat{r}'_{k,j} + \psi_{sk+j}$$

Error in basis change

Attainable accuracy of s-step CG

$f_i \equiv b - A\hat{x}_i - \hat{r}_i$

For CG:

$$\|f_i\| \leq \|f_0\| + \varepsilon \sum_{m=1}^{i} (1+N)\|A\|\|\hat{x}_m\| + \|\hat{r}_m\|$$

Attainable accuracy of s-step CG

$f_i \equiv b - A\hat{x}_i - \hat{r}_i$

For CG:

$$\|f_i\| \leq \|f_0\| + \varepsilon \sum_{m=1}^{i} (1+N)\|A\|\|\hat{x}_m\| + \|\hat{r}_m\|$$

For s-step CG: $i \equiv sk + j$

$$\|f_{sk+j}\| \leq \|f_0\| + \varepsilon \mathbf{\Gamma_k} \sum_{m=1}^{sk+j} (1+N)\|A\|\|\hat{x}_m\| + \|\hat{r}_m\|$$

where

$$\Gamma_k = \max_{\ell \leq k} c \cdot \|\hat{y}_\ell^+\| \, \|\hat{y}_\ell\|$$

(see C., 2015)

where c is a low-degree polynomial in s

Attainable accuracy of s-step CG

$f_i \equiv b - A\hat{x}_i - \hat{r}_i$

For CG:

$$\|f_i\| \leq \|f_0\| + \varepsilon \sum_{m=1}^{i} (1+N)\|A\|\|\hat{x}_m\| + \|\hat{r}_m\|$$

For s-step CG: $i \equiv sk + j$

$$\|f_{sk+j}\| \leq \|f_0\| + \varepsilon \boldsymbol{\Gamma_k} \sum_{m=1}^{sk+j} (1+N)\|A\|\|\hat{x}_m\| + \|\hat{r}_m\|$$

where

$$\Gamma_k = \max_{\ell \leq k} c \cdot \|\hat{\mathcal{Y}}_\ell^+\| \, \|\hat{\mathcal{Y}}_\ell\|$$

(see C., 2015)

where c is a low-degree polynomial in s

Conditioning of computed "s-step basis" plays a huge role in determining numerical behavior!

Choosing a Polynomial Basis

- Recall: in each outer loop of s-step CG, we compute bases for some Krylov subspaces, e.g., $\mathcal{K}_{s+1}(A, p_i) = \text{span}\{p_i, Ap_i, \ldots, A^s p_i\}$

Choosing a Polynomial Basis

- Recall: in each outer loop of s-step CG, we compute bases for some Krylov subspaces, e.g., $\mathcal{K}_{s+1}(A, p_i) = \text{span}\{p_i, Ap_i, \ldots, A^s p_i\}$

- Simple loop unrolling gives monomial basis, e.g., $\mathcal{Y}_k = [p_m, Ap_m, \ldots, A^s p_m]$
  - Condition number can grow exponentially with s
  - Recognized early on that this negatively affects convergence and accuracy (Leland, 1989), (Chronopoulous & Swanson, 1995)

Choosing a Polynomial Basis

- Recall: in each outer loop of s-step CG, we compute bases for some Krylov subspaces, e.g., $\mathcal{K}_{s+1}(A, p_i) = \text{span}\{p_i, Ap_i, \ldots, A^s p_i\}$

- Simple loop unrolling gives monomial basis, e.g., $\mathcal{Y}_k = [p_m, Ap_m, \ldots, A^s p_m]$
  - Condition number can grow exponentially with s
  - Recognized early on that this negatively affects convergence and accuracy (Leland, 1989), (Chronopoulous & Swanson, 1995)

- **Improve basis condition number to improve numerical behavior**: Use different polynomials to compute a basis for the same subspace.

Choosing a Polynomial Basis

- Recall: in each outer loop of s-step CG, we compute bases for some Krylov subspaces, e.g., $\mathcal{K}_{s+1}(A, p_i) = \text{span}\{p_i, Ap_i, \ldots, A^s p_i\}$

- Simple loop unrolling gives monomial basis, e.g., $\mathcal{Y}_k = [p_m, Ap_m, \ldots, A^s p_m]$
  - Condition number can grow exponentially with s
  - Recognized early on that this negatively affects convergence and accuracy (Leland, 1989), (Chronopoulous & Swanson, 1995)

- **Improve basis condition number to improve numerical behavior**: Use different polynomials to compute a basis for the same subspace.

- Two choices based on spectral information that usually lead to well-conditioned bases:
  - **Newton polynomials**
  - **Chebyshev polynomials**

"Backwards-like" analysis of Greenbaum

- Anne Greenbaum (1989): finite precision CG with matrix A behaves like exact CG run on a larger matrix $\tilde{A}$ whose eigenvalues lie in tight clusters around the eigenvalues of A

"Backwards-like" analysis of Greenbaum

- Anne Greenbaum (1989): finite precision CG with matrix A behaves like exact CG run on a larger matrix $\tilde{A}$ whose eigenvalues lie in tight clusters around the eigenvalues of A

- Based on work of Chris Paige for finite precision Lanczos (1976, 1980):
  - Complete rounding error analysis
  - Computed eigenvalues lie between extreme eigenvalues of A to within a small multiple of machine precision
  - At least one small interval containing an eigenvalue of A is found by the Nth iteration
  - The algorithm behaves as if it used full reorthogonalization until a close eigenvalue approximation is found
  - Loss of orthogonality among basis vectors follows a rigorous pattern and implies that some eigenvalue approximation has converged

"Backwards-like" analysis of Greenbaum

- Anne Greenbaum (1989): finite precision CG with matrix A behaves like exact CG run on a larger matrix $\tilde{A}$ whose eigenvalues lie in tight clusters around the eigenvalues of A

- Based on work of Chris Paige for finite precision Lanczos (1976, 1980):
  - Complete rounding error analysis
  - Computed eigenvalues lie between extreme eigenvalues of A to within a small multiple of machine precision
  - At least one small interval containing an eigenvalue of A is found by the Nth iteration
  - The algorithm behaves as if it used full reorthogonalization until a close eigenvalue approximation is found
  - Loss of orthogonality among basis vectors follows a rigorous pattern and implies that some eigenvalue approximation has converged

- Can we make similar statements for s-step variants?

Roundoff Error in Lanczos vs. s-step Lanczos

Finite precision Lanczos process: (A is $N \times N$ with at most n nonzeros per row)

$$A\hat{V}_m = \hat{V}_m \hat{T}_m + \hat{\beta}_{m+1}\hat{v}_{m+1}e_m^T + \delta\hat{V}_m$$

$$\hat{V}_m = [\hat{v}_1, \ldots, \hat{v}_m], \qquad \delta\hat{V}_m = [\delta\hat{v}_1, \ldots, \delta\hat{v}_m], \qquad \hat{T}_m = \begin{bmatrix} \hat{\alpha}_1 & \hat{\beta}_2 & & \\ \hat{\beta}_2 & \ddots & \ddots & \\ & \ddots & \ddots & \hat{\beta}_m \\ & & \hat{\beta}_m & \hat{\alpha}_m \end{bmatrix}$$

for $i \in \{1, \ldots, m\}$,

$$\begin{aligned} \|\delta\hat{v}_i\|_2 &\leq \varepsilon_1 \sigma \\ \hat{\beta}_{i+1}\left|\hat{v}_i^T \hat{v}_{i+1}\right| &\leq 2\varepsilon_0 \sigma \\ \left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right| &\leq \varepsilon_0/2 \\ \left|\hat{\beta}_{i+1}^2 + \hat{\alpha}_i^2 + \hat{\beta}_i^2 - \|A\hat{v}_i\|_2^2\right| &\leq 4i(3\varepsilon_0 + \varepsilon_1)\sigma^2 \end{aligned}$$

$$\sigma \equiv \|A\|_2$$
$$\theta\sigma \equiv \||A|\|_2$$

Lanczos [Paige, 1976]

$$\varepsilon_0 = O(\varepsilon N)$$
$$\varepsilon_1 = O(\varepsilon n \theta)$$

Roundoff Error in Lanczos vs. s-step Lanczos

Finite precision Lanczos process: (A is $N \times N$ with at most n nonzeros per row)

$$A\hat{V}_m = \hat{V}_m \hat{T}_m + \hat{\beta}_{m+1}\hat{v}_{m+1}e_m^T + \delta\hat{V}_m$$

$$\hat{V}_m = [\hat{v}_1, \ldots, \hat{v}_m], \qquad \delta\hat{V}_m = [\delta\hat{v}_1, \ldots, \delta\hat{v}_m], \qquad \hat{T}_m = \begin{bmatrix} \hat{\alpha}_1 & \hat{\beta}_2 & & \\ \hat{\beta}_2 & \ddots & \ddots & \\ & \ddots & \ddots & \hat{\beta}_m \\ & & \hat{\beta}_m & \hat{\alpha}_m \end{bmatrix}$$

for $i \in \{1, \ldots, m\}$,

$$\|\delta\hat{v}_i\|_2 \le \varepsilon_1 \sigma$$
$$\hat{\beta}_{i+1}\left|\hat{v}_i^T \hat{v}_{i+1}\right| \le 2\varepsilon_0 \sigma$$
$$\left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right| \le \varepsilon_0/2$$
$$\left|\hat{\beta}_{i+1}^2 + \hat{\alpha}_i^2 + \hat{\beta}_i^2 - \|A\hat{v}_i\|_2^2\right| \le 4i(3\varepsilon_0 + \varepsilon_1)\sigma^2$$

$$\sigma \equiv \|A\|_2$$
$$\theta\sigma \equiv \||A|\|_2$$

Lanczos [Paige, 1976]

$$\varepsilon_0 = O(\varepsilon N)$$
$$\varepsilon_1 = O(\varepsilon n \theta)$$

s-step Lanczos [C., Demmel, 2015]:

$$\varepsilon_0 = O(\varepsilon N \mathbf{\Gamma^2})$$
$$\varepsilon_1 = O(\varepsilon n \theta \mathbf{\Gamma})$$

$$\Gamma = c \cdot \max_{\ell} \|\hat{y}_\ell^+\| \, \|\hat{y}_\ell\|$$

Convergence of Ritz Values in s-step Lanczos

- All results of Paige [1980], e.g., loss of orthogonality → eigenvalue convergence, hold for s-step Lanczos as long as

$$\left(\Gamma = c \cdot \max_{\ell} \|\hat{y}_\ell^+\| \, \|\hat{y}_\ell\|\right)$$

$$\Gamma \leq \left(24\varepsilon(N + 11s + 15)\right)^{-1/2} \approx \frac{1}{\sqrt{N\varepsilon}}$$

Convergence of Ritz Values in s-step Lanczos

- All results of Paige [1980], e.g., loss of orthogonality → eigenvalue convergence, hold for s-step Lanczos as long as

$$\left(\Gamma = c \cdot \max_{\ell} \|\hat{y}_{\ell}^{+}\| \, \|\hat{y}_{\ell}\|\right)$$

$$\Gamma \leq \left(24\varepsilon(N + 11s + 15)\right)^{-1/2} \approx \frac{1}{\sqrt{N\varepsilon}}$$

- Bounds on accuracy of Ritz values depend on Γ^2

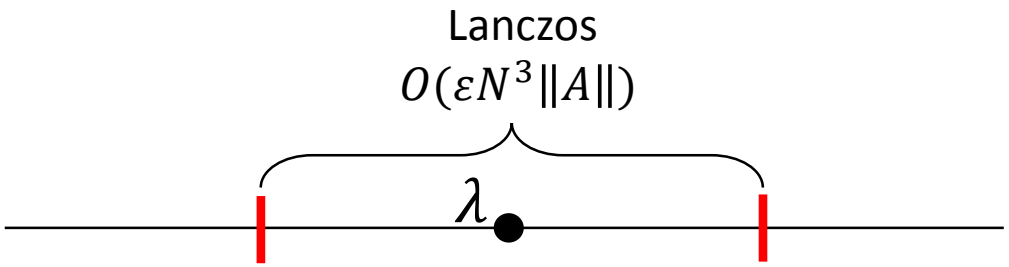
Convergence of Ritz Values in s-step Lanczos

- All results of Paige [1980], e.g., loss of orthogonality → eigenvalue convergence, hold for s-step Lanczos as long as

$$\left(\Gamma = c \cdot \max_{\ell} \|\hat{y}_{\ell}^{+}\| \, \|\hat{y}_{\ell}\|\right)$$

$$\Gamma \leq \left(24\varepsilon(N + 11s + 15)\right)^{-1/2} \approx \frac{1}{\sqrt{N\varepsilon}}$$

- Bounds on accuracy of Ritz values depend on Γ^2



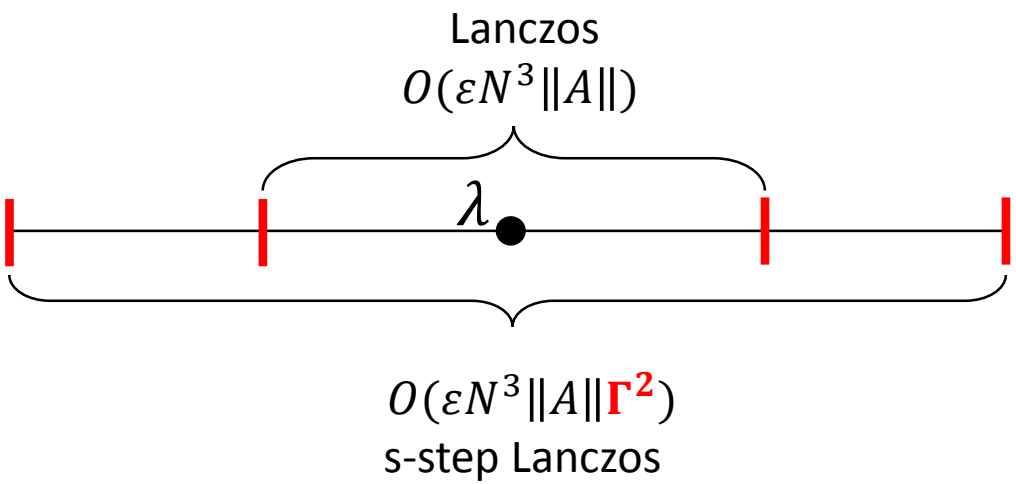
Convergence of Ritz Values in s-step Lanczos

- All results of Paige [1980], e.g., loss of orthogonality → eigenvalue convergence, hold for s-step Lanczos as long as

$$\Gamma \leq \left(24\varepsilon(N+11s+15)\right)^{-1/2} \approx \frac{1}{\sqrt{N\varepsilon}}$$

$$\left(\Gamma = c \cdot \max_{\ell} \|\hat{\mathcal{Y}}_{\ell}^{+}\| \, \|\hat{\mathcal{Y}}_{\ell}\|\right)$$

- Bounds on accuracy of Ritz values depend on Γ^2



Convergence of Ritz Values in s-step Lanczos

- All results of Paige [1980], e.g., loss of orthogonality → eigenvalue convergence, hold for s-step Lanczos as long as

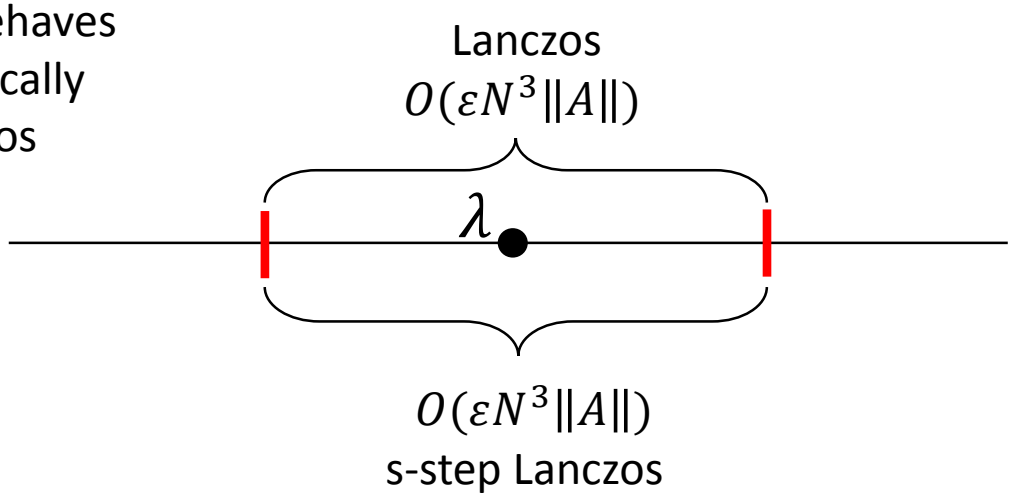
$$\left(\Gamma = c \cdot \max_{\ell} \|\hat{y}_{\ell}^{+}\| \, \|\hat{y}_{\ell}\|\right)$$

$$\Gamma \leq \left(24\varepsilon(N + 11s + 15)\right)^{-1/2} \approx \frac{1}{\sqrt{N\varepsilon}}$$

- Bounds on accuracy of Ritz values depend on Γ^2

If $\boldsymbol{\Gamma \approx 1}$:

s-step Lanczos behaves the same numerically as classical Lanczos



Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

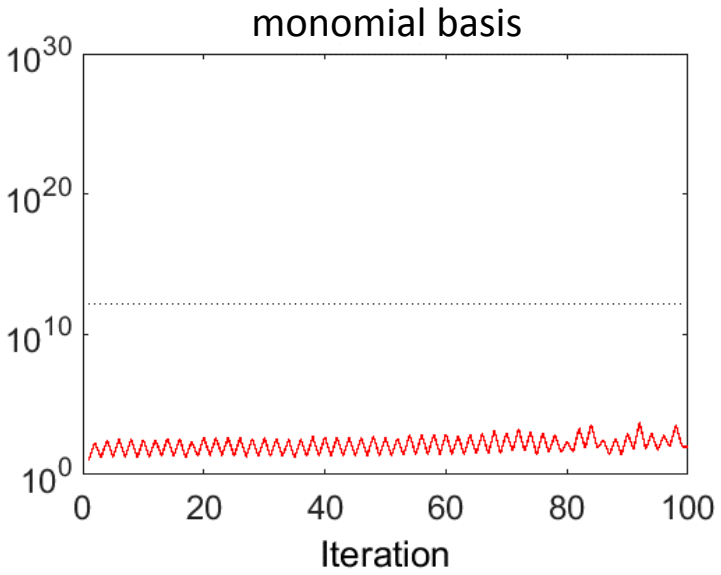
Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

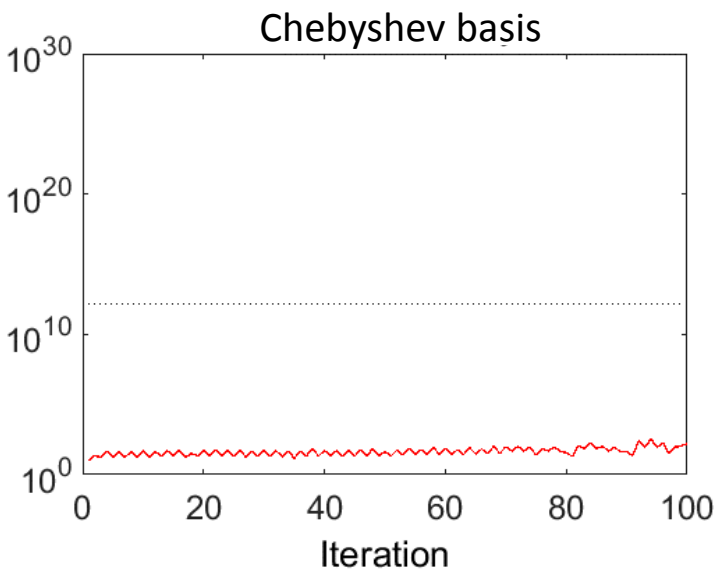
$\boldsymbol{s = 2}$

Top plots:

Computed $\Gamma_{k,j}^2$

$(24(\varepsilon(n + 11s + 15))^{-1}$





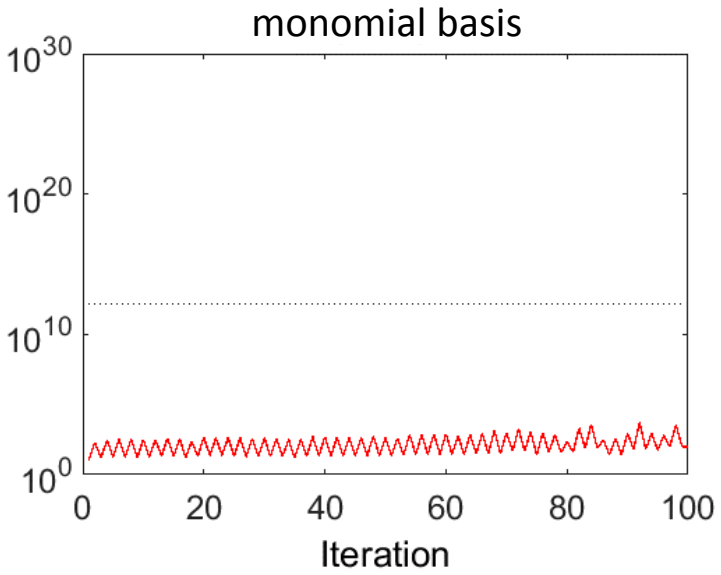
Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

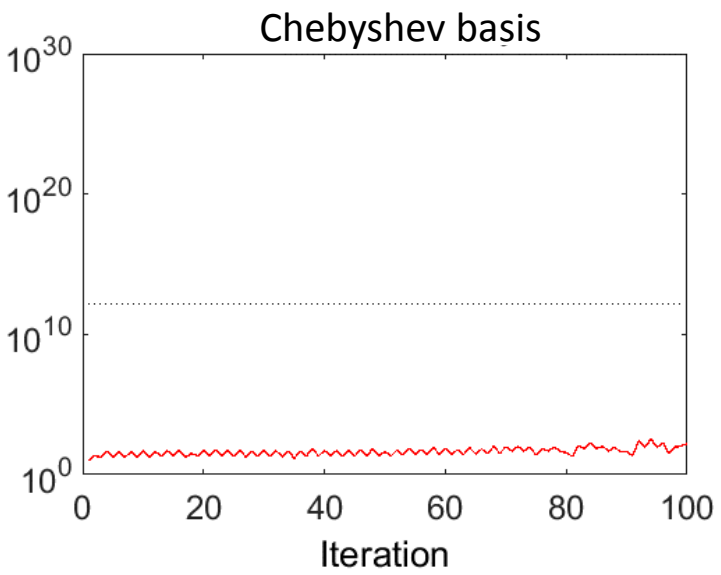
$\boldsymbol{s = 2}$

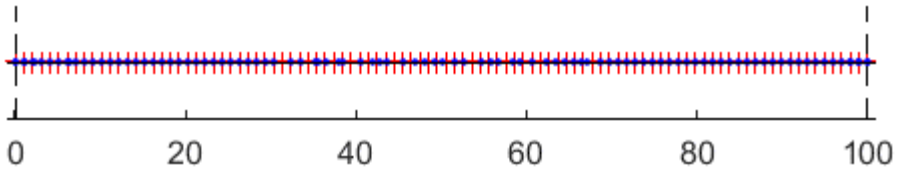
Top plots:

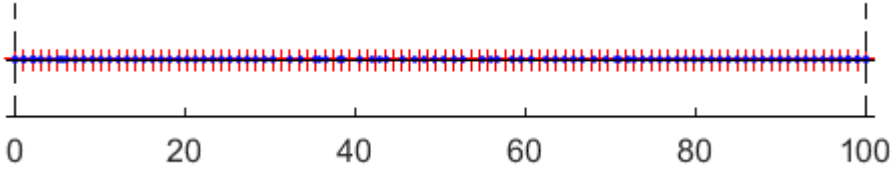
Computed $\Gamma_{k,j}^2$

$(24(\varepsilon(n + 11s + 15))^{-1}$









Bottom Plots:

True eigenvalues

Computed Ritz values

Bounds on range of computed Ritz values

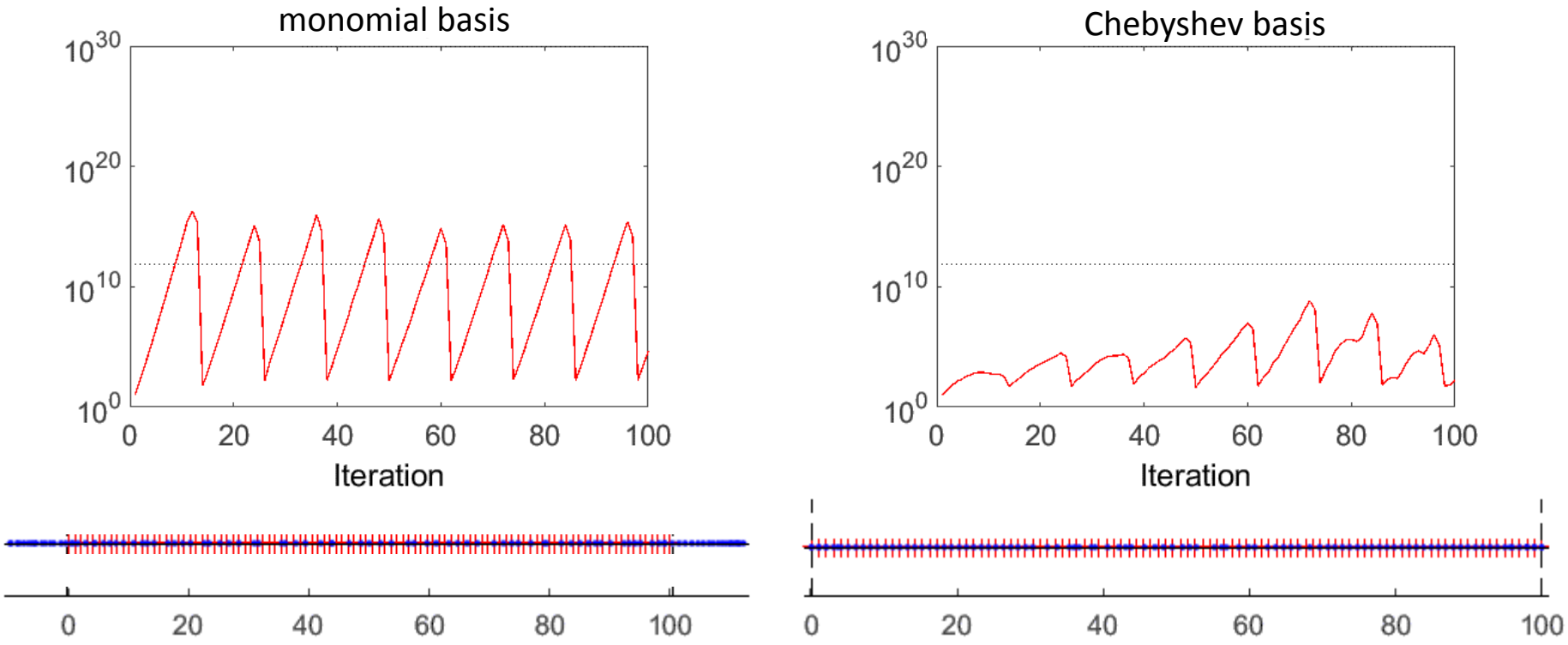
Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$s = 12$

Top plots:

Computed $\Gamma_{k,j}^2$

$(24(\varepsilon(n + 11s + 15))^{-1}$



Bottom Plots:

True eigenvalues

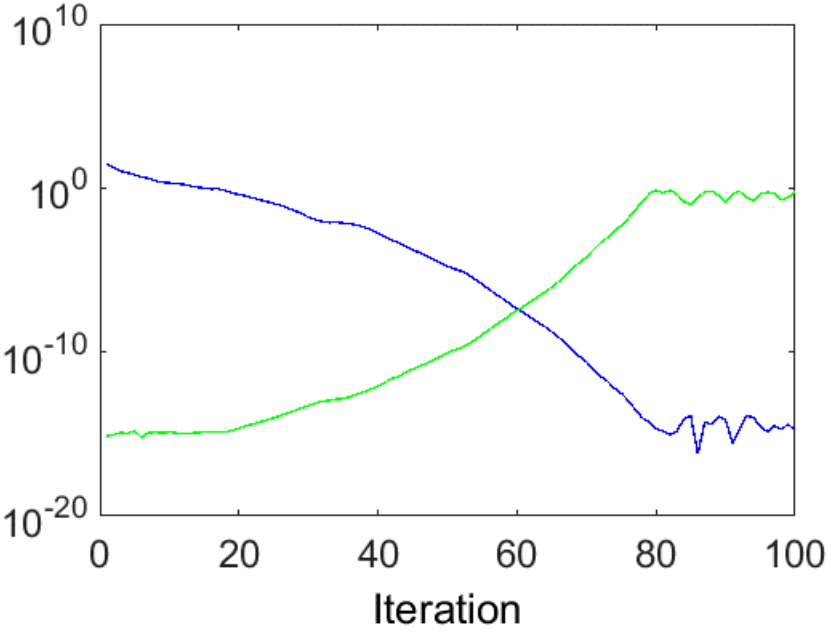
Computed Ritz values

Bounds on range of computed Ritz values

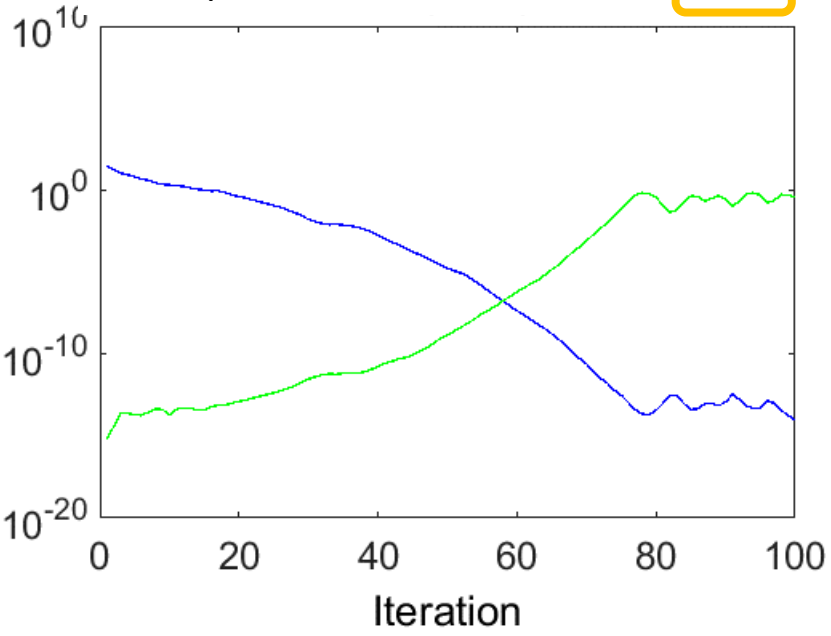
Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$\Gamma \leq 7 \times 10^2$

classical Lanczos



s-step Lanczos, monomial basis, $s = 2$



Measure of Ritz value convergence

$\max_i |z_i^{(m)T} \hat{v}_{m+1}|$

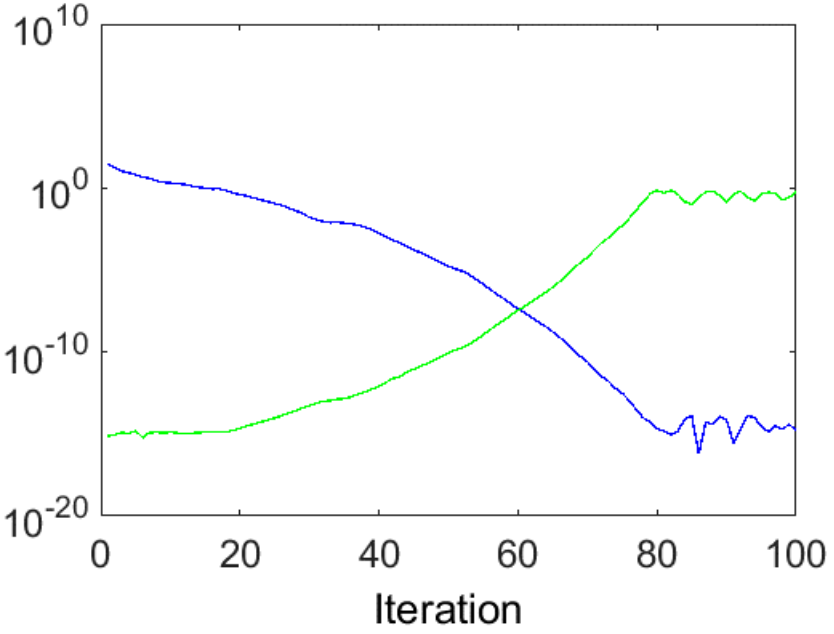
$\min_i \hat{\beta}_{m+1} \eta_{m,i}^{(m)}$

Measure of loss of orthogonality

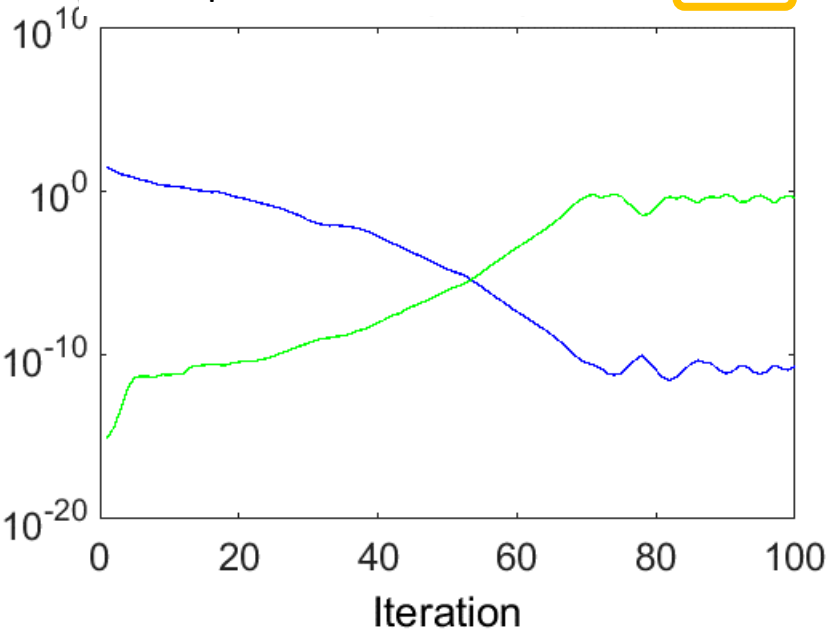
Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$\Gamma \leq 3 \times 10^3$

classical Lanczos



s-step Lanczos, monomial basis, $s = 4$



Measure of Ritz value convergence

$\max_i |z_i^{(m)T} \hat{v}_{m+1}|$

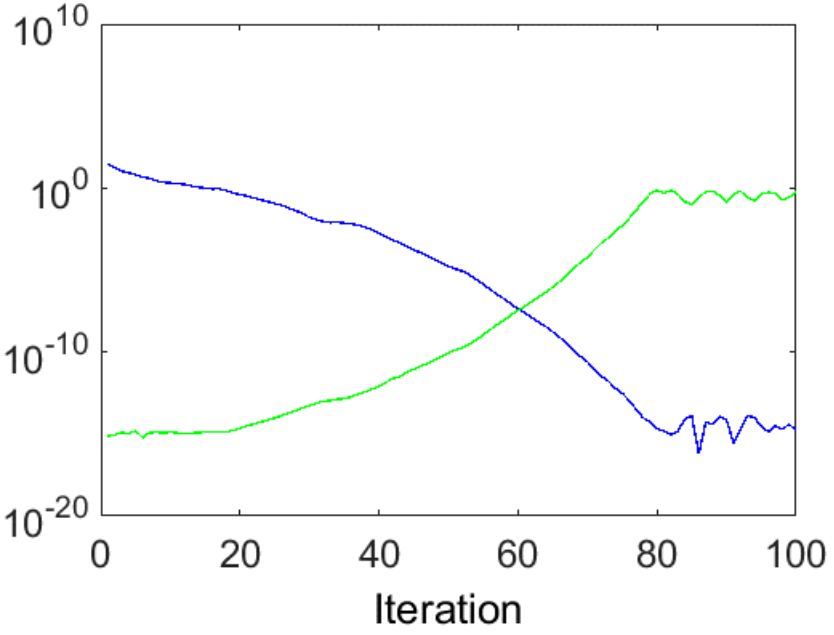
$\min_i \hat{\beta}_{m+1} \eta_{m,i}^{(m)}$

Measure of loss of orthogonality

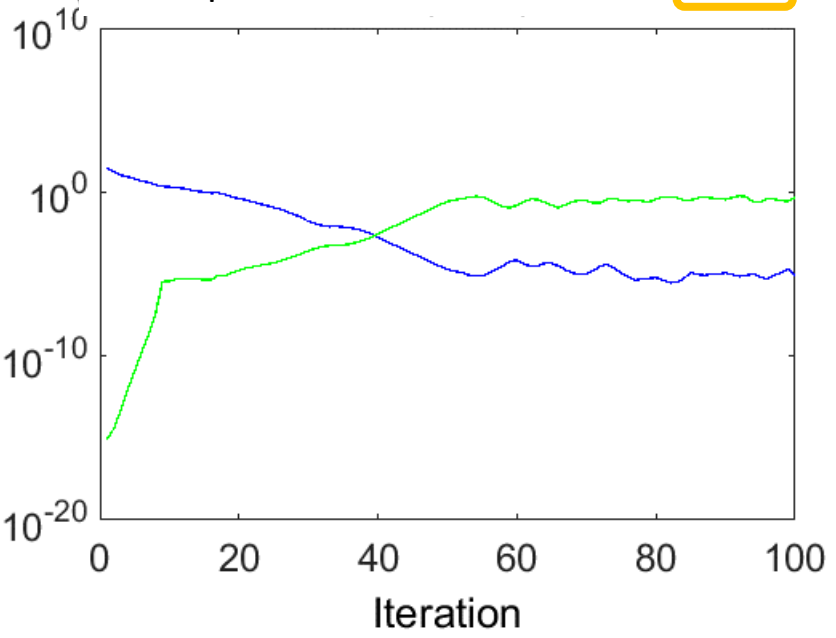
Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$\Gamma \leq 2 \times 10^6$

classical Lanczos



s-step Lanczos, monomial basis, $s = 8$



Measure of Ritz value convergence

$\max_i |z_i^{(m)T} \hat{v}_{m+1}|$

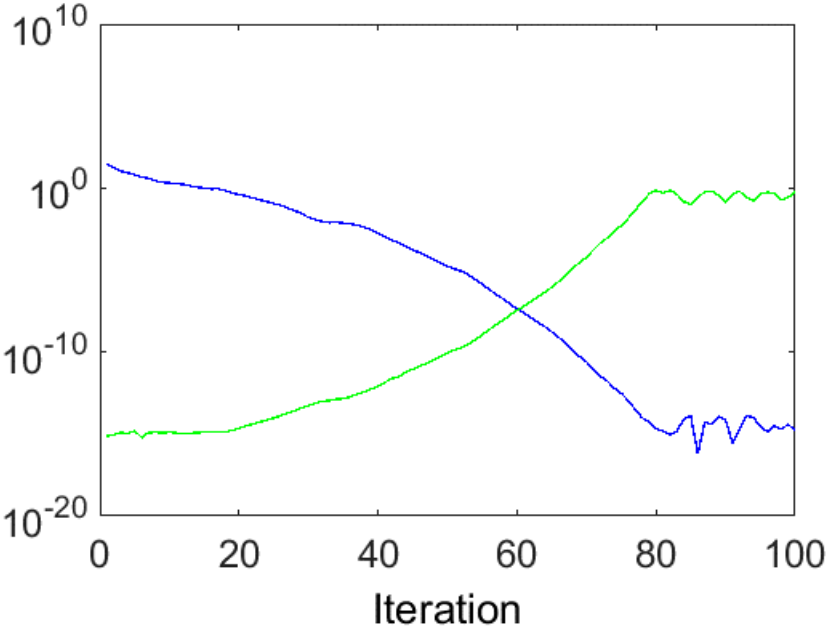
$\min_i \hat{\beta}_{m+1} \eta_{m,i}^{(m)}$

Measure of loss of orthogonality

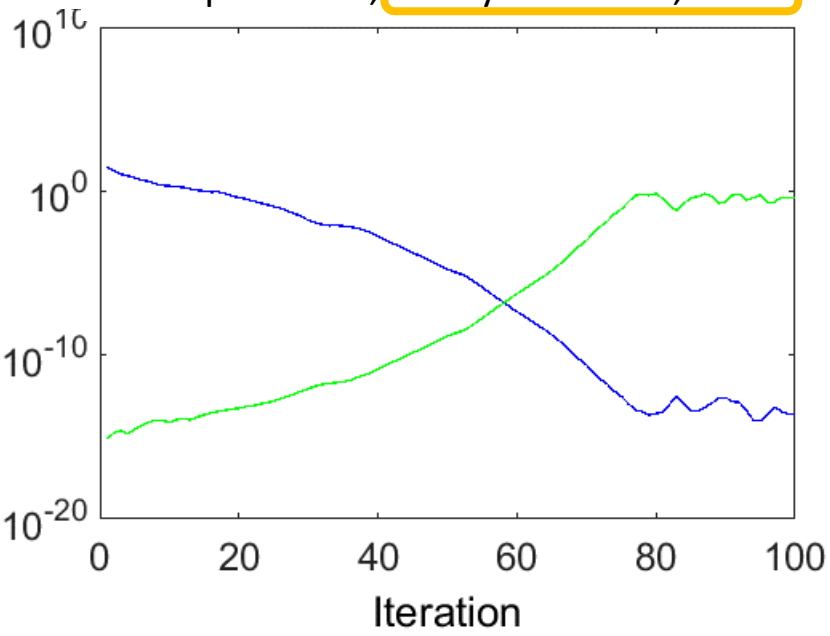
Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$\Gamma \leq 2 \times 10^3$

classical Lanczos



s-step Lanczos, Chebyshev basis, $s = 8$



Measure of Ritz value convergence

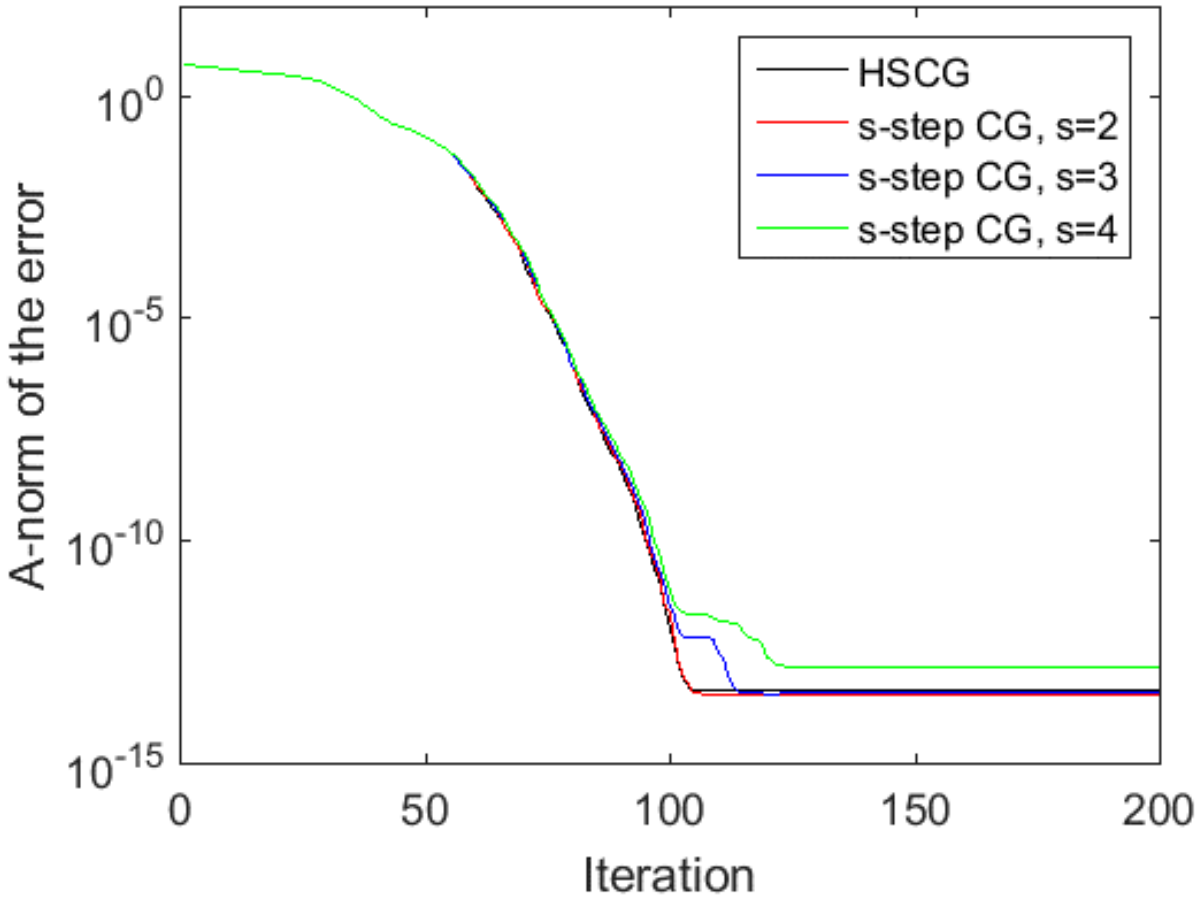
$\max_i |z_i^{(m)T} \hat{v}_{m+1}|$

$\min_i \hat{\beta}_{m+1} \eta_{m,i}^{(m)}$

Measure of loss of orthogonality

A different problem...

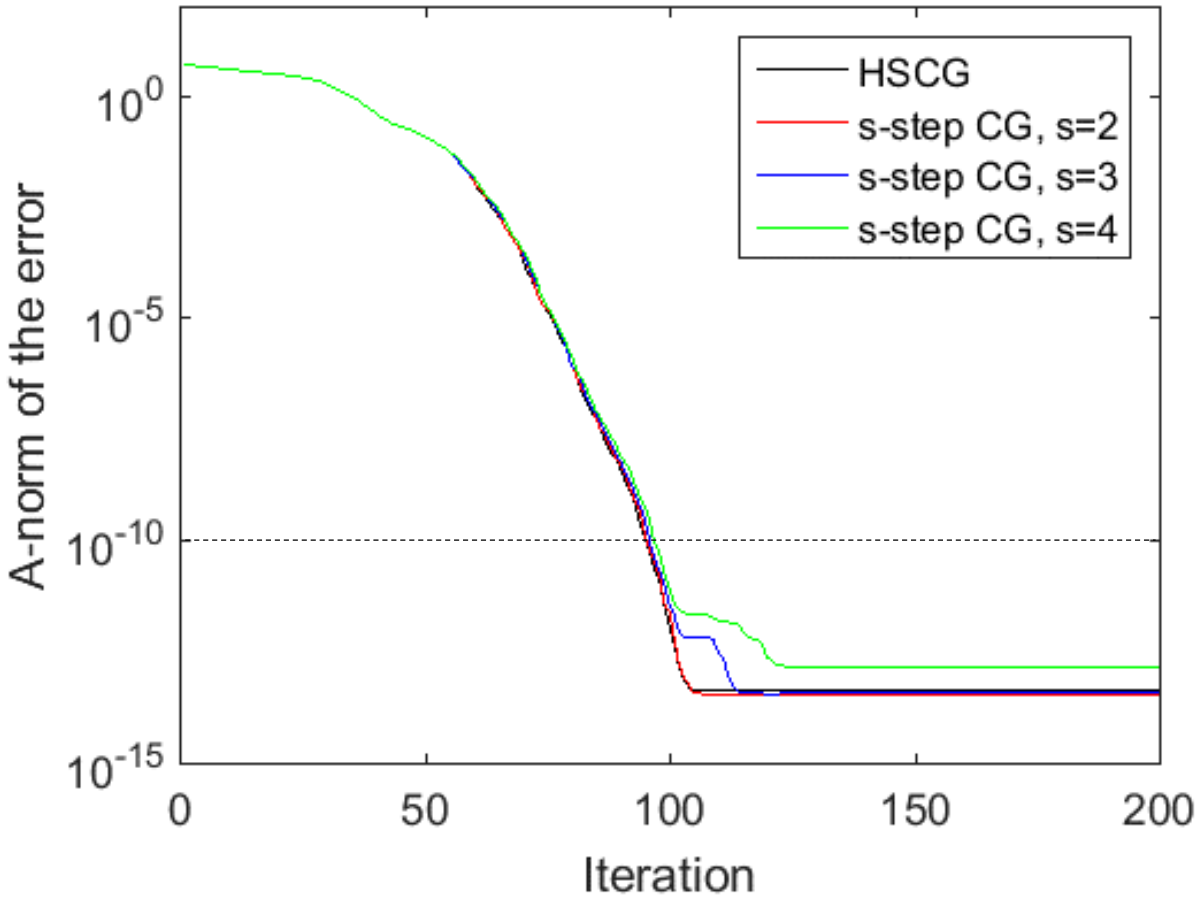
A: **nos4** from UFSMC,
b: equal components in the eigenbasis
of A and $\|b\| = 1$
$N = 100, \kappa(A) \approx 2\text{e}3$



A different problem...

A: **nos4** from UFSMC,
b: equal components in the eigenbasis of A and $\|b\| = 1$
$N = 100, \kappa(A) \approx 2\text{e}3$

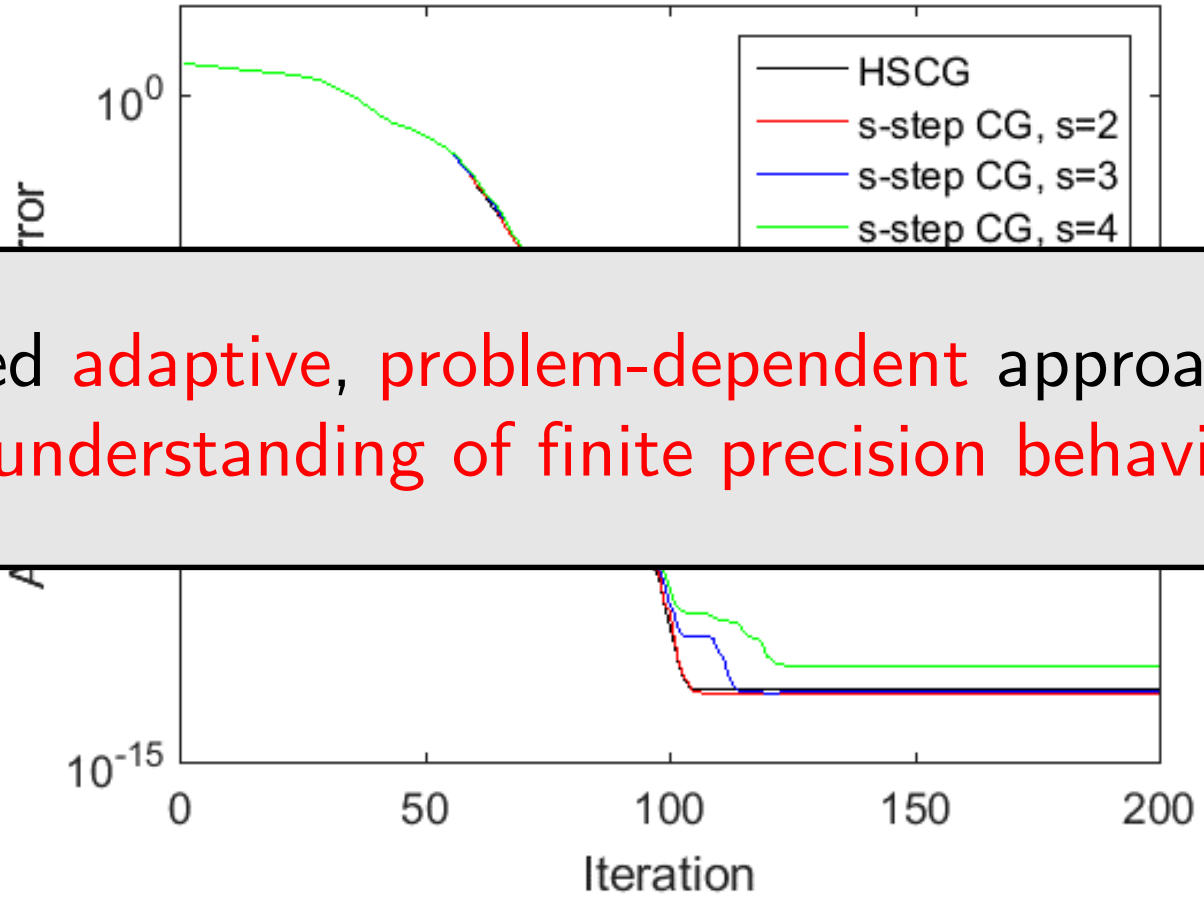
If application only requires
$\|x - x_i\|_A \approx 10^{-10}$,
finite precision effects negligible relative to classical method!



A different problem...

A: **nos4** from UFSMC,
b: equal components in the eigenbasis of A and $\|b\| = 1$
$N = 100, \kappa(A) \approx 2\text{e}3$

If application only requires $\|x - x_i\|_A \approx 10^{-10}$, finite precision effects negligible relative to classical method!



Need adaptive, problem-dependent approach based on understanding of finite precision behavior!

Adaptive s-step CG

- Consider the growth of the relative residual gap caused by errors in outer loop k, which begins with global iteration number m

Adaptive s-step CG

- Consider the growth of the relative residual gap caused by errors in outer loop k, which begins with global iteration number m
- We can approximate an upper bound on this quantity by

$$\frac{\|f_{m+s} - f_m\|}{\|A\|\|x\|} \lesssim \varepsilon \left(1 + \kappa(A)\Gamma_k \frac{\max\limits_{j\in\{0,\ldots,s\}} \|\hat{r}_{m+j}\|}{\|A\|\|x\|} \right) \qquad f_i \equiv b - A\hat{x}_i - \hat{r}_i$$

Adaptive s-step CG

- Consider the growth of the relative residual gap caused by errors in outer loop k, which begins with global iteration number m
- We can approximate an upper bound on this quantity by

$$\frac{\|f_{m+s} - f_m\|}{\|A\|\|x\|} \lesssim \varepsilon \left(1 + \kappa(A)\Gamma_k \frac{\max_{j\in\{0,\ldots,s\}} \|\hat{r}_{m+j}\|}{\|A\|\|x\|} \right) \qquad f_i \equiv b - A\hat{x}_i - \hat{r}_i$$

- If our application requires relative accuracy ε^*, we must have

$$\Gamma_k \equiv c \cdot \|\hat{y}_k^+\| \, \||\hat{y}_k|\| \lesssim \frac{\varepsilon^*}{\varepsilon \max_{j\in\{0,\ldots,s\}} \|\hat{r}_{m+j}\|}$$

Adaptive s-step CG

- Consider the growth of the relative residual gap caused by errors in outer loop k, which begins with global iteration number m
- We can approximate an upper bound on this quantity by

$$\frac{\|f_{m+s} - f_m\|}{\|A\|\|x\|} \lesssim \varepsilon \left(1 + \kappa(A)\Gamma_k \frac{\max_{j\in\{0,\ldots,s\}} \|\hat{r}_{m+j}\|}{\|A\|\|x\|} \right) \qquad f_i \equiv b - A\hat{x}_i - \hat{r}_i$$

- If our application requires relative accuracy ε^*, we must have

$$\Gamma_k \equiv c \cdot \|\hat{y}_k^+\| \, \||\hat{y}_k|\| \lesssim \frac{\varepsilon^*}{\varepsilon \max_{j\in\{0,\ldots,s\}} \|\hat{r}_{m+j}\|}$$

- $\|\hat{r}_i\|$ large → Γ_k must be small; $\|\hat{r}_i\|$ small → Γ_k can grow

Adaptive s-step CG

- Consider the growth of the relative residual gap caused by errors in outer loop k, which begins with global iteration number m
- We can approximate an upper bound on this quantity by

$$\frac{\|f_{m+s} - f_m\|}{\|A\|\|x\|} \lesssim \varepsilon \left(1 + \kappa(A) \Gamma_k \frac{\max_{j \in \{0,\ldots,s\}} \|\hat{r}_{m+j}\|}{\|A\|\|x\|} \right) \qquad f_i \equiv b - A\hat{x}_i - \hat{r}_i$$

- If our application requires relative accuracy ε^*, we must have

$$\Gamma_k \equiv c \cdot \|\hat{y}_k^+\| \, \||\hat{y}_k|\| \lesssim \frac{\varepsilon^*}{\varepsilon \max_{j \in \{0,\ldots,s\}} \|\hat{r}_{m+j}\|}$$

- $\|\hat{r}_i\|$ large → Γ_k must be small; $\|\hat{r}_i\|$ small → Γ_k can grow

⇒ adaptive s-step approach [C., 2018]

- s starts off small, increases at rate depending on $\|\hat{r}_i\|$ and ε^*

Improving Adaptive s-step CG

- Method of Meurant and Tichý (2018) for cheap approximation of extremal Ritz values
  - Uses Cholesky factors of Lanczos tridiagonal T_i, $T_i = L_i L_i^T$
  - Use α and β computed during each iteration to incrementally update estimates of $\|L_i\|_2^2 = \lambda_{max}(T_i) \approx \lambda_{max}(A)$, $\left\|L_i^{-1}\right\|_2^{-2} = \lambda_{min}(T_i) \approx \lambda_{min}(A)$
    - Essentially no extra work, no extra communication

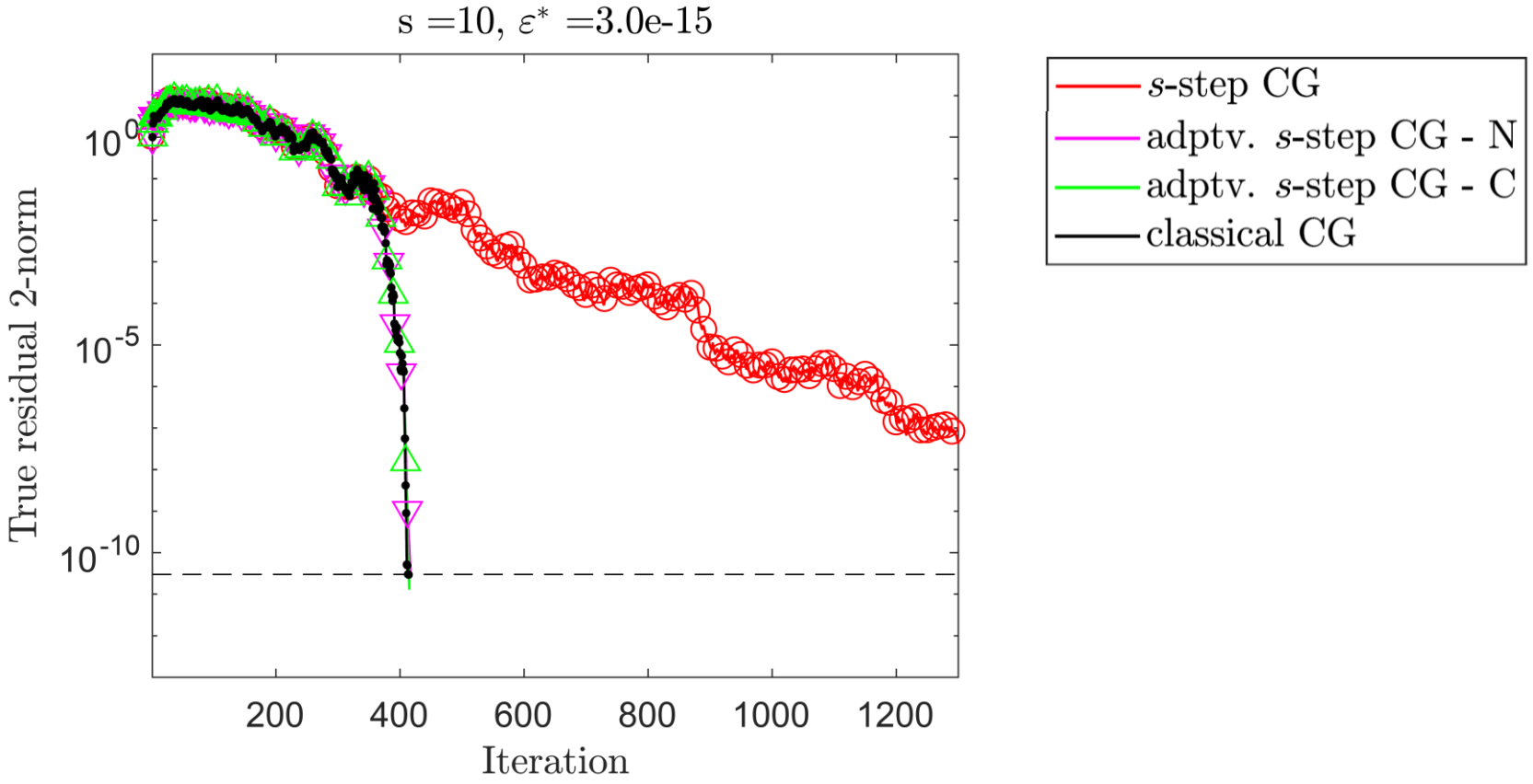
Improving Adaptive s-step CG

- Method of Meurant and Tichý (2018) for cheap approximation of extremal Ritz values
  - Uses Cholesky factors of Lanczos tridiagonal T_i, $T_i = L_i L_i^T$
  - Use α and β computed during each iteration to incrementally update estimates of $\|L_i\|_2^2 = \lambda_{max}(T_i) \approx \lambda_{max}(A)$, $\left\|L_i^{-1}\right\|_2^{-2} = \lambda_{min}(T_i) \approx \lambda_{min}(A)$
    - Essentially no extra work, no extra communication

- Can be used in two ways in adaptive algorithm
  1. Incrementally refine estimate of $\kappa(A)$ (used in determining which s to use)
  2. Incrementally refine parameters used to construct Newton or Chebyshev polynomials

A = 494bus from SuiteSparse

$b_{\mathrm{i}} = 1/\sqrt{N}$



Number of global synchronizations

Fixed s-step	Improved adaptive s-step w/Newton	Improved adaptive s-step w/Chebyshev	classical CG
-	**59**	**53**	414

Summary

- In order to truly claim that a modified variant of a Krylov subspace method is suitable for HPC/more efficient than the classical approach, we must **understand its behavior in finite precision**

- In s-step variants of Krylov subspace methods, local roundoff errors are amplified by a factor related to the conditioning of the computed "s-step bases"
  - Bounds on maximum attainable accuracy
  - Working towards understanding convergence delay

- Understanding finite precision behavior can allow us to develop adaptive approaches that are both accurate and efficient

Thank you!

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