High-Performance Variants of Krylov Subspace Methods: II/II

Erin C. Carson

Katedra numerické matematiky, Matematicko-fyzikální fakulta, Univerzita Karlova

SNA '19

January 21-25, 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

Review

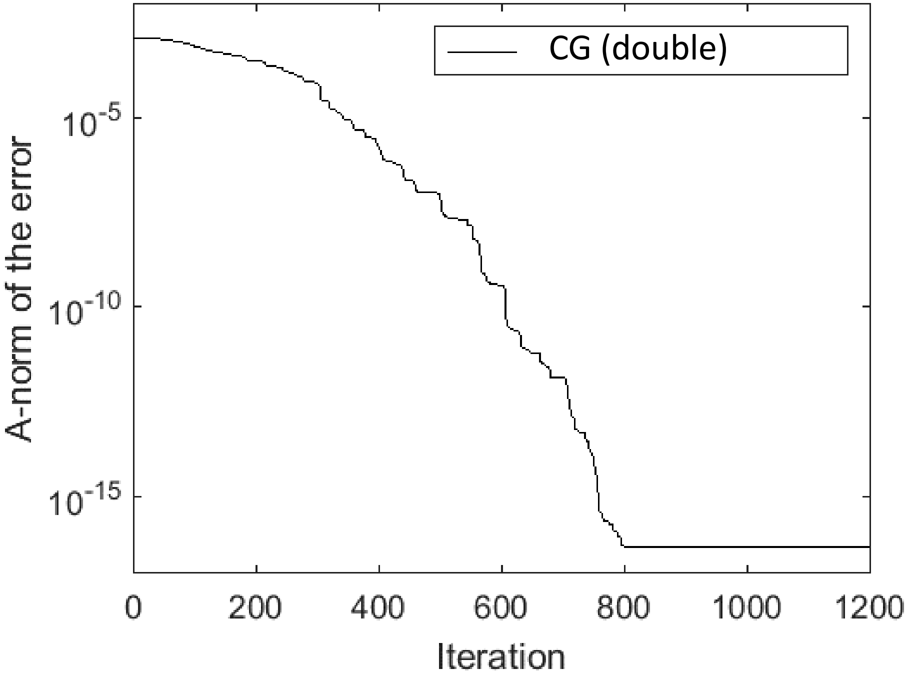
- Cost of data movement (relative to low computational cost) causes bottlenecks in classical formulations of Krylov subspace methods
- Motivates various approaches
    - Pipelined Krylov subspace methods
        - Add auxiliary recurrences to enable decoupling of inner products and SpMVs; can then be overlapped using non-blocking MPI
        - Effectively hides the cost of synchronization in each iteration
    - s-step Krylov subspace methods
        - Block iterations in groups of s; use block computation of O(s) basis vectors and block orthogonalization
        - Increases temporal locality, allowing asymptotic reduction in number of messages per iteration
    - Many practical implementation details: choosing parameters, preconditioning, etc.
- For certain (e.g., latency-bound) problems, these approaches can reduce the time-per-iteration cost

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!



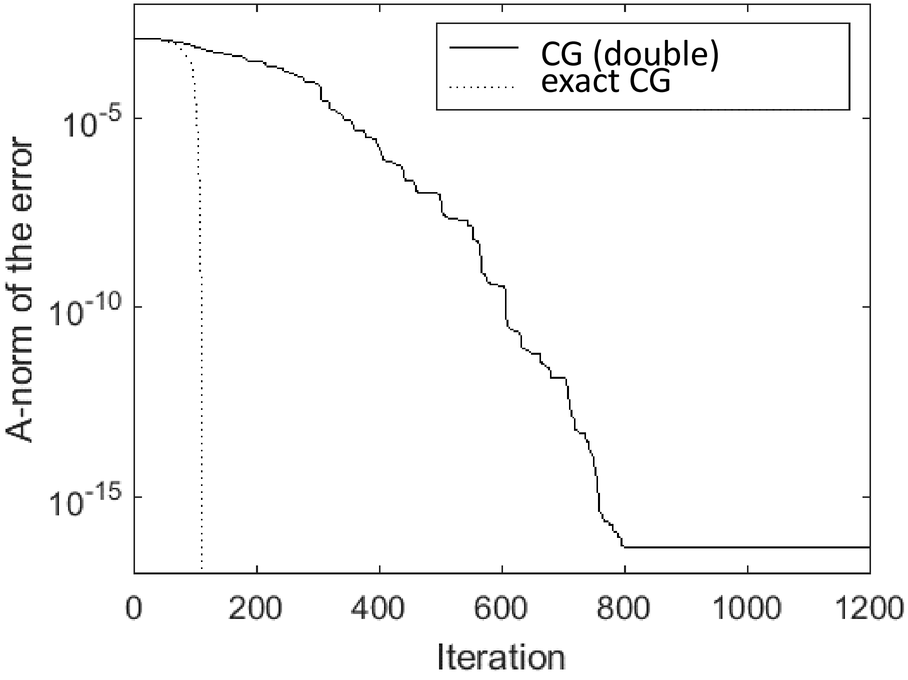
A: bcsstk03 from SuiteSparse,
b: equal components in the eigenbasis of A,$||b|| = 1$
$N = 112, \kappa(A) \approx 7\text{e}6$

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!



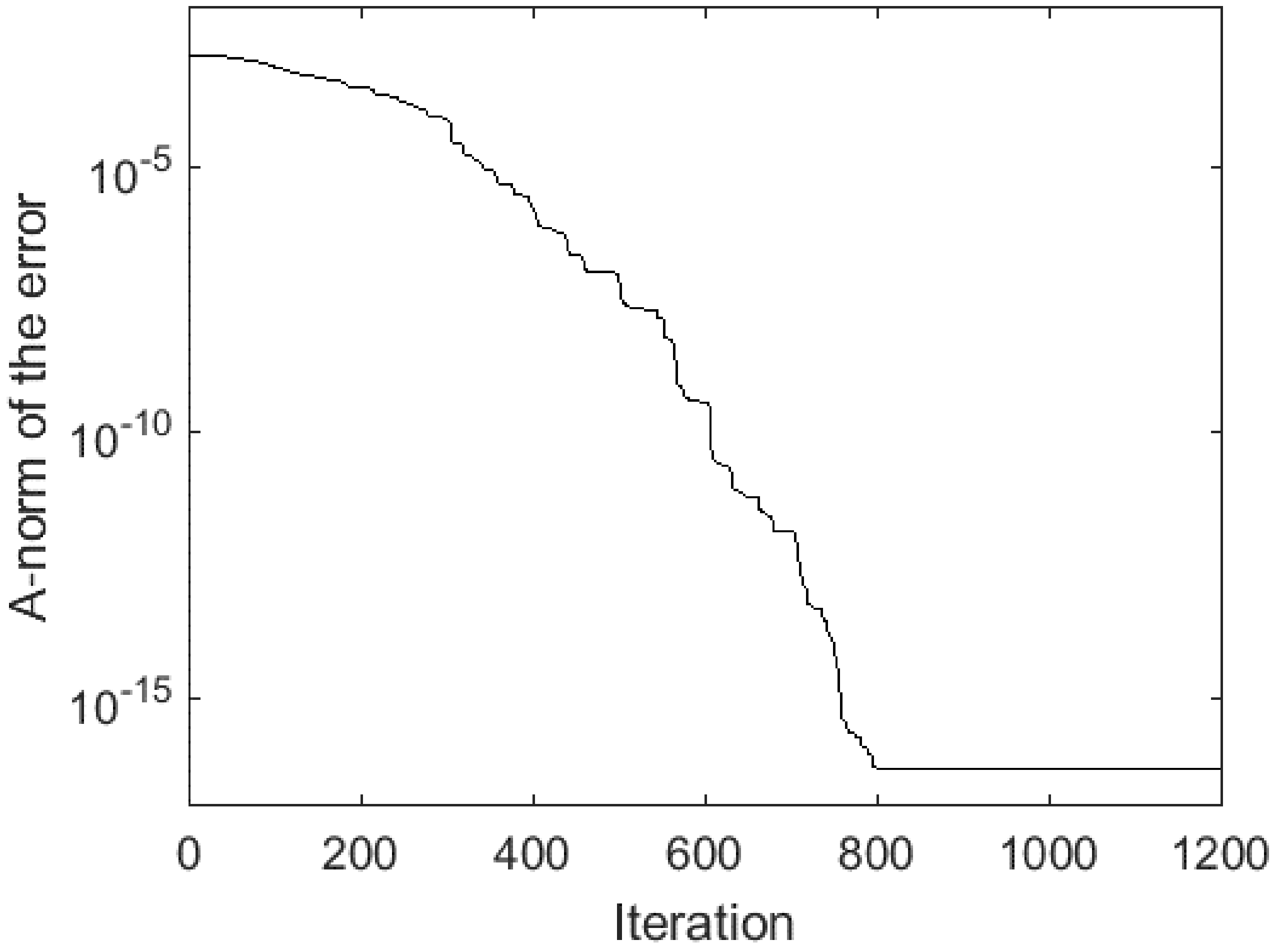
A: bcsstk03 from SuiteSparse,
b: equal components in the eigenbasis of A,$\|b\| = 1$
$N = 112, \kappa(A) \approx 7e6$

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

Conjugate Gradient method for solving Ax = b
double precision ($\varepsilon = 2^{-53}$)

$$\|x_i - x\|_A = \sqrt{(x_i - x)^T A (x_i - x)}$$

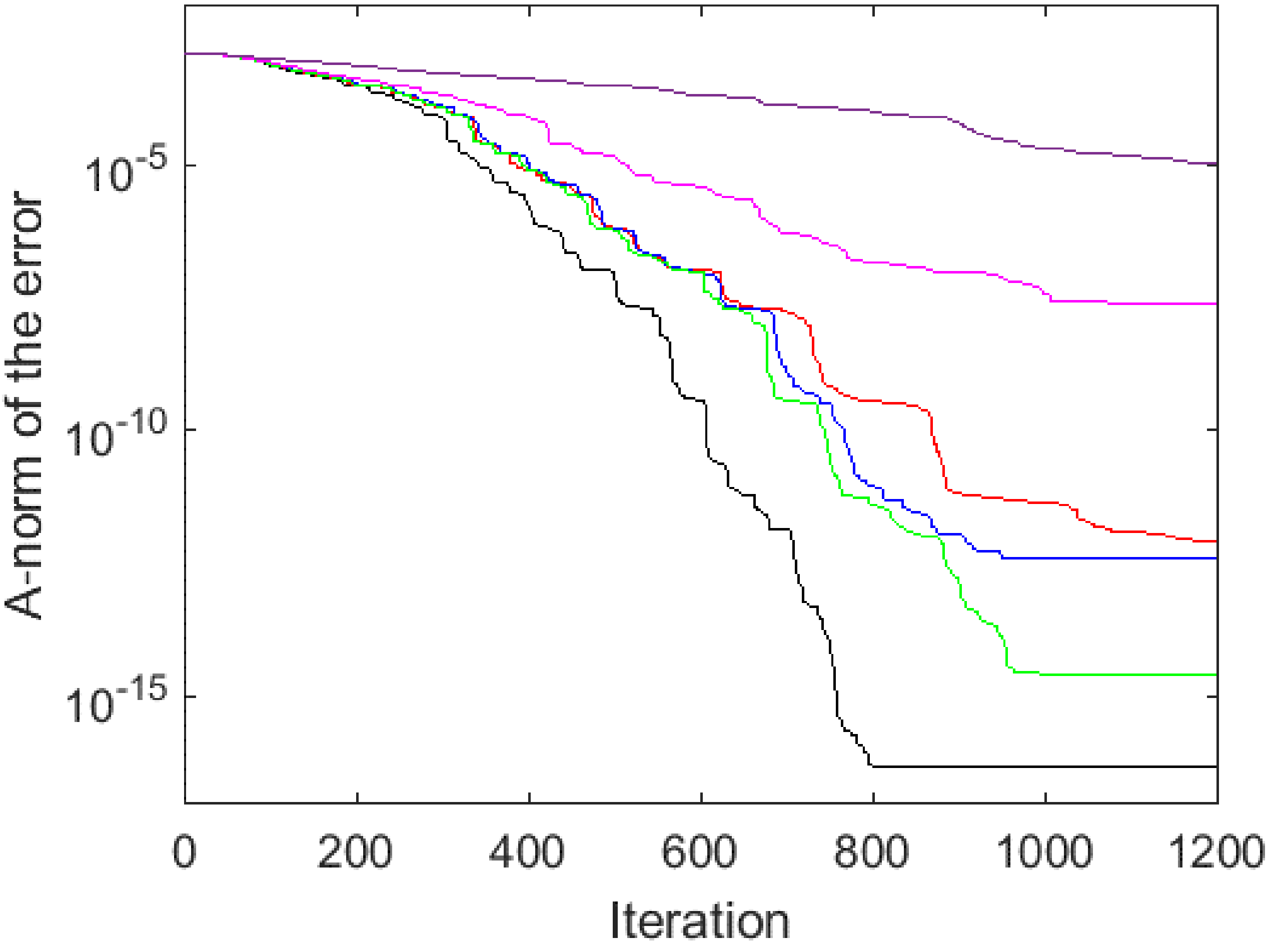
$$\boxed{\begin{aligned} x_i &= x_{i-1} + \alpha_i p_i \\ r_i &= r_{i-1} - \alpha_i A p_i \\ p_i &= r_i + \beta_i p_i \end{aligned}}$$



Conjugate Gradient method for solving Ax = b
double precision ($\varepsilon = 2^{-53}$)

$$\|x_i - x\|_A = \sqrt{(x_i - x)^T A (x_i - x)}$$

$$\boxed{\begin{aligned} x_i &= x_{i-1} + \alpha_i p_i \\ r_i &= r_{i-1} - \alpha_i A p_i \\ p_i &= r_i + \beta_i p_i \end{aligned}}$$



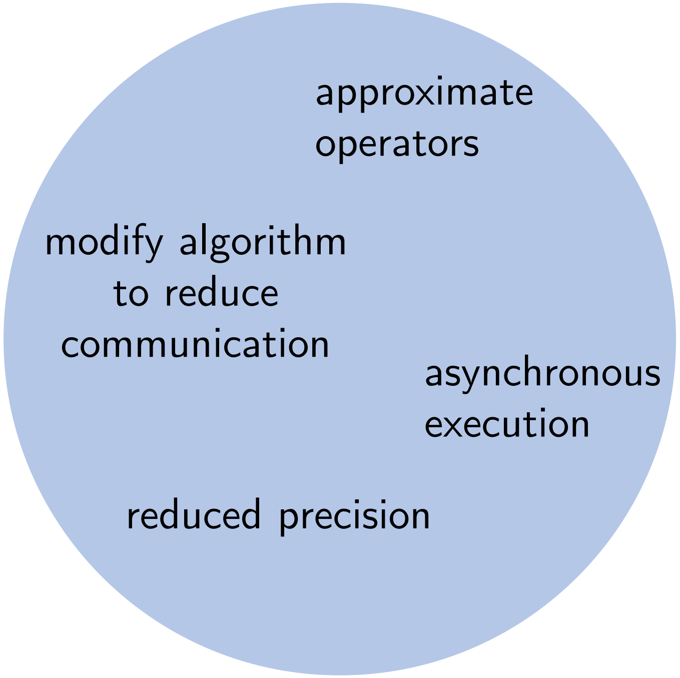
Improving Performance of Iterative Solvers

$$\text{runtime} = \begin{pmatrix}\text{time per}\\ \text{iteration}\end{pmatrix} \times \begin{pmatrix}\text{number of iterations}\\ \text{until convergence}\end{pmatrix}$$

Improving Performance of Iterative Solvers

$$\text{runtime} = \begin{pmatrix}\text{time per}\\ \text{iteration}\end{pmatrix} \times \begin{pmatrix}\text{number of iterations}\\ \text{until convergence}\end{pmatrix}$$

Reduce time per iteration



approximate operators

modify algorithm to reduce communication

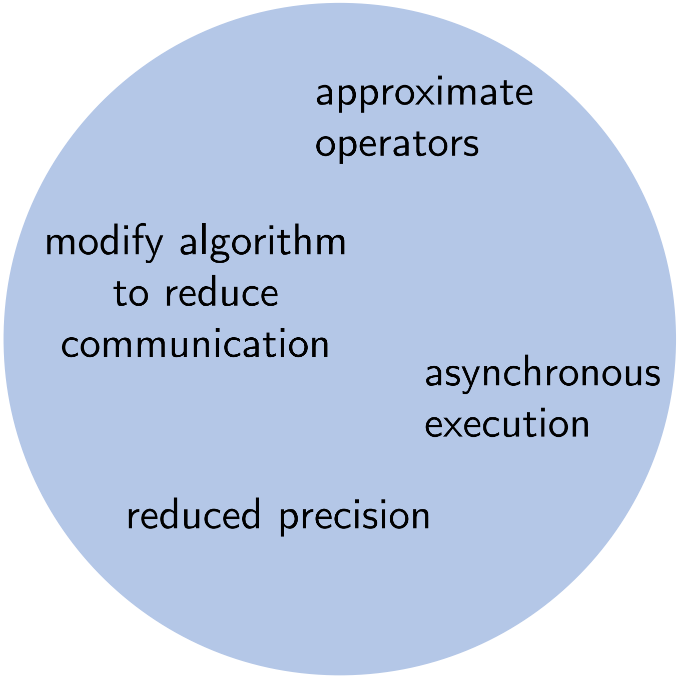
asynchronous execution

reduced precision

Improving Performance of Iterative Solvers

$$\text{runtime} = \left(\begin{matrix}\text{time per}\\ \text{iteration}\end{matrix}\right) \times \left(\begin{matrix}\text{number of iterations}\\ \text{until convergence}\end{matrix}\right)$$

Reduce time per iteration



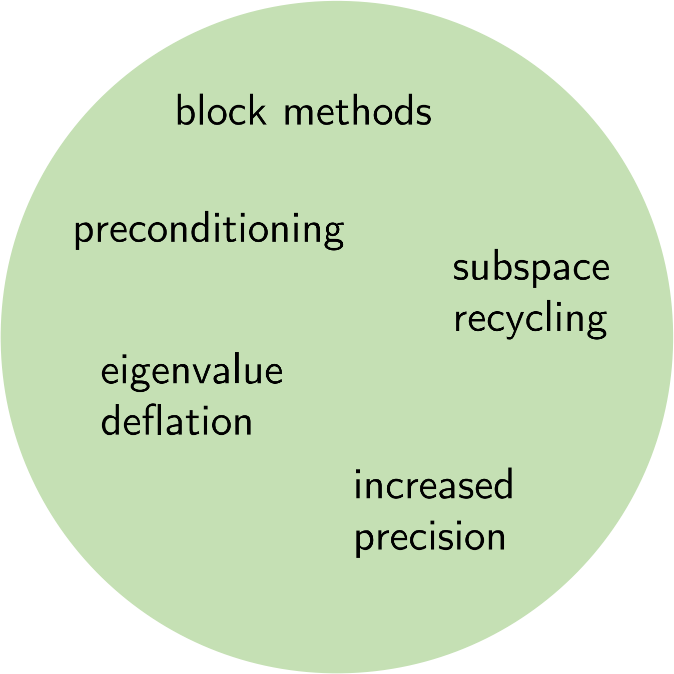
approximate operators

modify algorithm to reduce communication

asynchronous execution

reduced precision

Reduce number of iterations



block methods

preconditioning

subspace recycling

eigenvalue deflation

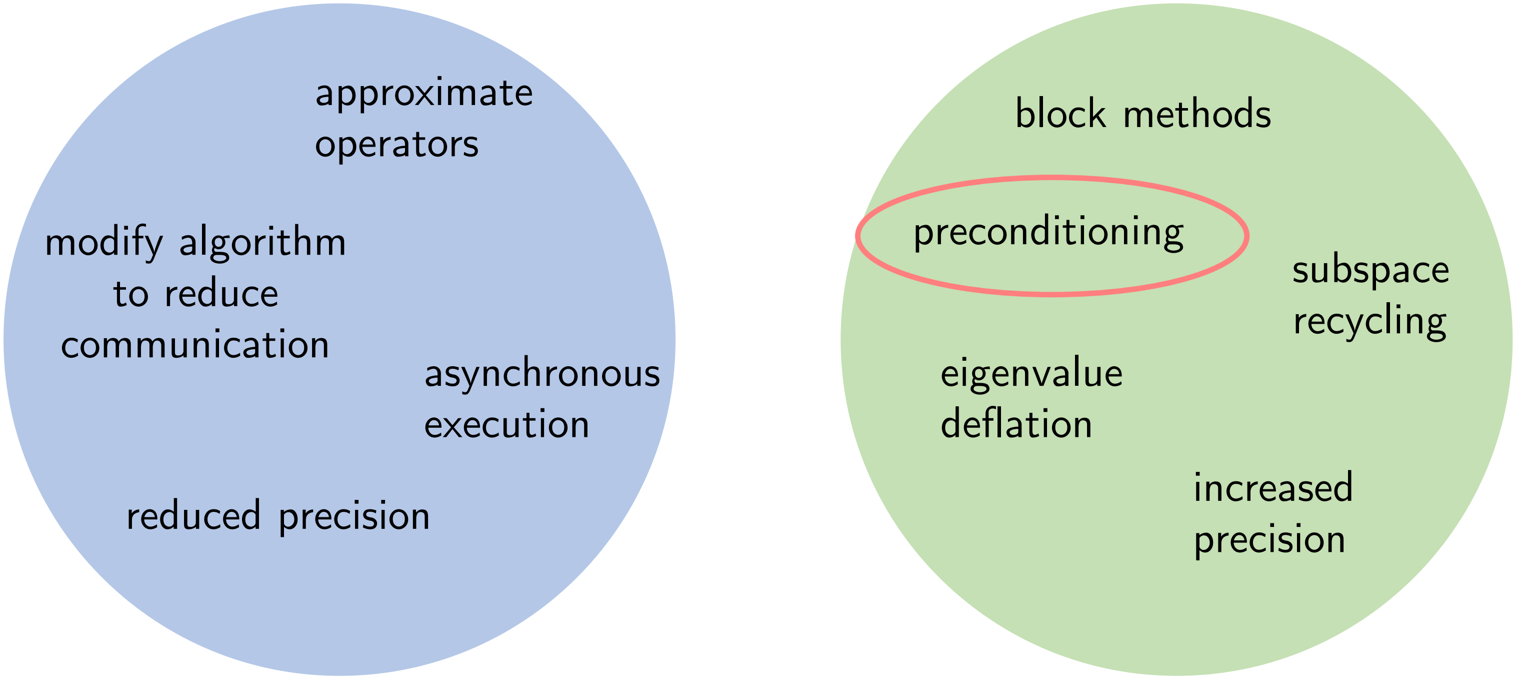
increased precision

Improving Performance of Iterative Solvers

$$\text{runtime} = \begin{pmatrix}\text{time per}\\ \text{iteration}\end{pmatrix} \times \begin{pmatrix}\text{number of iterations}\\ \text{until convergence}\end{pmatrix}$$

Reduce time per iteration

Reduce number of iterations



approximate operators

modify algorithm to reduce communication

asynchronous execution

reduced precision

block methods

preconditioning

subspace recycling

eigenvalue deflation

increased precision

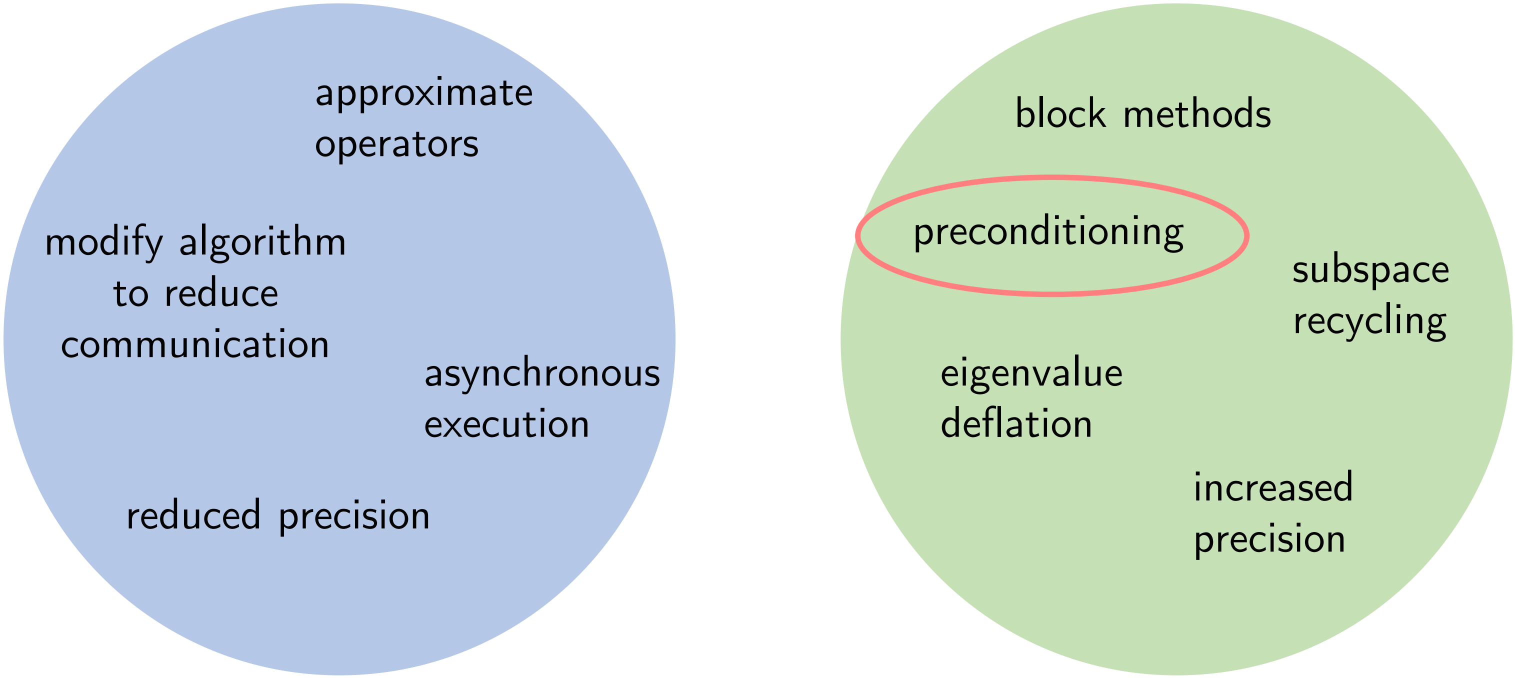
$$Ax = b \quad \Rightarrow \quad M_L^{-1} A M_R^{-1} u = M_L^{-1} b$$
$$x = M_R^{-1} u$$

Improving Performance of Iterative Solvers

$$\text{runtime} = \begin{pmatrix}\text{time per}\\ \text{iteration}\end{pmatrix} \times \begin{pmatrix}\text{number of iterations}\\ \text{until convergence}\end{pmatrix}$$

Reduce time per iteration

Reduce number of iterations



approximate operators

modify algorithm to reduce communication

asynchronous execution

reduced precision

block methods

preconditioning

subspace recycling

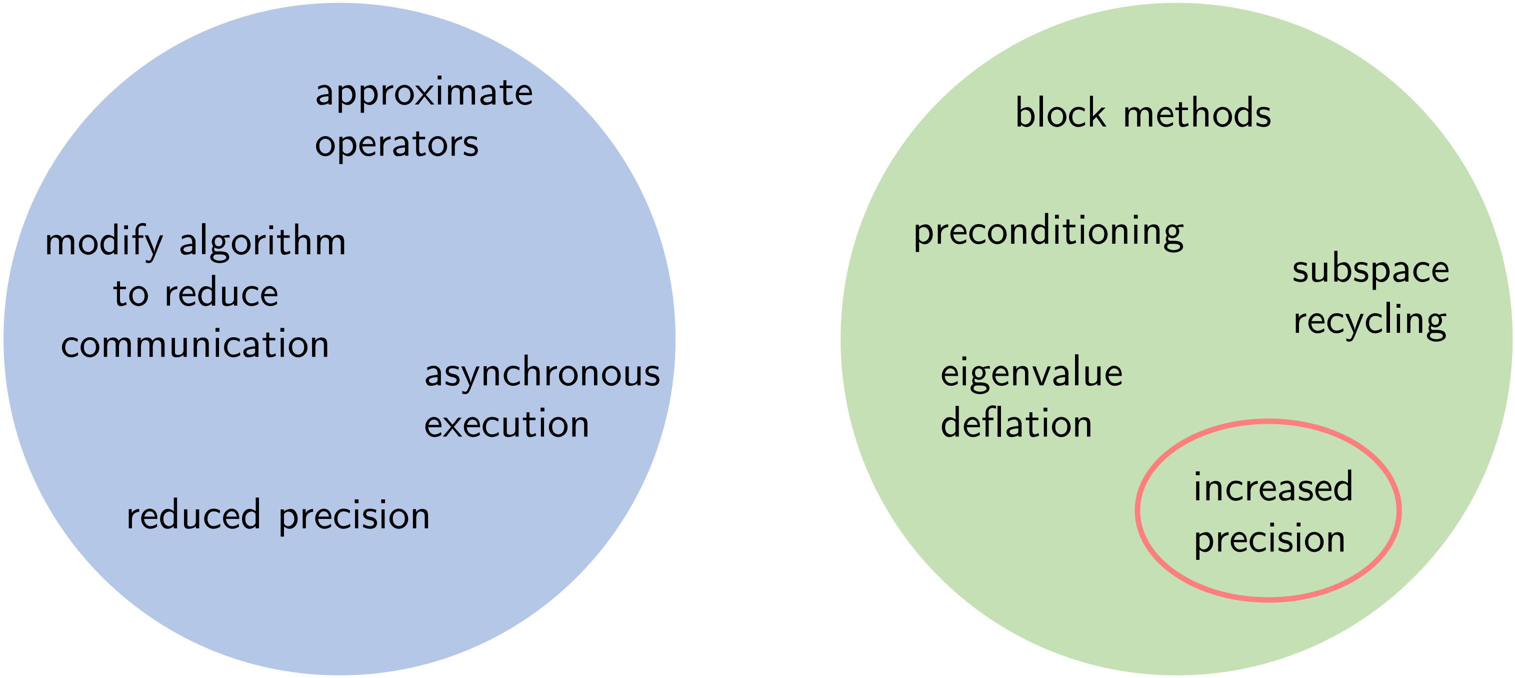
eigenvalue deflation

increased precision

$$\begin{aligned} Ax = b \quad \Rightarrow \quad M_L^{-1} A M_R^{-1} u &= M_L^{-1} b \\ x &= M_R^{-1} u \end{aligned}$$

Improving Performance of Iterative Solvers

$$\text{runtime} = \begin{pmatrix}\text{time per}\\ \text{iteration}\end{pmatrix}\uparrow \times \begin{pmatrix}\text{number of iterations}\\ \text{until convergence}\end{pmatrix}\downarrow$$



Reduce time per iteration

- approximate operators
- modify algorithm to reduce communication
- asynchronous execution
- reduced precision

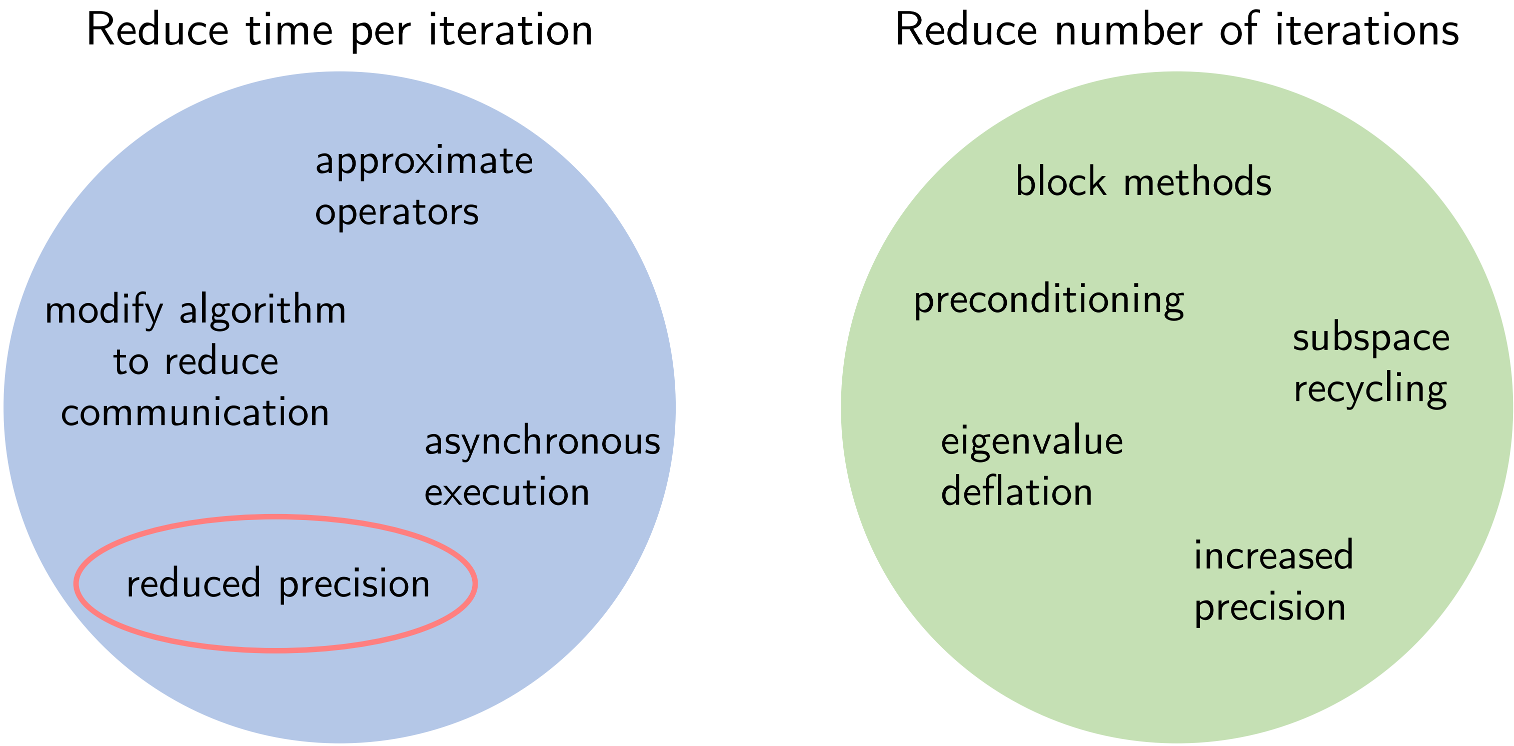
Reduce number of iterations

- block methods
- preconditioning
- subspace recycling
- eigenvalue deflation
- increased precision

doubled precision → twice as many bits moved

Improving Performance of Iterative Solvers

$$\text{runtime} = \begin{pmatrix}\text{time per}\\ \text{iteration}\end{pmatrix}\downarrow \times \begin{pmatrix}\text{number of iterations}\\ \text{until convergence}\end{pmatrix}\uparrow$$



Reduce time per iteration

approximate operators

modify algorithm to reduce communication

asynchronous execution

reduced precision

Reduce number of iterations

block methods

preconditioning

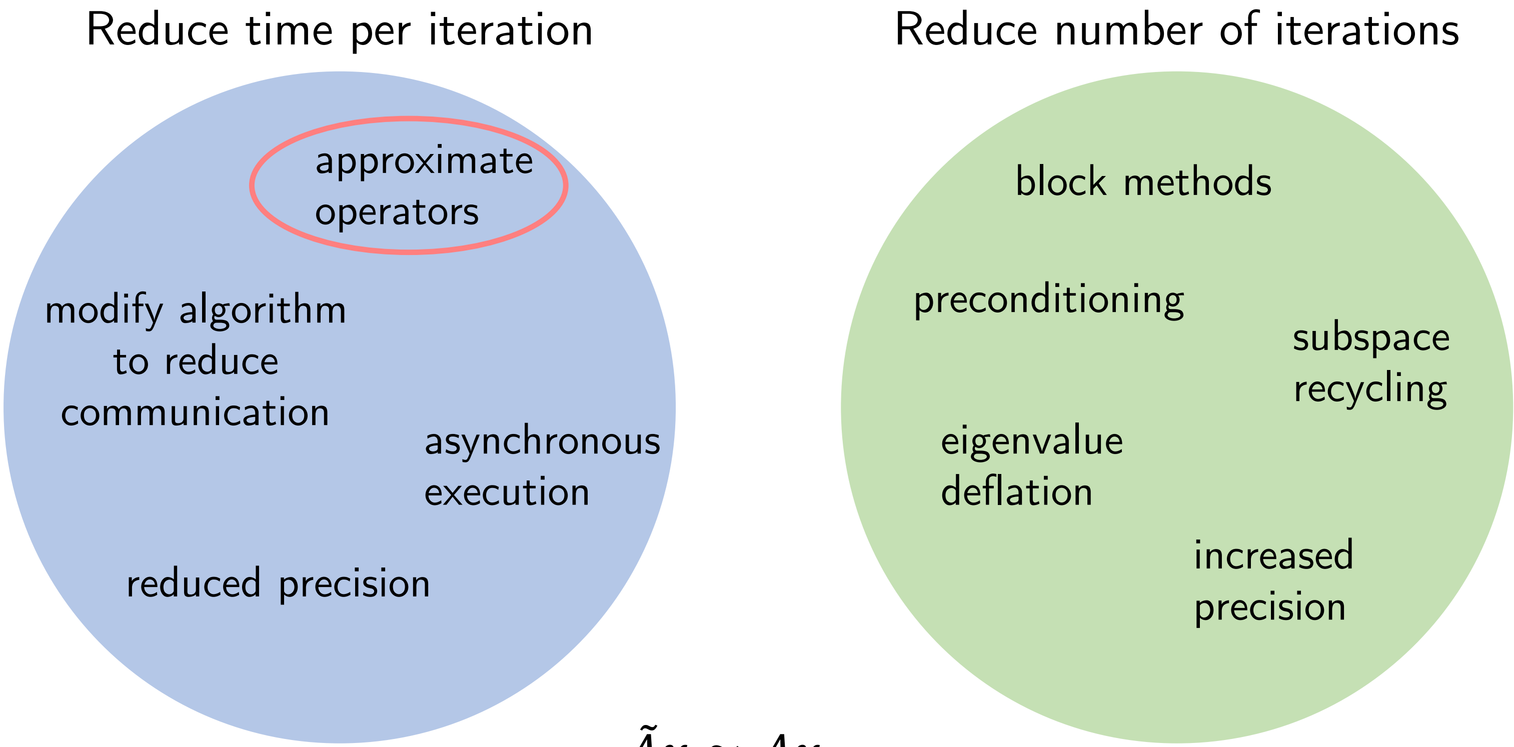
subspace recycling

eigenvalue deflation

increased precision

Improving Performance of Iterative Solvers

$$\text{runtime} = \begin{pmatrix}\text{time per}\\ \text{iteration}\end{pmatrix}\downarrow \times \begin{pmatrix}\text{number of iterations}\\ \text{until convergence}\end{pmatrix}\uparrow$$



Reduce time per iteration

approximate operators

modify algorithm to reduce communication

asynchronous execution

reduced precision

Reduce number of iterations

block methods

preconditioning

subspace recycling

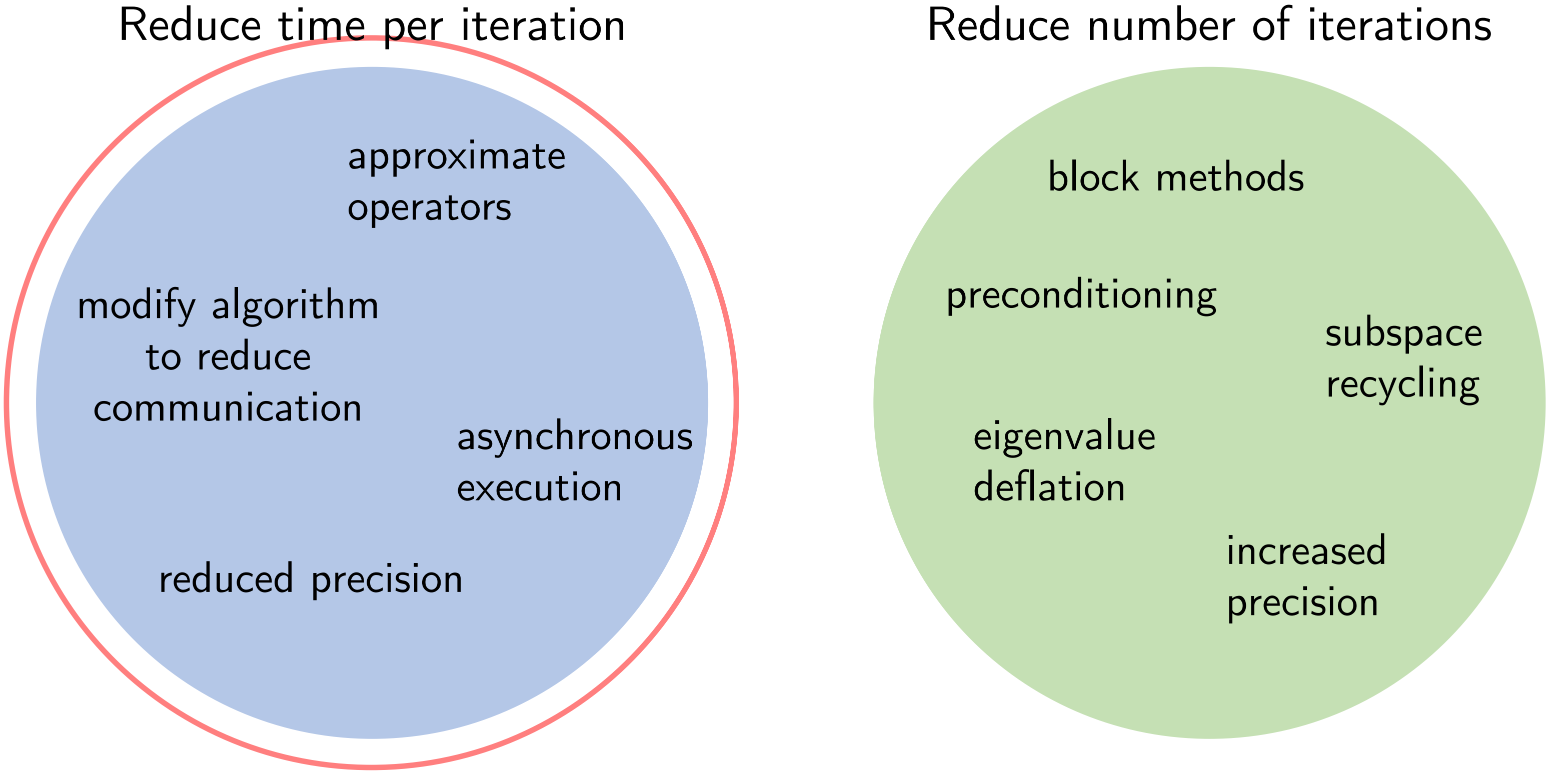
eigenvalue deflation

increased precision

$$\tilde{A}x \approx Ax$$

Improving Performance of Iterative Solvers

$$\text{runtime} = \begin{pmatrix}\text{time per}\\ \text{iteration}\end{pmatrix} \times \begin{pmatrix}\text{number of iterations}\\ \text{until convergence}\end{pmatrix}$$



Reduce time per iteration

approximate operators

modify algorithm to reduce communication

asynchronous execution

reduced precision

Reduce number of iterations

block methods

preconditioning

subspace recycling

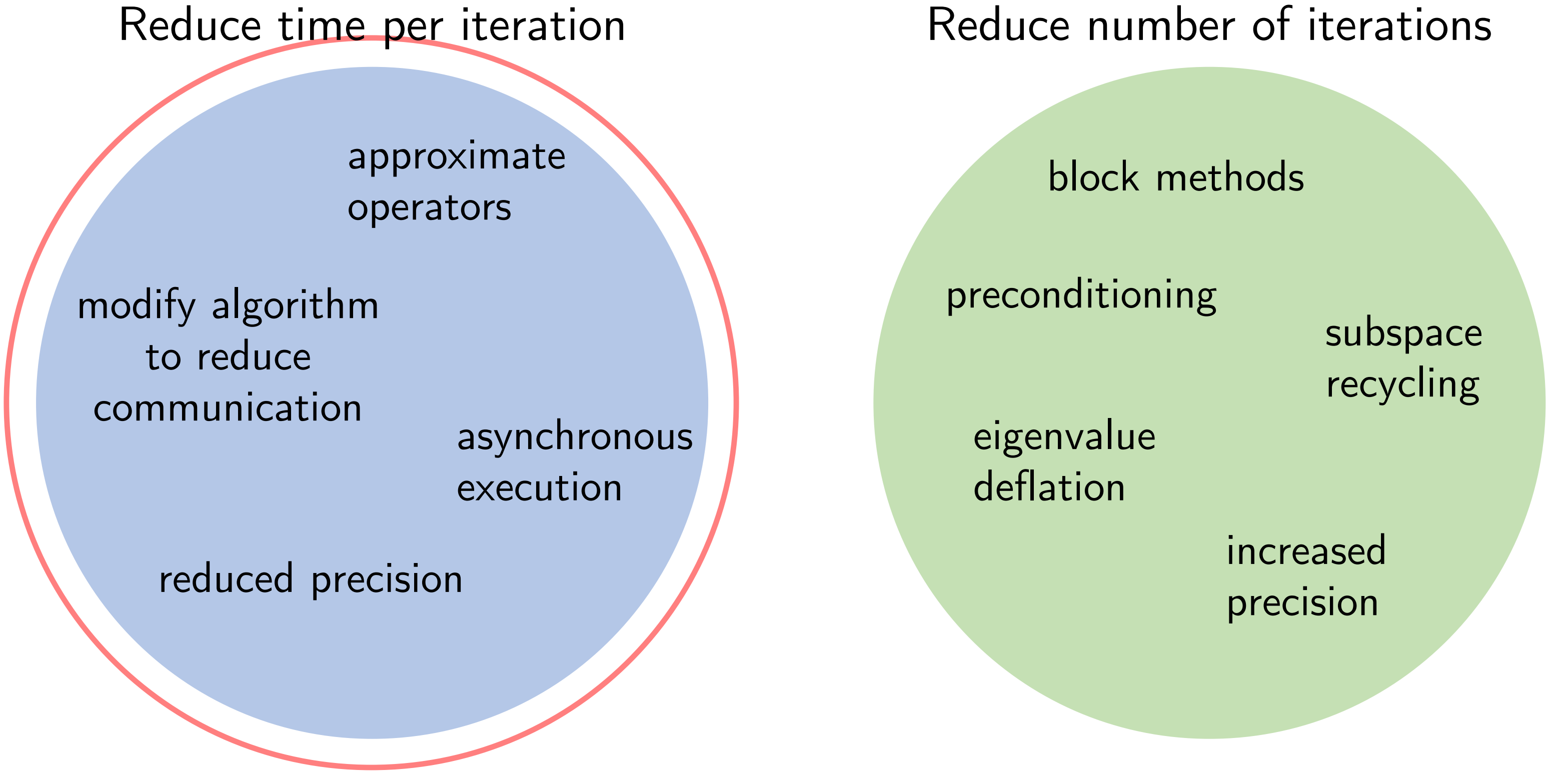
eigenvalue deflation

increased precision

convergence criteria never met: divergence, or convergence to inaccurate solution

Improving Performance of Iterative Solvers

$$\text{runtime} = \binom{\text{time per}}{\text{iteration}} \times \left(\overset{\infty}{\begin{matrix}\text{number of iterations}\\ \text{until convergence}\end{matrix}}\right)$$



Reduce time per iteration

- approximate operators
- modify algorithm to reduce communication
- asynchronous execution
- reduced precision

Reduce number of iterations

- block methods
- preconditioning
- subspace recycling
- eigenvalue deflation
- increased precision

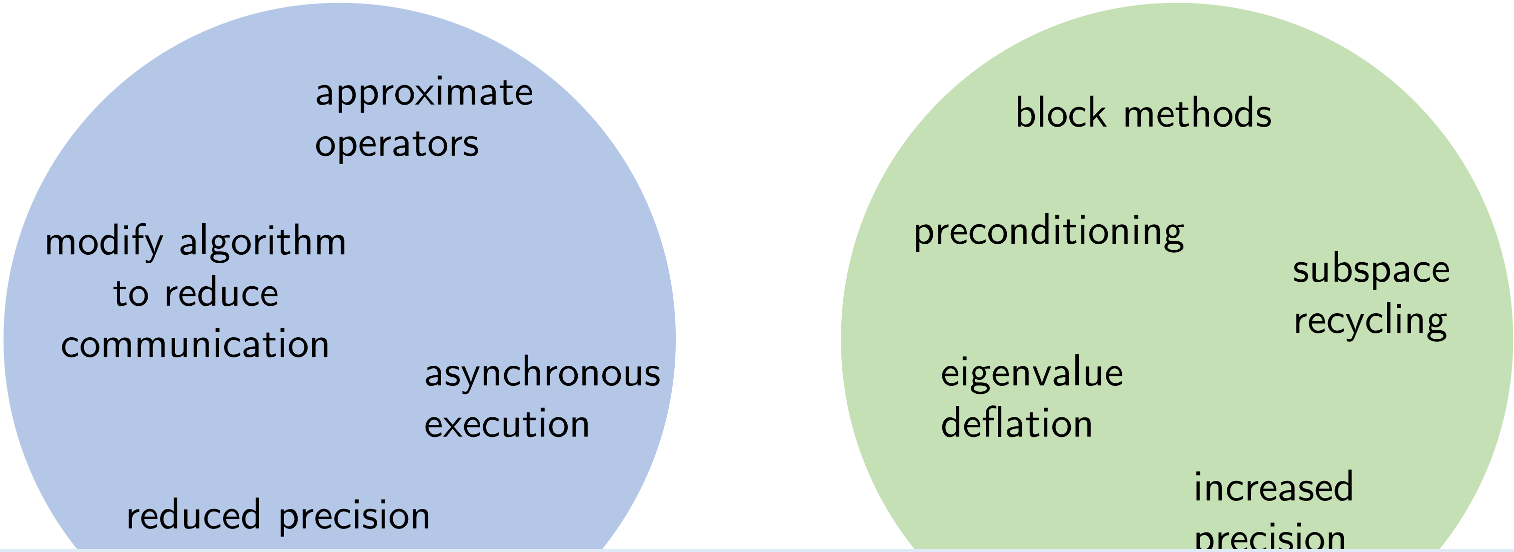
convergence criteria never met: divergence, or convergence to inaccurate solution

Improving Performance of Iterative Solvers

$$\text{runtime} = \begin{pmatrix}\text{time per}\\ \text{iteration}\end{pmatrix} \times \begin{pmatrix}\text{number of iterations}\\ \text{until convergence}\end{pmatrix}$$

Reduce time per iteration

Reduce number of iterations



approximate operators

modify algorithm to reduce communication

asynchronous execution

reduced precision

block methods

preconditioning

subspace recycling

eigenvalue deflation

increased precision

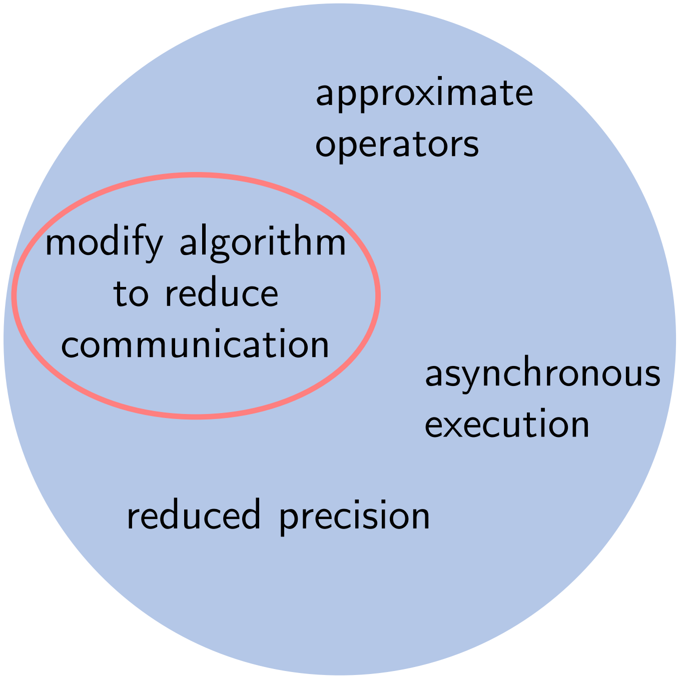
To minimize runtime, must understand how modifications affect:

1) attainable accuracy 2) convergence rate 3) time per iteration

Improving Performance of Iterative Solvers

$$\text{runtime} = \begin{pmatrix}\text{time per}\\ \text{iteration}\end{pmatrix} \times \begin{pmatrix}\text{number of iterations}\\ \text{until convergence}\end{pmatrix}$$

Reduce time per iteration



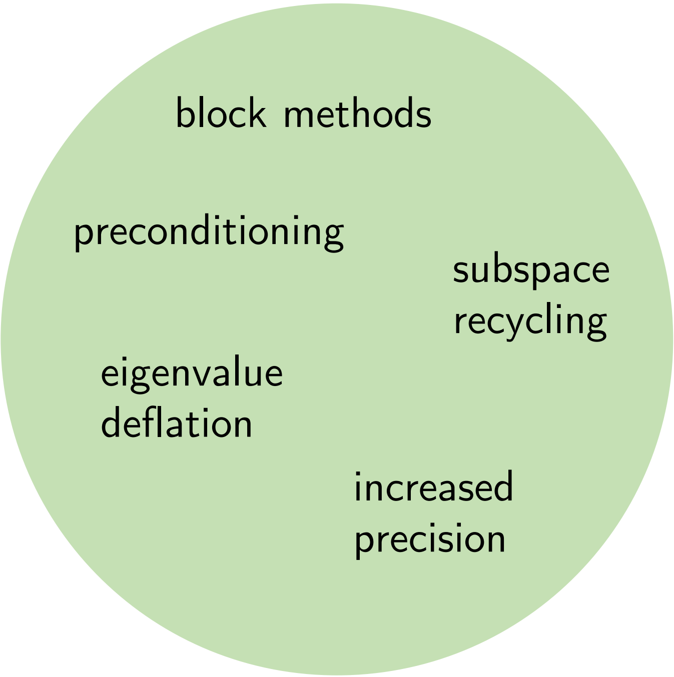
approximate operators

modify algorithm to reduce communication

asynchronous execution

reduced precision

Reduce number of iterations



block methods

preconditioning

subspace recycling

eigenvalue deflation

increased precision

Lecture Outline

- Effects of finite precision in Krylov subspace methods
  - Maximum attainable accuracy
  - Convergence delay
- Existing results for classical Krylov subspace methods
- Results for pipelined and s-step Krylov subspace methods
- Potential remedies for finite precision error in high-performance variants
- Choosing a method in practice
- The future of Krylov subspace methods

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

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

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

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

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

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

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

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

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

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

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

Maximum Attainable Accuracy in HPC Variants

- Various synchronization-reducing modifications/variants discussed in Part I
  - Modified recurrence coefficient computation
  - 3-term CG (STCG)
  - Addition of auxiliary recurrences
  - Pipelined CG
  - s-step methods

Modified recurrence coefficient computation

- What is the effect of changing the way the recurrence coefficients (α and β) are computed in HSCG?

Modified recurrence coefficient computation

- What is the effect of changing the way the recurrence coefficients (α and β) are computed in HSCG?

- Notice that neither α nor β appear in the bounds on $\|f_i\|$

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

Modified recurrence coefficient computation

- What is the effect of changing the way the recurrence coefficients (α and β) are computed in HSCG?

- Notice that neither α nor β appear in the bounds on $\|f_i\|$

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

- As long as the same $\hat{\alpha}_{i-1}$ is used in updating $\hat{x}_i$ and $\hat{r}_i$,

$$f_i = f_{i-1} + A\delta x_i + \delta r_i$$

still holds

- Rounding errors made in computing $\hat{\alpha}_{i-1}$ do not contribute to the residual gap

Modified recurrence coefficient computation

- What is the effect of changing the way the recurrence coefficients (α and β) are computed in HSCG?

- Notice that neither α nor β appear in the bounds on $\|f_i\|$

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

- As long as the same $\hat{\alpha}_{i-1}$ is used in updating $\hat{x}_i$ and $\hat{r}_i$,

$$f_i = f_{i-1} + A\delta x_i + \delta r_i$$

still holds

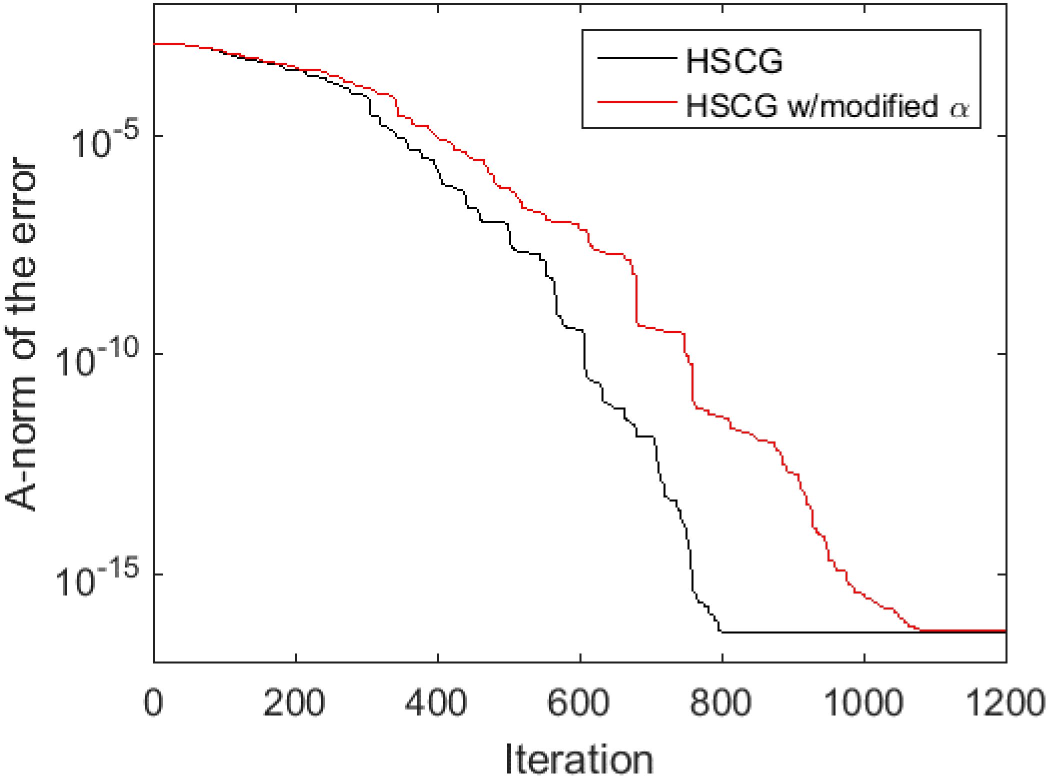
- Rounding errors made in computing $\hat{\alpha}_{i-1}$ do not contribute to the residual gap

- But may change computed $\hat{x}_i$, $\hat{r}_i$, which can affect convergence rate...

Modified recurrence coefficient computation

Example: HSCG with modified formula for α_{i-1}

$$\alpha_{i-1} = \left(\frac{r_{i-1}^T A r_{i-1}}{r_{i-1}^T r_{i-1}} - \frac{\beta_{i-1}}{\alpha_{i-2}} \right)^{-1}$$



Attainable accuracy of STCG

- Analyzed by Gutknecht and Strakoš (2000)
- Attainable accuracy for STCG can be much worse than for HSCG

Attainable accuracy of STCG

- Analyzed by Gutknecht and Strakoš (2000)
- Attainable accuracy for STCG can be much worse than for HSCG

- Residual gap bounded by sum of local errors PLUS local errors multiplied by factors which depend on

$$\max_{0 \leq \ell < j \leq i} \frac{\|r_j\|^2}{\|r_\ell\|^2}$$

Attainable accuracy of STCG

- Analyzed by Gutknecht and Strakoš (2000)
- Attainable accuracy for STCG can be much worse than for HSCG

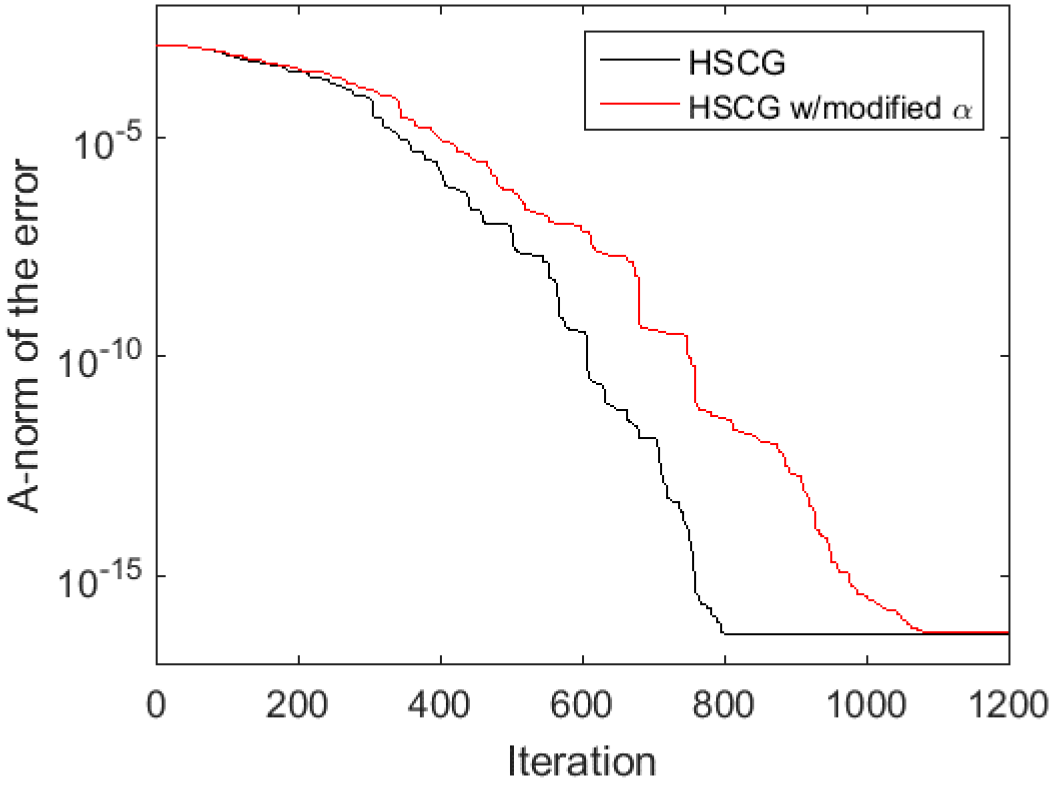
- Residual gap bounded by sum of local errors PLUS local errors multiplied by factors which depend on

$$\max_{0 \le \ell < j \le i} \frac{\|r_j\|^2}{\|r_\ell\|^2}$$

⇒ Large residual oscillations can cause these factors to be large!

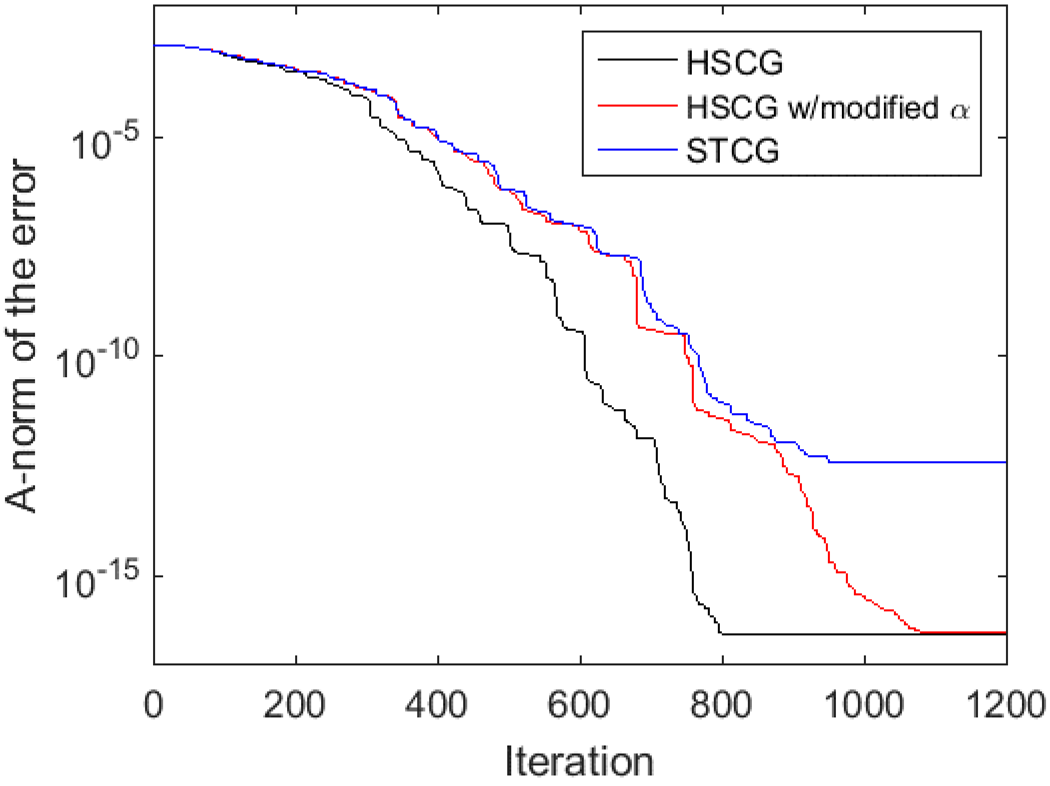
⇒ Local errors can be amplified!

STCG



A-norm of the error
10^{-5}
10^{-10}
10^{-15}
0
200
400
600
800
1000
1200
Iteration
HSCG
HSCG w/modified α

STCG


HSCG
HSCG w/modified α
STCG
A-norm of the error
10^{-5}
10^{-10}
10^{-15}
0
200
400
600
800
1000
1200
Iteration

Attainable accuracy of pipelined CG

- What is the effect of adding auxiliary recurrences to the CG method?

Attainable accuracy of pipelined CG

- What is the effect of adding auxiliary recurrences to the CG method?
- To isolate the effects, we consider a simplified version of a pipelined method

```
$r_0 = b - Ax_0, p_0 = r_0, s_0 = Ap_0$
for $i = 1$:nmax
        $\alpha_{i-1} = \frac{(r_{i-1},r_{i-1})}{(p_{i-1},s_{i-1})}$
        $x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$
        $r_i = r_{i-1} - \alpha_{i-1} s_{i-1}$
        $\beta_i = \frac{(r_i,r_i)}{(r_{i-1},r_{i-1})}$
        $p_i = r_i + \beta_i p_{i-1}$
        $s_i = Ar_i + \beta_i s_{i-1}$
end
```

Attainable accuracy of pipelined CG

- What is the effect of adding auxiliary recurrences to the CG method?
- To isolate the effects, we consider a simplified version of a pipelined method
  - Uses same update formulas for α and β as HSCG, but uses additional recurrence for Ap_i

```
$r_0 = b - Ax_0, p_0 = r_0, s_0 = Ap_0$
for $i$ = 1:nmax
    $\alpha_{i-1} = \frac{(r_{i-1}, r_{i-1})}{(p_{i-1}, s_{i-1})}$
    $x_i = x_{i-1} + \alpha_{i-1} p_{i-1}$
    $r_i = r_{i-1} - \alpha_{i-1} s_{i-1}$
    $\beta_i = \frac{(r_i, r_i)}{(r_{i-1}, r_{i-1})}$
    $p_i = r_i + \beta_i p_{i-1}$
    $s_i = Ar_i + \beta_i s_{i-1}$
end
```

see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]

Attainable accuracy of simple pipelined CG

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

see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]

Attainable accuracy of simple pipelined CG

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

$$f_i = \hat{r}_i - (b - A\hat{x}_i)$$

see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]

Attainable accuracy of simple pipelined CG

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

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

see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]

Attainable accuracy of simple pipelined CG

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

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

where

$$G_i = \hat{S}_i - A\hat{P}_i, \quad d_i = [\hat{\alpha}_0, \dots, \hat{\alpha}_{i-1}]^T$$

see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]

Attainable accuracy of simple pipelined CG

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

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

where

$$G_i = \hat{S}_i - A\hat{P}_i, \quad d_i = [\hat{\alpha}_0, \dots, \hat{\alpha}_{i-1}]^T$$

see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]

Attainable accuracy of simple pipelined CG

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

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

where

$$G_i = \hat{S}_i - A\hat{P}_i, \quad d_i = [\hat{\alpha}_0, \dots, \hat{\alpha}_{i-1}]^T$$

see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]

Attainable accuracy of simple pipelined CG

$$\|G_i\| \leq \frac{O(\varepsilon)}{1 - O(\varepsilon)} \left(\kappa(\widehat{U}_i)\|A\| \left\|\widehat{P}_i\right\| + \|A\| \left\|\widehat{R}_i\right\| \left\|\widehat{U}_i^{-1}\right\| \right)$$

$$\widehat{U}_i = \begin{bmatrix} 1 & -\hat{\beta}_1 & 0 & 0 \\ 0 & 1 & \ddots & 0 \\ \vdots & \ddots & 1 & -\hat{\beta}_{i-1} \\ 0 & \dots & 0 & 1 \end{bmatrix} \qquad \widehat{U}_i^{-1} = \begin{bmatrix} 1 & \hat{\beta}_1 & \dots & \dots & \hat{\beta}_1\hat{\beta}_2 \cdots \hat{\beta}_{i-1} \\ 0 & 1 & \hat{\beta}_2 & \dots & \hat{\beta}_2 \cdots \hat{\beta}_{i-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & \hat{\beta}_{i-1} \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]

Attainable accuracy of simple pipelined CG

$$\|G_i\| \leq \frac{O(\varepsilon)}{1 - O(\varepsilon)} \left(\kappa(\widehat{U}_i) \|A\| \left\|\widehat{P}_i\right\| + \|A\| \left\|\widehat{R}_i\right\| \left\|\widehat{U}_i^{-1}\right\| \right)$$

$$\widehat{U}_i = \begin{bmatrix} 1 & -\hat{\beta}_1 & 0 & 0 \\ 0 & 1 & \ddots & 0 \\ \vdots & \ddots & 1 & -\hat{\beta}_{i-1} \\ 0 & \dots & 0 & 1 \end{bmatrix} \qquad \widehat{U}_i^{-1} = \begin{bmatrix} 1 & \hat{\beta}_1 & \dots & \dots & \hat{\beta}_1\hat{\beta}_2\cdots\hat{\beta}_{i-1} \\ 0 & 1 & \hat{\beta}_2 & \dots & \hat{\beta}_2\cdots\hat{\beta}_{i-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & \hat{\beta}_{i-1} \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

$$\beta_\ell \beta_{\ell+1} \cdots \beta_j = \frac{\left\|r_j\right\|^2}{\|r_{\ell-1}\|^2}, \qquad \ell < j$$

see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]

Attainable accuracy of simple pipelined CG

$$\|G_i\| \leq \frac{O(\varepsilon)}{1 - O(\varepsilon)} \left(\kappa(\widehat{U}_i)\|A\|\|\widehat{P}_i\| + \|A\|\|\widehat{R}_i\|\|\widehat{U}_i^{-1}\| \right)$$

$$\widehat{U}_i = \begin{bmatrix} 1 & -\hat{\beta}_1 & 0 & 0 \\ 0 & 1 & \ddots & 0 \\ \vdots & \ddots & 1 & -\hat{\beta}_{i-1} \\ 0 & \dots & 0 & 1 \end{bmatrix} \qquad \widehat{U}_i^{-1} = \begin{bmatrix} 1 & \hat{\beta}_1 & \dots & \dots & \hat{\beta}_1\hat{\beta}_2\cdots\hat{\beta}_{i-1} \\ 0 & 1 & \hat{\beta}_2 & \dots & \hat{\beta}_2\cdots\hat{\beta}_{i-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & \hat{\beta}_{i-1} \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

$$\beta_\ell\beta_{\ell+1}\cdots\beta_j = \frac{\|r_j\|^2}{\|r_{\ell-1}\|^2}, \qquad \ell < j$$

- Residual oscillations can cause these factors to be large!
- Errors in computed recurrence coefficients can be amplified!

see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]

Attainable accuracy of simple pipelined CG

$$\|G_i\| \leq \frac{O(\varepsilon)}{1 - O(\varepsilon)} \left(\kappa(\widehat{U}_i) \|A\| \|\widehat{P}_i\| + \|A\| \|\widehat{R}_i\| \|\widehat{U}_i^{-1}\| \right)$$

$$\widehat{U}_i = \begin{bmatrix} 1 & -\hat{\beta}_1 & 0 & 0 \\ 0 & 1 & \ddots & 0 \\ \vdots & \ddots & 1 & -\hat{\beta}_{i-1} \\ 0 & \dots & 0 & 1 \end{bmatrix} \qquad \widehat{U}_i^{-1} = \begin{bmatrix} 1 & \hat{\beta}_1 & \dots & \dots & \hat{\beta}_1 \hat{\beta}_2 \cdots \hat{\beta}_{i-1} \\ 0 & 1 & \hat{\beta}_2 & \dots & \hat{\beta}_2 \cdots \hat{\beta}_{i-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & \hat{\beta}_{i-1} \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

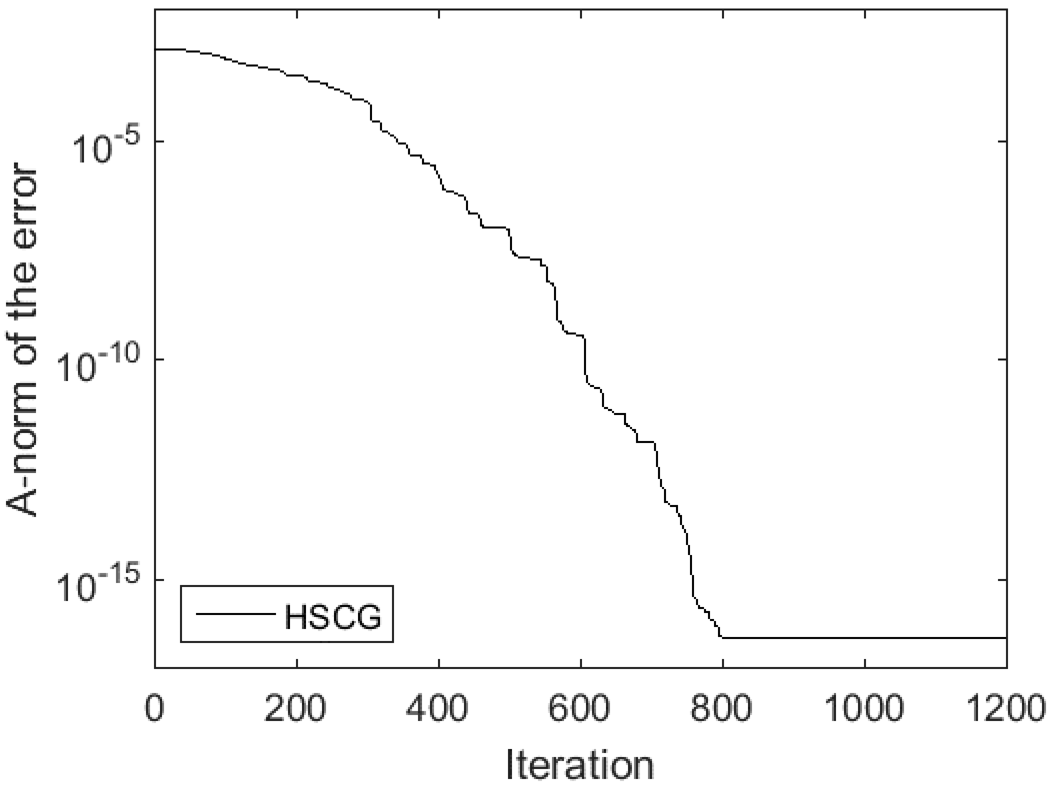
$$\beta_\ell \beta_{\ell+1} \cdots \beta_j = \frac{\|r_j\|^2}{\|r_{\ell-1}\|^2}, \qquad \ell < j$$

- Residual oscillations can cause these factors to be large!
- Errors in computed recurrence coefficients can be amplified!

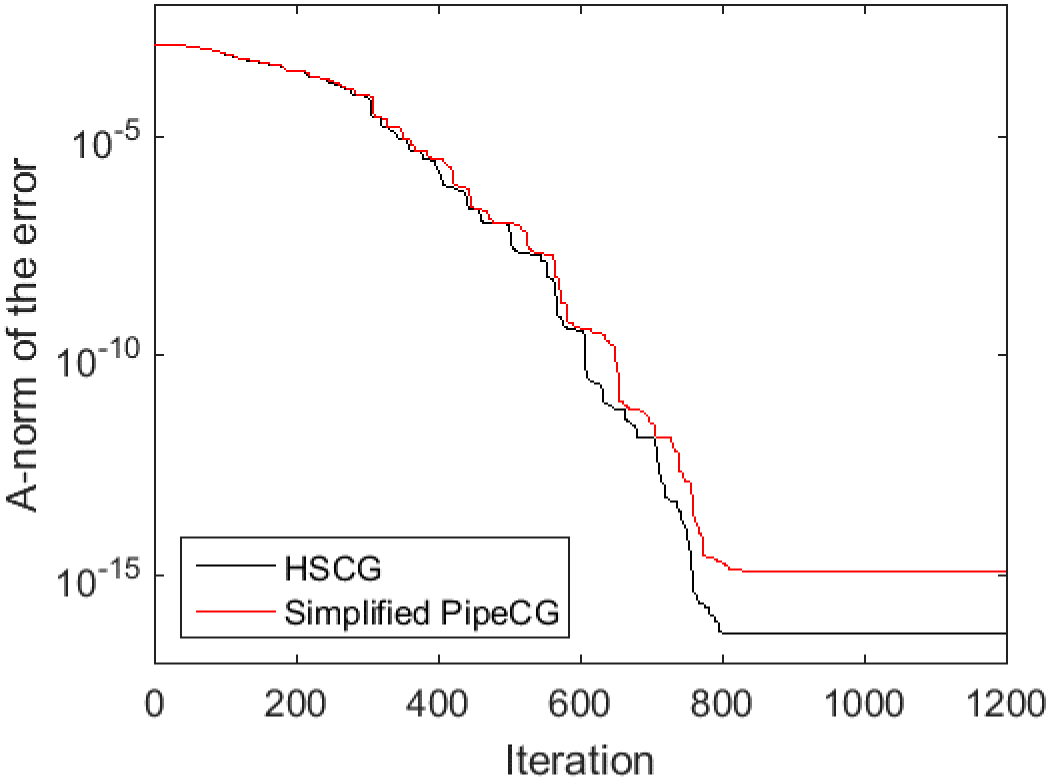
- Very similar to the results for attainable accuracy in the 3-term STCG
- Seemingly innocuous change can cause drastic loss of accuracy

see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]

Simple pipelined CG

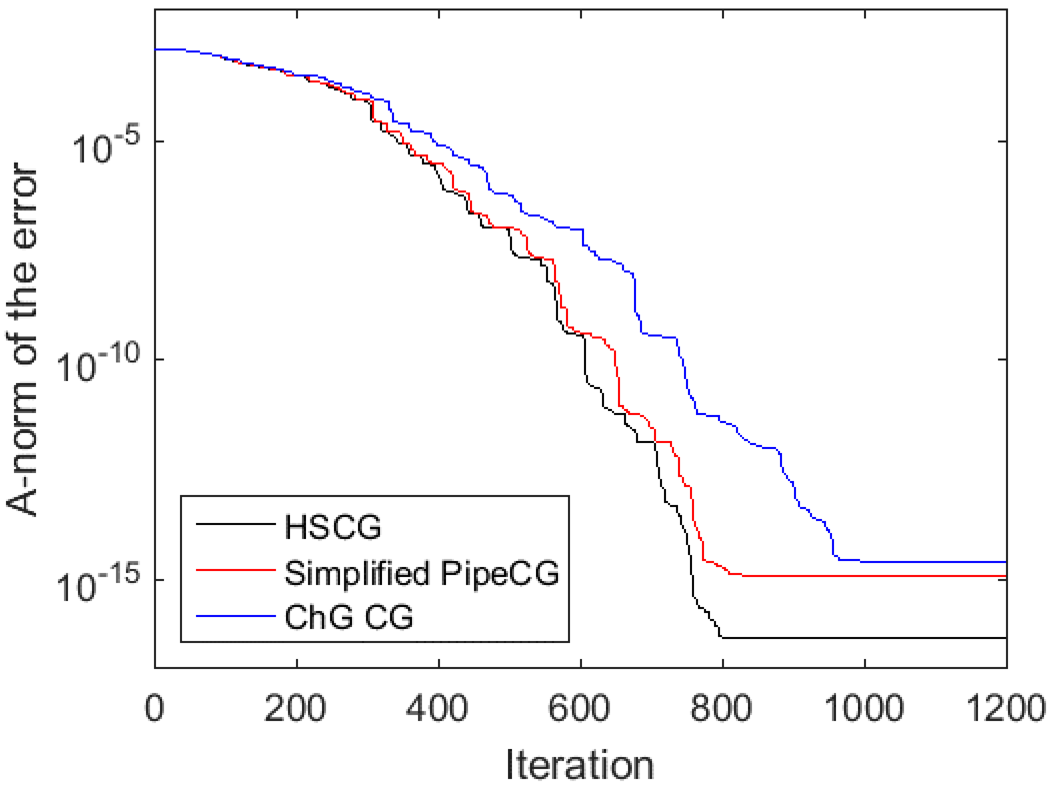


Simple pipelined CG



effect of using auxiliary vector $s_i \equiv Ap_i$

Simple pipelined CG



effect of changing formula for recurrence coefficient α and using auxiliary vector $s_i \equiv Ap_i$

Attainable Accuracy of Pipelined CG

(Cools, et al., 2018)

Pipelined CG uses 5 auxiliary recurrences:

$s_i \equiv Ap_i, \qquad q_i \equiv M^{-1}Ap_i, \qquad u_i \equiv M^{-1}r_i, \qquad w_i = AM^{-1}r_i, \qquad z_i \equiv AM^{-1}Ap_i$

Computed explicitly: $m_i \equiv M^{-1}w_i \; (\equiv M^{-1}AM^{-1}r_i), \;\; v_i = Am_i \; (\equiv AM^{-1}AM^{-1}r_i)$

Attainable Accuracy of Pipelined CG

(Cools, et al., 2018)

Pipelined CG uses 5 auxiliary recurrences:

$$s_i \equiv Ap_i, \qquad q_i \equiv M^{-1}Ap_i, \qquad u_i \equiv M^{-1}r_i, \qquad w_i = AM^{-1}r_i, \qquad z_i \equiv AM^{-1}Ap_i$$

Computed explicitly: $m_i \equiv M^{-1}w_i \; (\equiv M^{-1}AM^{-1}r_i)$, $v_i = Am_i \; (\equiv AM^{-1}AM^{-1}r_i)$

$$\begin{aligned}
\hat{p}_i &= \hat{u}_i + \hat{\beta}_i \hat{p}_{i-1} + \delta_i^p \\
\hat{s}_i &= \widehat{w}_i + \hat{\beta}_i \hat{s}_{i-1} + \delta_i^s \\
\hat{z}_i &= A\widehat{m}_i + \hat{\beta}_i \hat{z}_{i-1} + \delta_i^z \\
\hat{q}_i &= \widehat{m}_i + \hat{\beta}_i \hat{q}_{i-1} + \delta_i^q
\end{aligned}$$

$$\begin{aligned}
\hat{x}_{i+1} &= \hat{x}_i + \hat{\alpha}_i \hat{p}_i + \delta_i^x \\
\hat{r}_{i+1} &= \hat{r}_i - \hat{\alpha}_i \hat{s}_i + \delta_i^r \\
\widehat{w}_{i+1} &= \widehat{w}_i - \hat{\alpha}_i \hat{z}_i + \delta_i^w \\
\hat{u}_{i+1} &= u_i - \hat{\alpha}_i \hat{q}_i + \delta_i^u
\end{aligned}$$

Attainable Accuracy of Pipelined CG

(Cools, et al., 2018)

Pipelined CG uses 5 auxiliary recurrences:

$$s_i \equiv Ap_i, \qquad q_i \equiv M^{-1}Ap_i, \qquad u_i \equiv M^{-1}r_i, \qquad w_i = AM^{-1}r_i, \qquad z_i \equiv AM^{-1}Ap_i$$

Computed explicitly: $m_i \equiv M^{-1}w_i \; (\equiv M^{-1}AM^{-1}r_i)$, $v_i = Am_i \; (\equiv AM^{-1}AM^{-1}r_i)$

$$\begin{aligned} \hat{p}_i &= \hat{u}_i + \hat{\beta}_i \hat{p}_{i-1} + \delta_i^p \\ \hat{s}_i &= \widehat{w}_i + \hat{\beta}_i \hat{s}_{i-1} + \delta_i^s \\ \hat{z}_i &= A\widehat{m}_i + \hat{\beta}_i \hat{z}_{i-1} + \delta_i^z \\ \hat{q}_i &= \widehat{m}_i + \hat{\beta}_i \hat{q}_{i-1} + \delta_i^q \end{aligned} \qquad\qquad \begin{aligned} \hat{x}_{i+1} &= \hat{x}_i + \hat{\alpha}_i \hat{p}_i + \delta_i^x \\ \hat{r}_{i+1} &= \hat{r}_i - \hat{\alpha}_i \hat{s}_i + \delta_i^r \\ \widehat{w}_{i+1} &= \widehat{w}_i - \hat{\alpha}_i \hat{z}_i + \delta_i^w \\ \hat{u}_{i+1} &= u_i - \hat{\alpha}_i \hat{q}_i + \delta_i^u \end{aligned}$$

$$\begin{aligned} f_{i+1} &= (b - A\hat{x}_{i+1}) - \hat{r}_{i+1} \\ &= f_i - \hat{\alpha}_i (A\hat{p}_i - \hat{s}_i) - A\delta_i^x - \delta_i^r \end{aligned}$$

Attainable Accuracy of Pipelined CG

(Cools, et al., 2018)

Pipelined CG uses 5 auxiliary recurrences:

$$s_i \equiv Ap_i, \qquad q_i \equiv M^{-1}Ap_i, \qquad u_i \equiv M^{-1}r_i, \qquad w_i = AM^{-1}r_i, \qquad z_i \equiv AM^{-1}Ap_i$$

Computed explicitly: $m_i \equiv M^{-1}w_i \; (\equiv M^{-1}AM^{-1}r_i)$, $\; v_i = Am_i \; (\equiv AM^{-1}AM^{-1}r_i)$

$$\begin{aligned} \hat{p}_i &= \hat{u}_i + \hat{\beta}_i \hat{p}_{i-1} + \delta_i^p \\ \hat{s}_i &= \widehat{w}_i + \hat{\beta}_i \hat{s}_{i-1} + \delta_i^s \\ \hat{z}_i &= A\widehat{m}_i + \hat{\beta}_i \hat{z}_{i-1} + \delta_i^z \\ \hat{q}_i &= \widehat{m}_i + \hat{\beta}_i \hat{q}_{i-1} + \delta_i^q \end{aligned} \qquad\qquad \begin{aligned} \hat{x}_{i+1} &= \hat{x}_i + \hat{\alpha}_i \hat{p}_i + \delta_i^x \\ \hat{r}_{i+1} &= \hat{r}_i - \hat{\alpha}_i \hat{s}_i + \delta_i^r \\ \widehat{w}_{i+1} &= \widehat{w}_i - \hat{\alpha}_i \hat{z}_i + \delta_i^w \\ \hat{u}_{i+1} &= u_i - \hat{\alpha}_i \hat{q}_i + \delta_i^u \end{aligned}$$

$$\begin{aligned} f_{i+1} &= (b - A\hat{x}_{i+1}) - \hat{r}_{i+1} \\ &= f_i - \hat{\alpha}_i (\underbrace{A\hat{p}_i - \hat{s}_i}_{g_i \,=\, \hat{\beta}_i g_{i-1} + (A\hat{u}_{i+1} - \widehat{w}_{i+1}) + A\delta_i^p - \delta_i^s}) - A\delta_i^x - \delta_i^r \end{aligned}$$

Attainable Accuracy of Pipelined CG

(Cools, et al., 2018)

Pipelined CG uses 5 auxiliary recurrences:

$$s_i \equiv Ap_i, \qquad q_i \equiv M^{-1}Ap_i, \qquad u_i \equiv M^{-1}r_i, \qquad w_i = AM^{-1}r_i, \qquad z_i \equiv AM^{-1}Ap_i$$

Computed explicitly: $m_i \equiv M^{-1}w_i \; (\equiv M^{-1}AM^{-1}r_i)$, $v_i = Am_i \; (\equiv AM^{-1}AM^{-1}r_i)$

$$\begin{aligned}
\hat{p}_i &= \hat{u}_i + \hat{\beta}_i \hat{p}_{i-1} + \delta_i^p \\
\hat{s}_i &= \widehat{w}_i + \hat{\beta}_i \hat{s}_{i-1} + \delta_i^s \\
\hat{z}_i &= A\widehat{m}_i + \hat{\beta}_i \hat{z}_{i-1} + \delta_i^z \\
\hat{q}_i &= \widehat{m}_i + \hat{\beta}_i \hat{q}_{i-1} + \delta_i^q
\end{aligned}
\qquad\qquad
\begin{aligned}
\hat{x}_{i+1} &= \hat{x}_i + \hat{\alpha}_i \hat{p}_i + \delta_i^x \\
\hat{r}_{i+1} &= \hat{r}_i - \hat{\alpha}_i \hat{s}_i + \delta_i^r \\
\widehat{w}_{i+1} &= \widehat{w}_i - \hat{\alpha}_i \hat{z}_i + \delta_i^w \\
\hat{u}_{i+1} &= u_i - \hat{\alpha}_i \hat{q}_i + \delta_i^u
\end{aligned}$$

$$f_{i+1} = (b - A\hat{x}_{i+1}) - \hat{r}_{i+1}$$

$$= f_i - \hat{\alpha}_i \underbrace{(A\hat{p}_i - \hat{s}_i)} - A\delta_i^x - \delta_i^r$$

$$g_i = \hat{\beta}_i g_{i-1} + \underbrace{(A\hat{u}_{i+1} - \widehat{w}_{i+1})} + A\delta_i^p - \delta_i^s$$

$$h_{i+1} = h_i - \hat{\alpha}_i (A\hat{q}_i - \hat{z}_i) + A\delta_i^u - \delta_i^w$$

Attainable Accuracy of Pipelined CG

(Cools, et al., 2018)

Pipelined CG uses 5 auxiliary recurrences:

$$s_i \equiv Ap_i, \qquad q_i \equiv M^{-1}Ap_i, \qquad u_i \equiv M^{-1}r_i, \qquad w_i = AM^{-1}r_i, \qquad z_i \equiv AM^{-1}Ap_i$$

Computed explicitly: $m_i \equiv M^{-1}w_i \; (\equiv M^{-1}AM^{-1}r_i)$, $v_i = Am_i \; (\equiv AM^{-1}AM^{-1}r_i)$

$$\begin{aligned} \hat{p}_i &= \hat{u}_i + \hat{\beta}_i\hat{p}_{i-1} + \delta_i^p \\ \hat{s}_i &= \widehat{w}_i + \hat{\beta}_i\hat{s}_{i-1} + \delta_i^s \\ \hat{z}_i &= A\widehat{m}_i + \hat{\beta}_i\hat{z}_{i-1} + \delta_i^z \\ \hat{q}_i &= \widehat{m}_i + \hat{\beta}_i\hat{q}_{i-1} + \delta_i^q \end{aligned} \qquad\qquad \begin{aligned} \hat{x}_{i+1} &= \hat{x}_i + \hat{\alpha}_i\hat{p}_i + \delta_i^x \\ \hat{r}_{i+1} &= \hat{r}_i - \hat{\alpha}_i\hat{s}_i + \delta_i^r \\ \widehat{w}_{i+1} &= \widehat{w}_i - \hat{\alpha}_i\hat{z}_i + \delta_i^w \\ \hat{u}_{i+1} &= u_i - \hat{\alpha}_i\hat{q}_i + \delta_i^u \end{aligned}$$

$$f_{i+1} = (b - A\hat{x}_{i+1}) - \hat{r}_{i+1}$$

$$= f_i - \hat{\alpha}_i\underbrace{(A\hat{p}_i - \hat{s}_i)} - A\delta_i^x - \delta_i^r$$

$$g_i = \hat{\beta}_i g_{i-1} + \underbrace{(A\hat{u}_{i+1} - \widehat{w}_{i+1})} + A\delta_i^p - \delta_i^s$$

$$h_{i+1} = h_i - \hat{\alpha}_i\underbrace{(A\hat{q}_i - \hat{z}_i)} + A\delta_i^u - \delta_i^w$$

$$j_i = \hat{\beta}_i j_{i-1} + A\delta_i^q - \delta_i^z$$

Attainable Accuracy of Pipelined CG

$$f_{i+1} = f_0 - \sum_{j=0}^{i} \hat{\alpha}_j g_j - \sum_{j=0}^{i} (A\delta_j^x + \delta_j^r)$$

Attainable Accuracy of Pipelined CG

$$f_{i+1} = f_0 - \sum_{j=0}^{i} \hat{\alpha}_j g_j - \sum_{j=0}^{i} (A\delta_j^x + \delta_j^r)$$

$$g_j = \left(\prod_{k=1}^{j} \hat{\beta}_k\right) g_0 + \sum_{k=1}^{j} \left(\prod_{\ell=k+1}^{j} \hat{\beta}_\ell\right) \left(A\delta_k^p - \delta_k^s\right) + \sum_{k=1}^{j} \left(\prod_{\ell=k+1}^{j} \hat{\beta}_\ell\right) h_k$$

$$h_k = h_0 - \sum_{\ell=0}^{k-1} \hat{\alpha}_\ell j_\ell + \sum_{\ell=0}^{k-1} (A\delta_\ell^u + \delta_\ell^w)$$

$$j_\ell = \left(\prod_{m=1}^{\ell} \hat{\beta}_m\right) j_0 + \sum_{m=1}^{\ell} \left(\prod_{n=m+1}^{\ell} \hat{\beta}_n\right) \left(A\delta_m^q - \delta_m^z\right)$$

Attainable Accuracy of Pipelined CG

$$f_{i+1} = f_0 - \sum_{j=0}^{i} \hat{\alpha}_j g_j - \sum_{j=0}^{i} (A\delta_j^x + \delta_j^r)$$

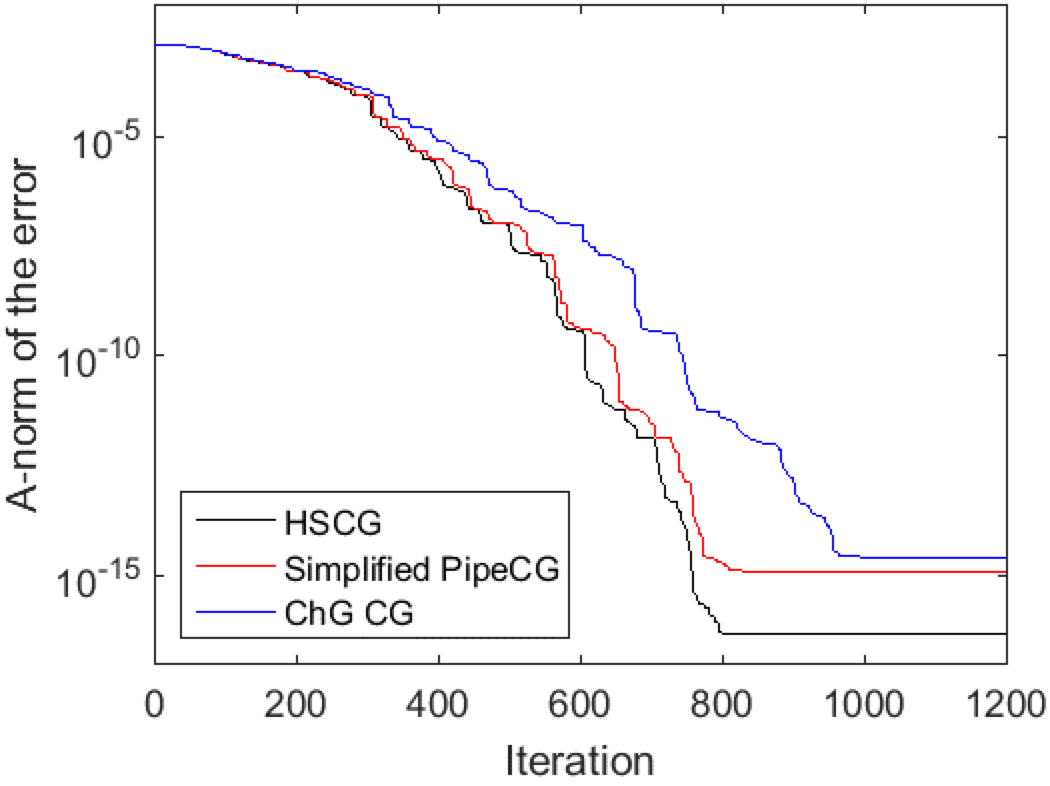
$$g_j = \left(\prod_{k=1}^{j} \hat{\beta}_k\right) g_0 + \sum_{k=1}^{j} \left(\prod_{\ell=k+1}^{j} \hat{\beta}_\ell\right) \left(A\delta_k^p - \delta_k^s\right) + \sum_{k=1}^{j} \left(\prod_{\ell=k+1}^{j} \hat{\beta}_\ell\right) h_k$$

$$h_k = h_0 - \sum_{\ell=0}^{k-1} \hat{\alpha}_\ell j_\ell + \sum_{\ell=0}^{k-1} (A\delta_\ell^u - \delta_\ell^w)$$

$$j_\ell = \left(\prod_{m=1}^{\ell} \hat{\beta}_m\right) j_0 + \sum_{m=1}^{\ell} \left(\prod_{n=m+1}^{\ell} \hat{\beta}_n\right) \left(A\delta_m^q - \delta_m^z\right)$$

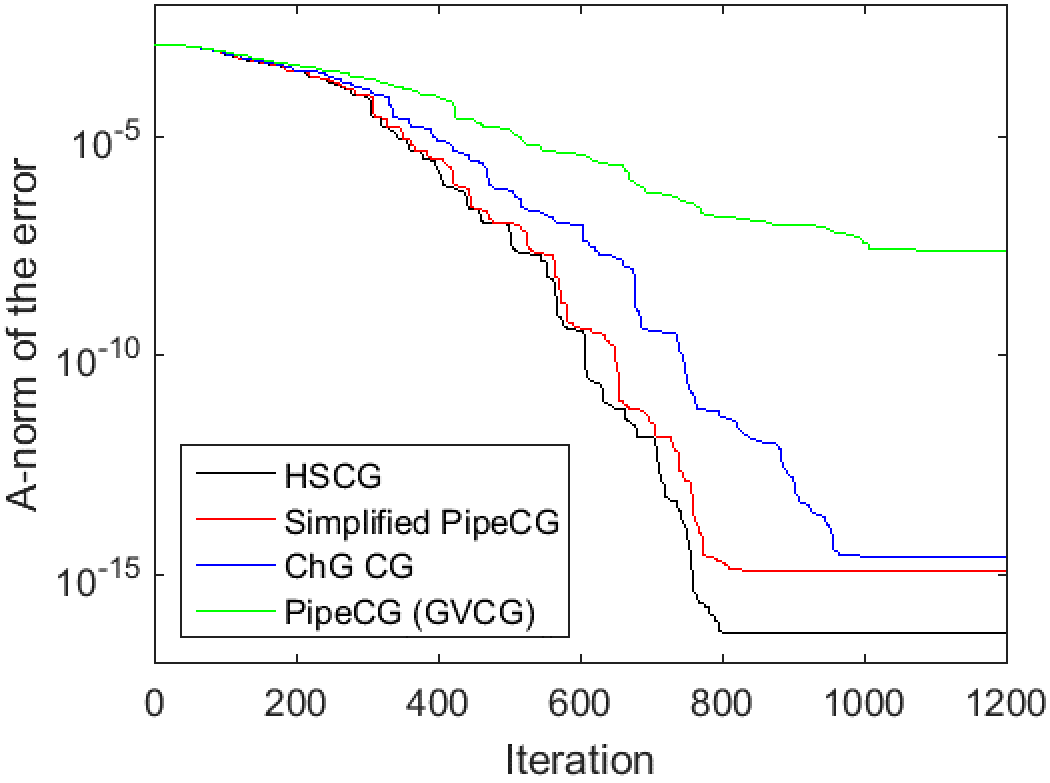
Local rounding errors
all potentially
amplified!

Pipelined CG



effect of changing formula for recurrence coefficient α and using auxiliary vector $s_i \equiv Ap_i$

Pipelined CG

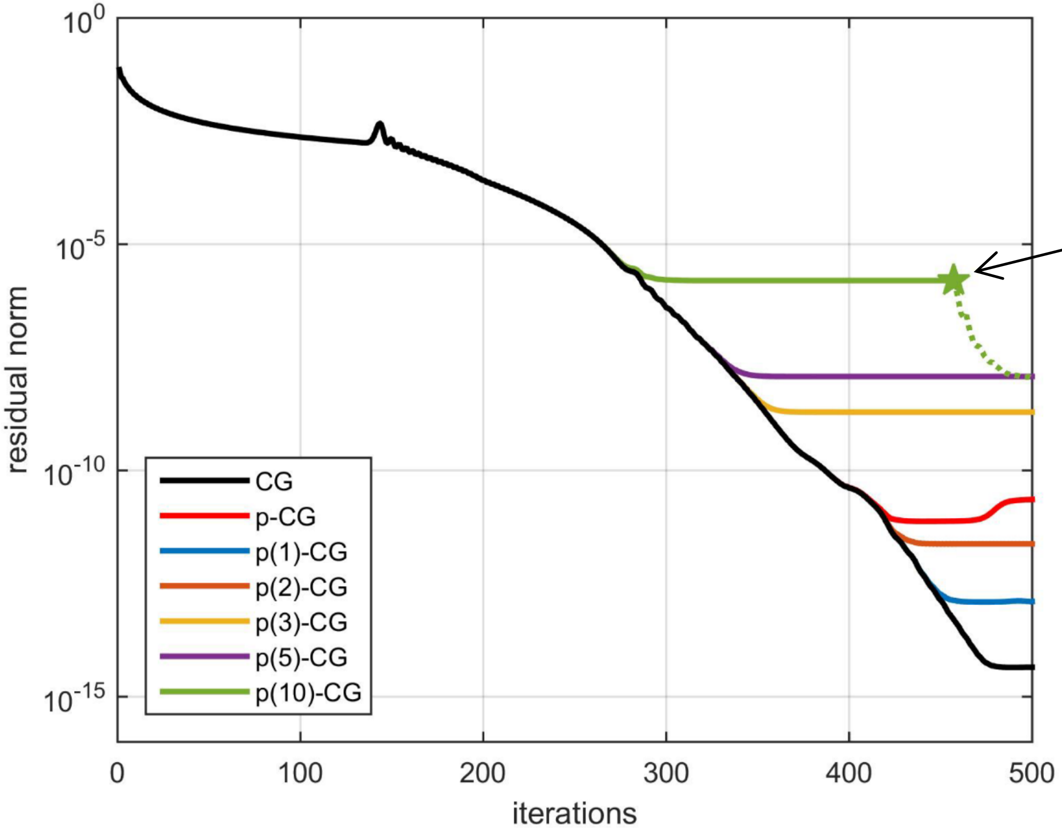


effect of changing formula for recurrence coefficient α and using auxiliary vectors $s_i \equiv Ap_i, w_i \equiv Ar_i, z_i \equiv A^2 r_i$

Effect of Deeper Pipelines

- Deeper pipeline -> effectively adding more auxiliary recurrences
- We expect residual gap to increase with increasing pipeline depth
- Some initial work (Cools, 2018) uses Chebyshev shifts to attempt to stabilize (deep) pipelined CG; but increasing gap is still apparent

2D Poisson problem, $N = 200$, b set such that $x_i = 1/\sqrt{N}$



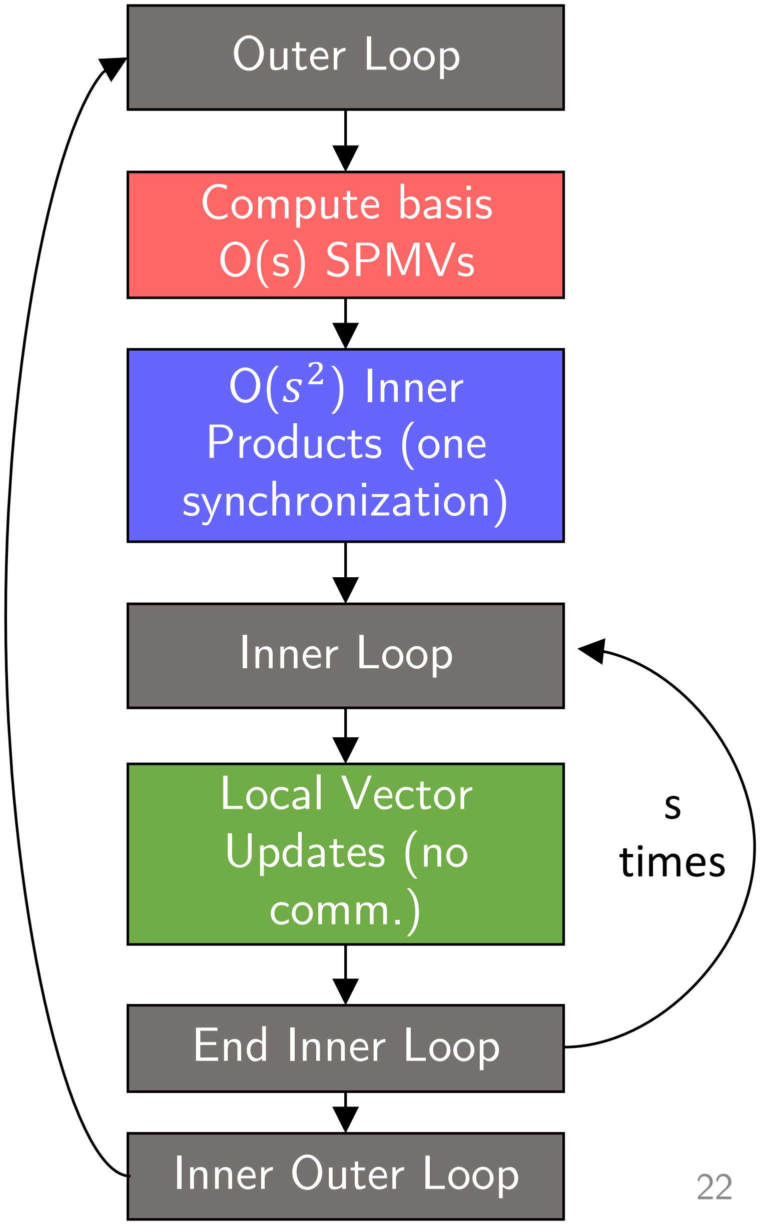
square root breakdown + explicit restart

(Cools, 2018)

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
        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 $j = 1: s$
        $\alpha_{sk+j-1} = \frac{r_{j-1}'^T \mathcal{G}_k r_{j-1}'}{p_{j-1}'^T \mathcal{G}_k \mathcal{B}_k p_{j-1}'}$
        $x_j' = x_{j-1}' + \alpha_{sk+j-1} p_{j-1}'$
        $r_j' = r_{j-1}' - \alpha_{sk+j-1} \mathcal{B}_k p_{j-1}'$
        $\beta_{sk+j} = \frac{r_j'^T \mathcal{G}_k r_j'}{r_{j-1}'^T \mathcal{G}_k r_{j-1}'}$
        $p_j' = r_j' + \beta_{sk+j} p_{j-1}'$
    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
```



Outer Loop

Compute basis O(s) SPMVs

O(s^2) Inner Products (one synchronization)

Inner Loop

Local Vector Updates (no comm.)

End Inner Loop

Inner Outer Loop

s times

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:

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

Recovering CG vectors for use in next outer loop:

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

Sources of local roundoff error in s-step CG

Computing the s-step Krylov subspace basis:

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

Error in computing s-step basis

Updating coordinate vectors in the inner loop:

$$\hat{x}'_{k,j} = \hat{x}'_{k,j-1} + \hat{q}'_{k,j-1} + \boxed{\xi_{k,j}}$$

$$\hat{r}'_{k,j} = \hat{r}'_{k,j-1} - \mathcal{B}_k\, \hat{q}'_{k,j-1} + \boxed{\eta_{k,j}}$$

$$\text{with} \quad \hat{q}'_{k,j-1} = \mathrm{fl}(\hat{\alpha}_{sk+j-1}\hat{p}'_{k,j-1})$$

Error in updating coefficient 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 coefficient 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

- We can write the gap between the true and updated residuals f in terms of these errors:

$$
\boxed{
\begin{aligned}
f_{sk+j} &= f_0 \\
&\quad - \sum_{\ell=0}^{k-1} \left[A\phi_{s\ell+s} + \psi_{s\ell+s} + \sum_{i=1}^{s} \left[A\hat{\mathcal{Y}}_{\ell}\xi_{\ell,i} + \hat{\mathcal{Y}}_{\ell}\eta_{\ell,i} - \Delta\mathcal{Y}_{\ell}\hat{q}'_{\ell,i-1} \right] \right] \\
&\quad - A\phi_{sk+j} - \psi_{sk+j} - \sum_{i=1}^{j} \left[A\hat{\mathcal{Y}}_{k}\xi_{k,i} + \hat{\mathcal{Y}}_{k}\eta_{k,i} - \Delta\mathcal{Y}_{\ell}\hat{q}'_{k,i-1} \right]
\end{aligned}
}
$$

- Using standard rounding error results, this allows us to obtain an upper bound on $\|f_{sk+j}\|$.

Attainable accuracy of s-step CG

- We can write the gap between the true and updated residuals f in terms of these errors:

$$\begin{aligned}
f_{sk+j} &= f_0 \\
&\quad - \sum_{\ell=0}^{k-1}\left[A\phi_{s\ell+s} + \psi_{s\ell+s} + \sum_{i=1}^{s}\left[A\hat{\mathcal{Y}}_\ell \xi_{\ell,i} + \hat{\mathcal{Y}}_\ell \eta_{\ell,i} - \Delta\mathcal{Y}_\ell \hat{q}'_{\ell,i-1}\right]\right] \\
&\quad - A\phi_{sk+j} - \psi_{sk+j} - \sum_{i=1}^{j}\left[A\hat{\mathcal{Y}}_k \xi_{k,i} + \hat{\mathcal{Y}}_k \eta_{k,i} - \Delta\mathcal{Y}_\ell \hat{q}'_{k,i-1}\right]
\end{aligned}$$

- Using standard rounding error results, this allows us to obtain an upper bound on $\|f_{sk+j}\|$.

Attainable accuracy of s-step CG

- We can write the gap between the true and updated residuals f in terms of these errors:

$$f_{sk+j} = f_0$$

$$-\sum_{\ell=0}^{k-1}\left[A\phi_{s\ell+s} + \psi_{s\ell+s} + \sum_{i=1}^{s}\left[A\hat{\mathcal{Y}}_\ell\xi_{\ell,i} + \hat{\mathcal{Y}}_\ell\eta_{\ell,i} - \Delta\mathcal{Y}_\ell\hat{q}'_{\ell,i-1}\right]\right]$$

$$-A\phi_{sk+j} - \psi_{sk+j} - \sum_{i=1}^{j}\left[A\hat{\mathcal{Y}}_k\xi_{k,i} + \hat{\mathcal{Y}}_k\eta_{k,i} - \Delta\mathcal{Y}_\ell\hat{q}'_{k,i-1}\right]$$

- Using standard rounding error results, this allows us to obtain an upper bound on $\|f_{sk+j}\|$.

Attainable accuracy of s-step CG

- We can write the gap between the true and updated residuals f in terms of these errors:

$$f_{sk+j} = f_0$$

$$-\sum_{\ell=0}^{k-1}\left[A\phi_{s\ell+s} + \psi_{s\ell+s} + \sum_{i=1}^{s}\left[A\hat{\mathcal{Y}}_\ell\xi_{\ell,i} + \hat{\mathcal{Y}}_\ell\eta_{\ell,i} - \Delta\mathcal{Y}_\ell\hat{q}'_{\ell,i-1}\right]\right]$$

$$-A\phi_{sk+j} - \psi_{sk+j} - \sum_{i=1}^{j}\left[A\hat{\mathcal{Y}}_k\xi_{k,i} + \hat{\mathcal{Y}}_k\eta_{k,i} - \Delta\mathcal{Y}_\ell\hat{q}'_{k,i-1}\right]$$

- Using standard rounding error results, this allows us to obtain an upper bound on $\|f_{sk+j}\|$.

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: $i \equiv sk + j$

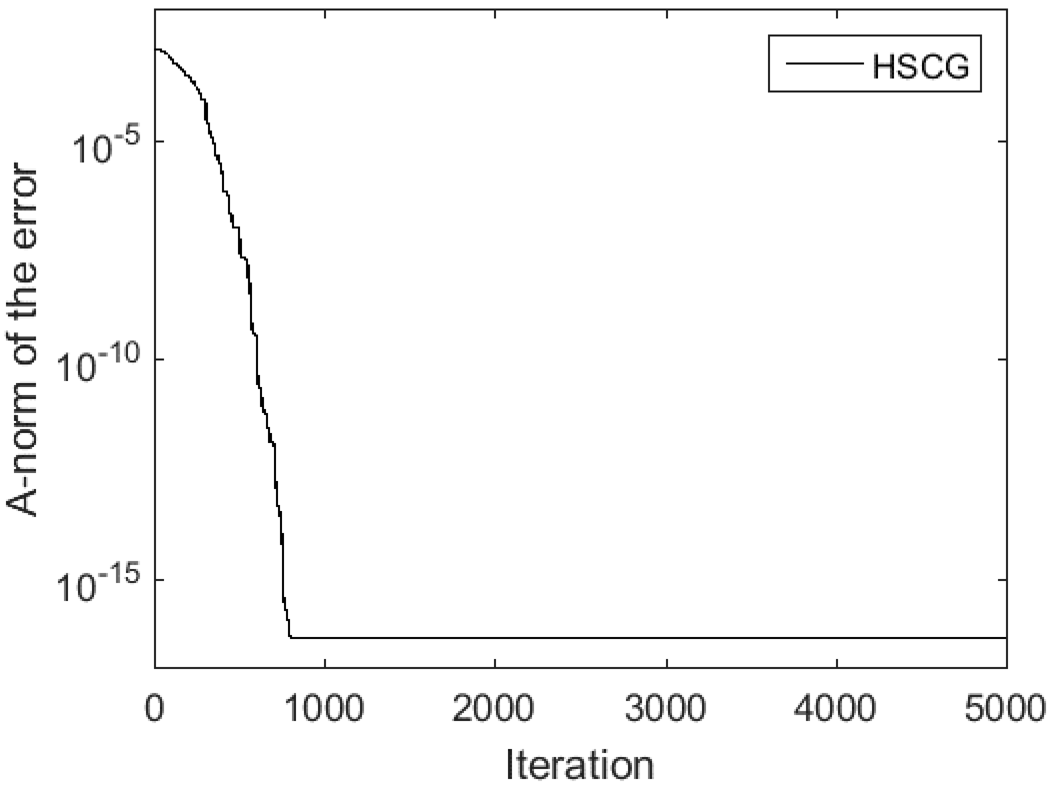
$$\|f_{sk+j}\| \leq \|f_0\| + \varepsilon \boldsymbol{c}\boldsymbol{\bar{\Gamma}_k} \sum_{m=1}^{sk+j} (1+N)\|A\|\|\hat{x}_m\| + \|\hat{r}_m\|$$

where c is a low-degree polynomial in s, and

$$\bar{\Gamma}_k = \max_{\ell \leq k} \Gamma_\ell \;, \qquad \text{where} \qquad \Gamma_\ell = \|\hat{\mathcal{Y}}_\ell^+\| \cdot \||\hat{\mathcal{Y}}_\ell|\|$$

(see C., 2015)

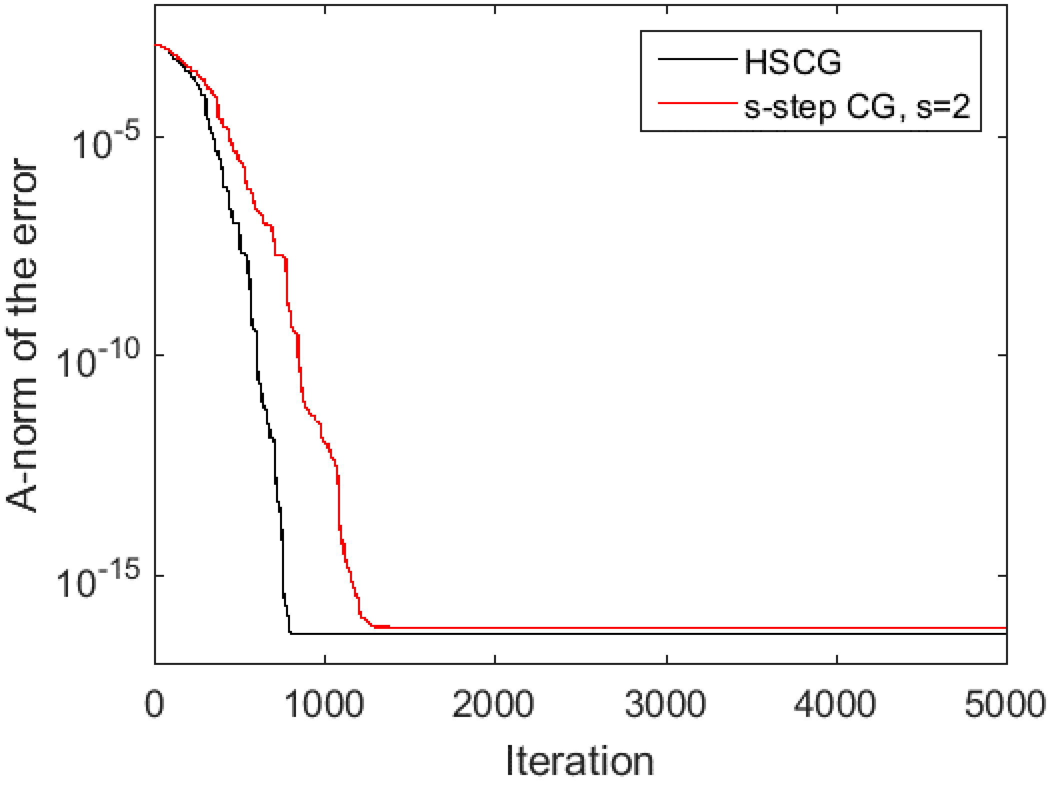
s-step CG



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

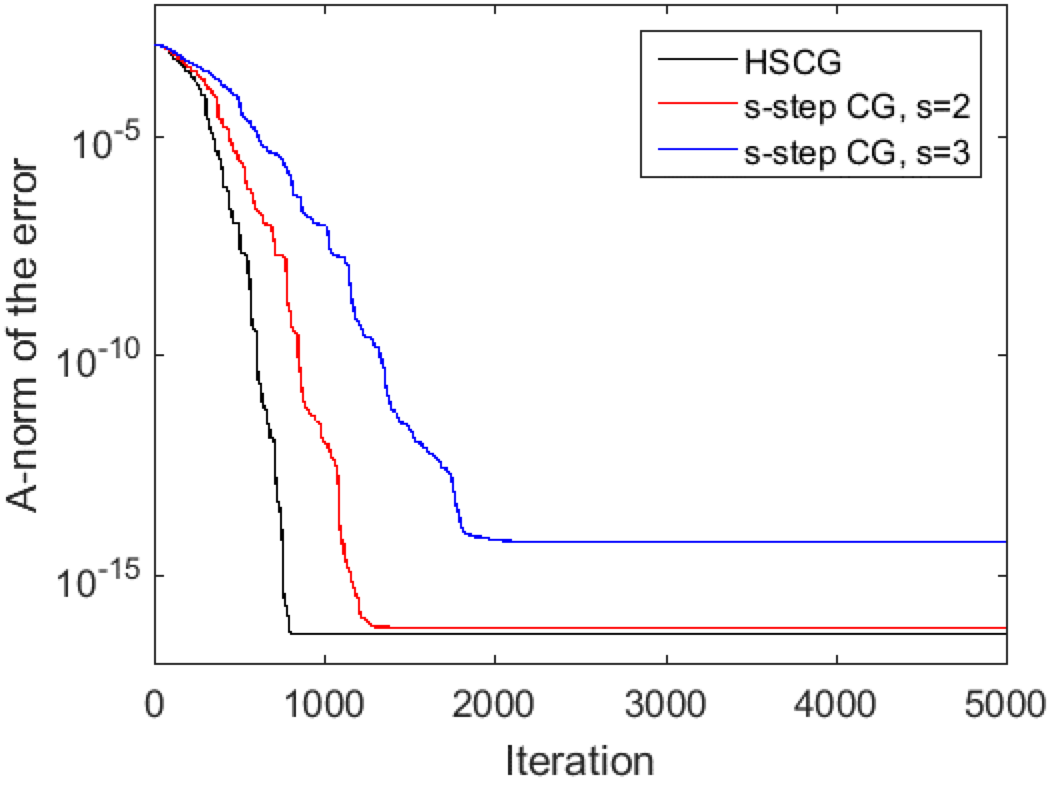
s-step CG

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



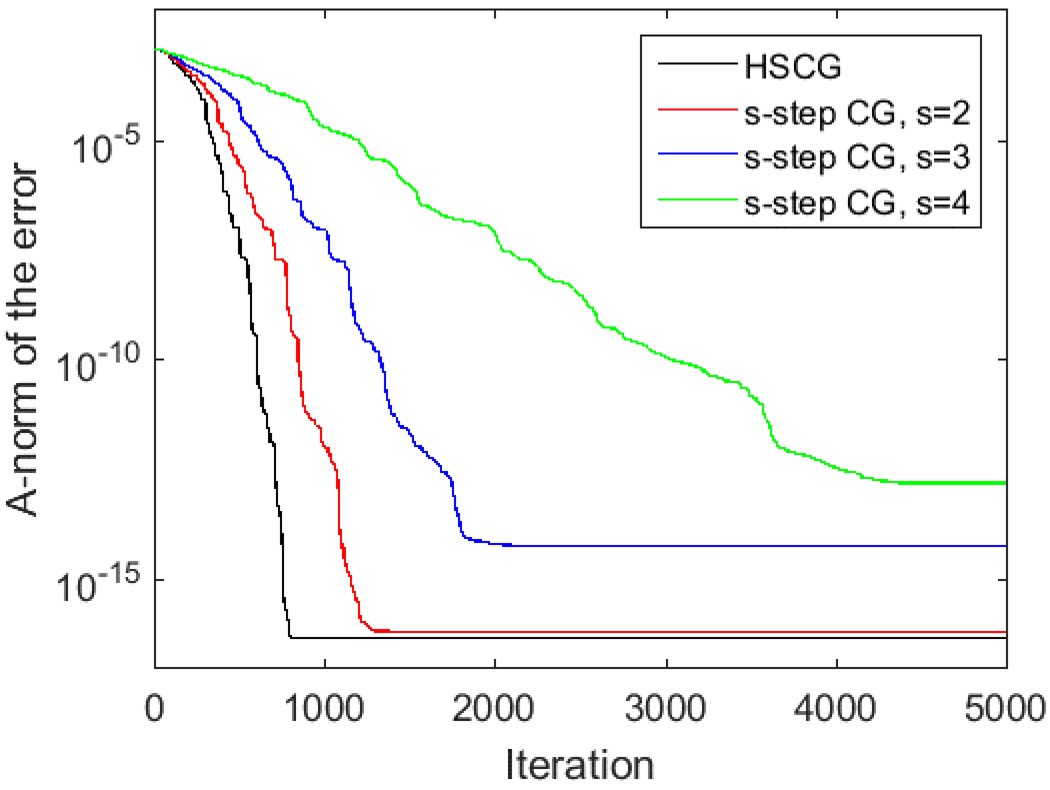
s-step CG

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



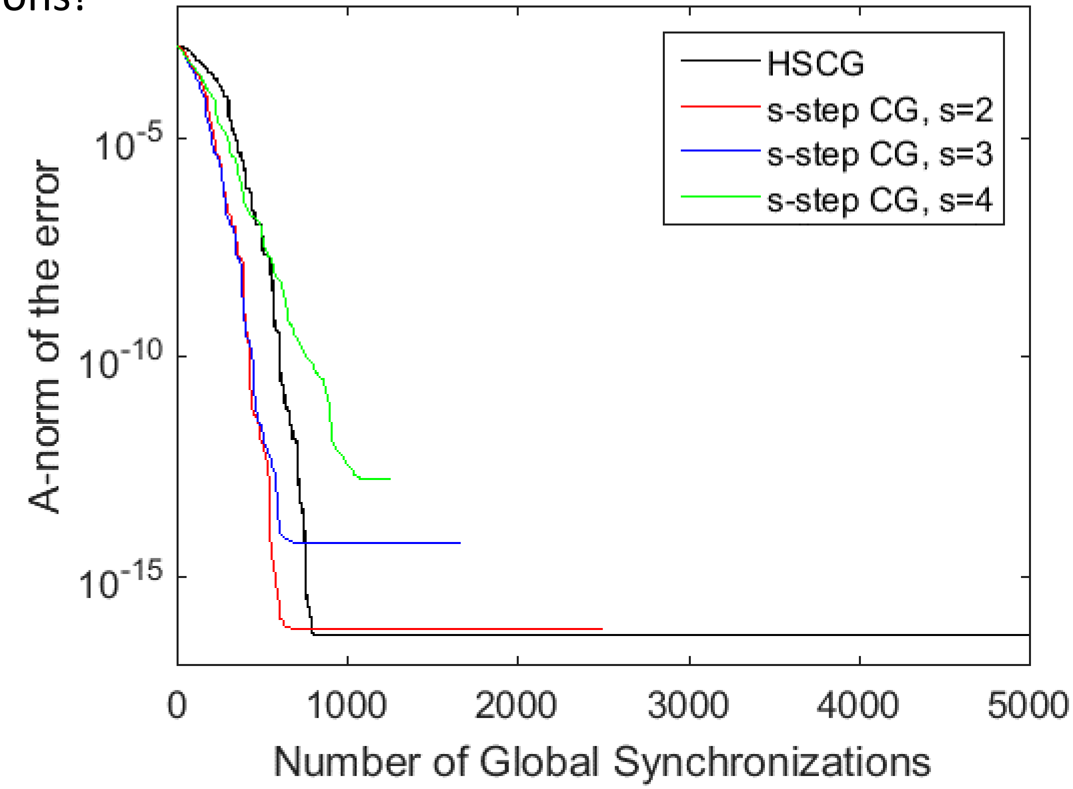
s-step CG

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



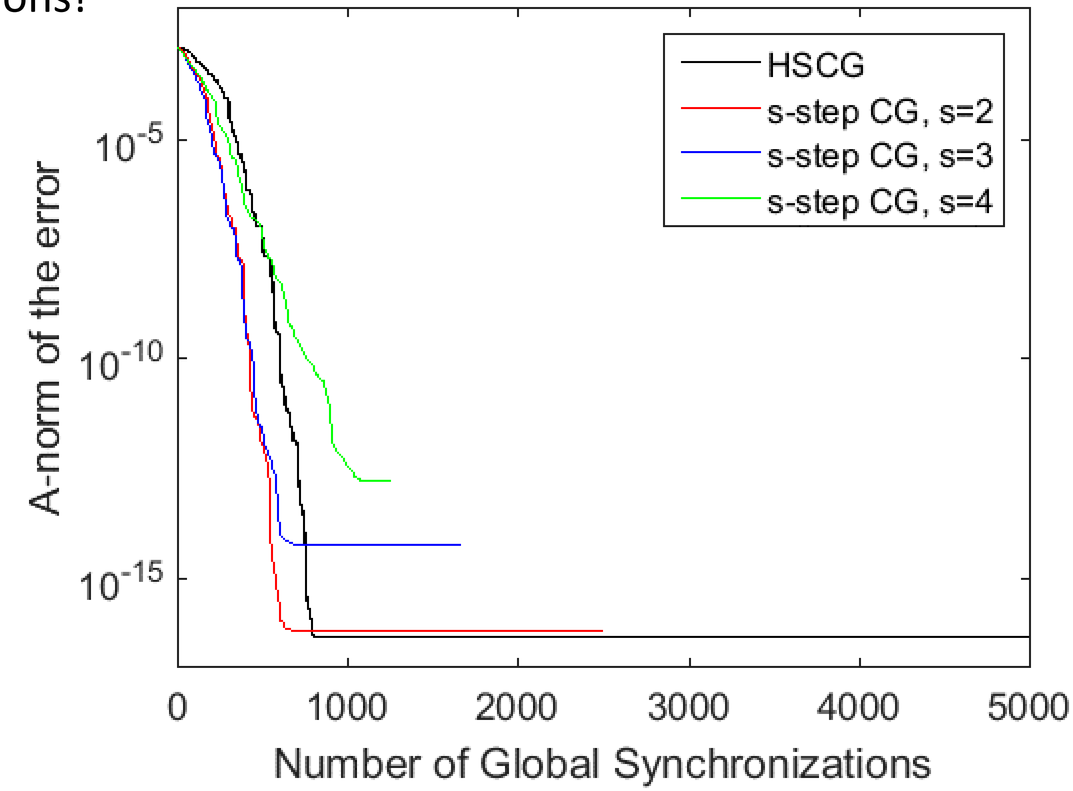
s-step CG

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!



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!



⇒ Can use other, more well-conditioned bases to improve convergence rate and accuracy (see, e.g. Philippe and Reichel, 2012).

Choosing a Polynomial Basis

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

Choosing a Polynomial Basis

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

- Simple loop unrolling gives monomial basis, e.g., $\mathcal{Y}_k = [p_m, Ap_m, \dots, A^s p_m]$
  - 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., $\mathcal{K}_{s+1}(A, p_i) = \text{span}\{p_i, Ap_i, \dots, A^s p_i\}$

- Simple loop unrolling gives monomial basis, e.g., $\mathcal{Y}_k = [p_m, Ap_m, \dots, A^s p_m]$
    - 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., $\mathcal{K}_{s+1}(A, p_i) = \text{span}\{p_i, Ap_i, \dots, A^s p_i\}$

- Simple loop unrolling gives monomial basis, e.g., $\mathcal{Y}_k = [p_m, Ap_m, \dots, A^s p_m]$
  - 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**

Better conditioned bases

- The Newton basis:

$$\{v, (A-\theta_1)v, (A-\theta_2)(A-\theta_1)v, \dots, (A-\theta_s)\cdots(A-\theta_1)v\}$$

  where $\{\theta_1, \dots, \theta_s\}$ are approximate eigenvalues of A, ordered according to Leja ordering

  – In practice: recover Ritz values from the first few iterations, iteratively refine eigenvalue estimates to improve basis

  – Used by many to improve s-step variants: e.g., Bai, Hu, and Reichel (1991), Erhel (1995), Hoemmen (2010)

Better conditioned bases

- The Newton basis:

$$\{v, (A-\theta_1)v, (A-\theta_2)(A-\theta_1)v, \dots, (A-\theta_s)\cdots(A-\theta_1)v\}$$

  where $\{\theta_1, \dots, \theta_s\}$ are approximate eigenvalues of A, ordered according to Leja ordering

  – In practice: recover Ritz values from the first few iterations, iteratively refine eigenvalue estimates to improve basis

  – Used by many to improve s-step variants: e.g., Bai, Hu, and Reichel (1991), Erhel (1995), Hoemmen (2010)

- Chebyshev basis: given ellipse enclosing spectrum of A with foci at $d \pm c$, we can generate the scaled and shifted Chebyshev polynomials as:

$$\tilde{\tau}_j(z) = \left(\tau_j\left(\frac{d-z}{c}\right)\right) \Big/ \left(\tau_j\left(\frac{d}{c}\right)\right)$$

  where $\{\tau_j\}_{j\geq 0}$ are the Chebyshev polynomials of the first kind

  – In practice: estimate d and c parameters from Ritz values recovered from the first few iterations

  – Used by many to improve s-step variants: e.g., de Sturler (1991), Joubert and Carey (1992), de Sturler and van der Vorst (1995)

"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 HPC 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 $i \in \{1, \ldots, m\}$,

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

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

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)

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

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

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

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

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

Lanczos [Paige, 1976]

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

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

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

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

The amplification term

- Roundoff errors in s-step variant follow same pattern as classical variant, but amplified by factor of Γ or Γ^2
    - Theoretically confirms empirical observations on importance of basis conditioning (dating back to late ‘80s)

The amplification term

- Roundoff errors in s-step variant follow same pattern as classical variant, but amplified by factor of Γ or Γ^2
  - Theoretically confirms empirical observations on importance of basis conditioning (dating back to late ‘80s)

- Using the definition

$$\Gamma \equiv \Gamma_k = \max_{\ell \leq k} \|\mathcal{Y}_\ell^+\| \cdot \||\mathcal{Y}_\ell|\|$$

  gives simple, but loose bounds

- What we really need: $\||\mathcal{Y}||y'|\| \leq \Gamma \|\mathcal{Y}y'\|$ to hold for the computed basis $\mathcal{Y}$ and coordinate vector y' in every bound.

The amplification term

- Roundoff errors in s-step variant follow same pattern as classical variant, but amplified by factor of Γ or Γ^2
    - **Theoretically confirms empirical observations** on importance of basis conditioning (dating back to late ‘80s)

- Using the definition
$$\Gamma \equiv \Gamma_k = \max_{\ell \leq k} \|\mathcal{Y}_\ell^+\| \cdot \||\mathcal{Y}_\ell|\|$$
gives simple, but loose bounds

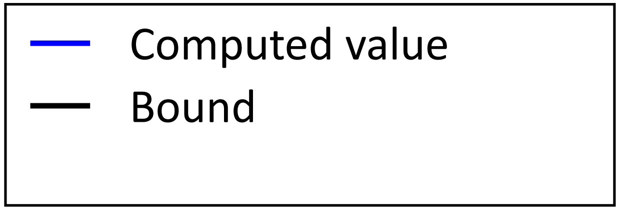
- What we really need: $\||\mathcal{Y}||y'|\| \leq \Gamma \|\mathcal{Y}y'\|$ to hold for the computed basis $\mathcal{Y}$ and coordinate vector y' in every bound.

- **Alternate definition of Γ gives tighter bounds**; requires light bookkeeping

- Example: for bounds on $\hat{\beta}_{i+1}\left|\hat{v}_i^T \hat{v}_{i+1}\right|$ and $\left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right|$, we can use the definition

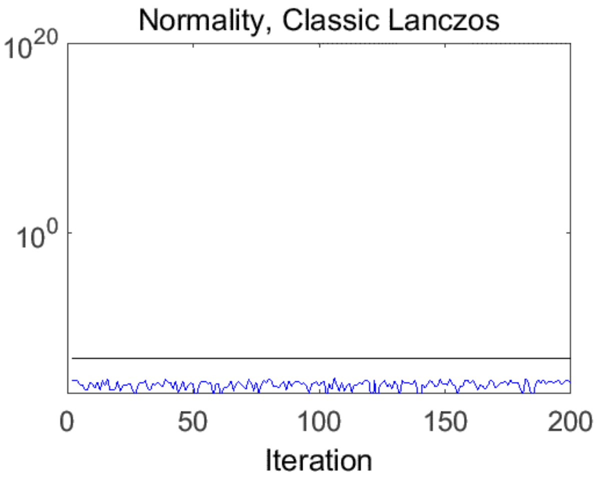
$$\Gamma_{k,j} \equiv \max_{x \in \{\hat{w}'_{k,j}, \hat{u}'_{k,j}, \hat{v}'_{k,j}, \hat{v}'_{k,j-1}\}} \frac{\||\hat{\mathcal{Y}}_k||x|\|}{\|\hat{\mathcal{Y}}_k x\|}$$

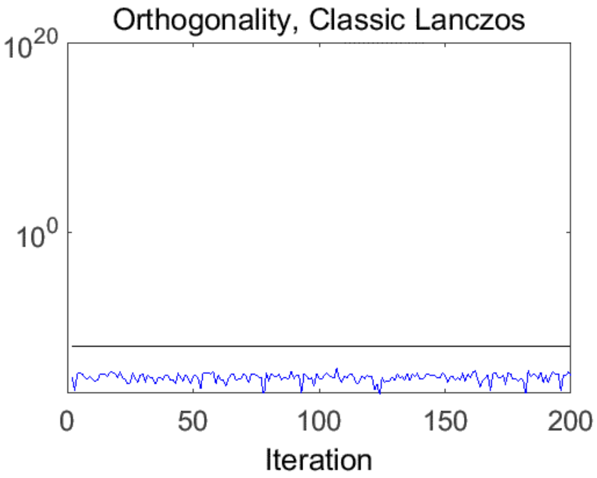
Problem: 2D Poisson,
$n = 256$,
random starting vector



$$\left|\hat{v}_{i+1}^T\hat{v}_{i+1} - 1\right| \le \varepsilon_0/2$$

$$\hat{\beta}_{i+1}\left|\hat{v}_i^T\hat{v}_{i+1}\right| \le 2\varepsilon_0\sigma$$





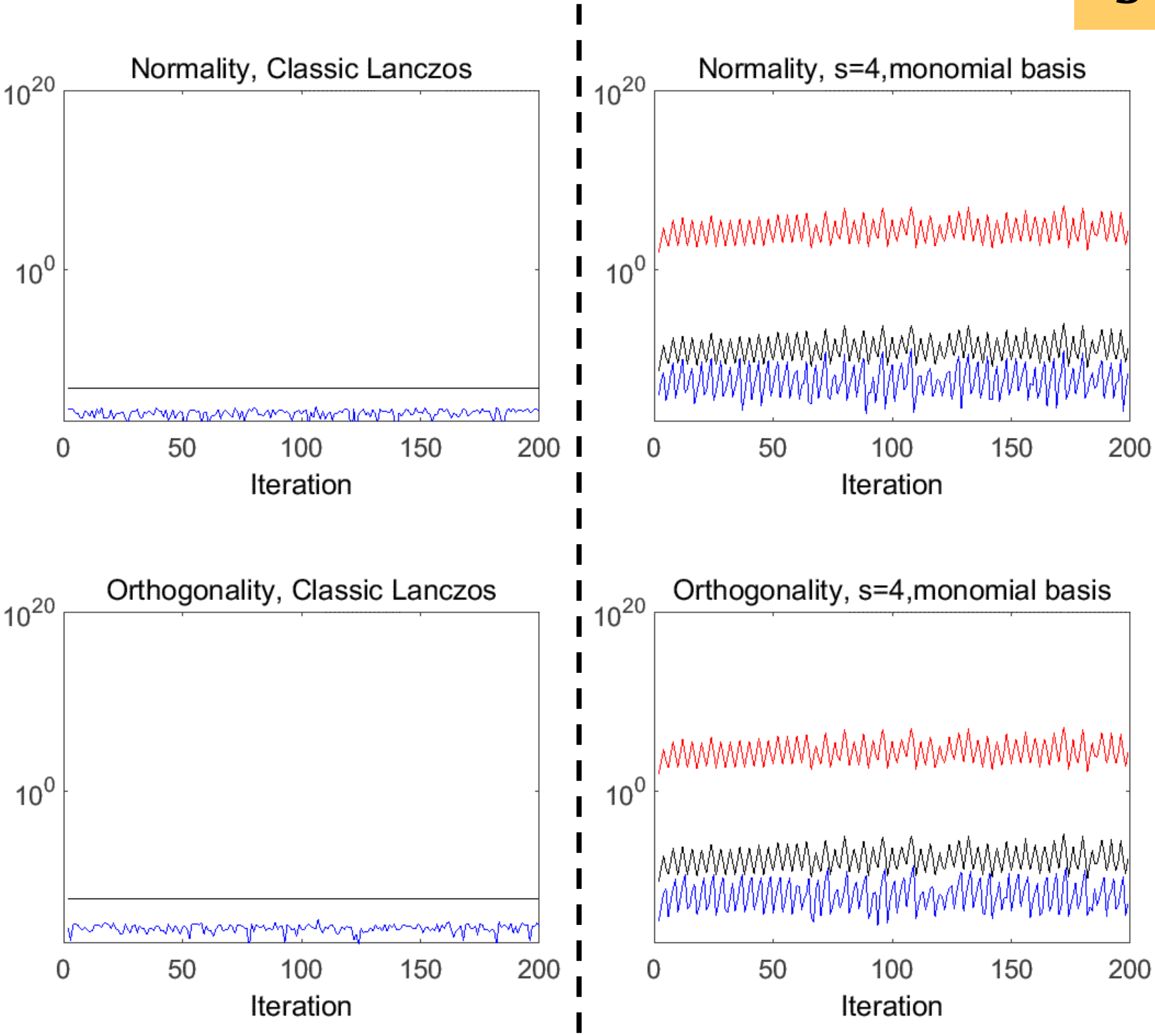
Problem: 2D Poisson,
$n = 256$,
random starting vector

🟦	Computed value
⬛	Bound
🟥	Amplification factor $\Gamma_{k,j}^2$

$$\left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right| \le \varepsilon_0/2$$

$$\hat{\beta}_{i+1}\left|\hat{v}_i^T \hat{v}_{i+1}\right| \le 2\varepsilon_0\sigma$$

$\boldsymbol{s = 4}$



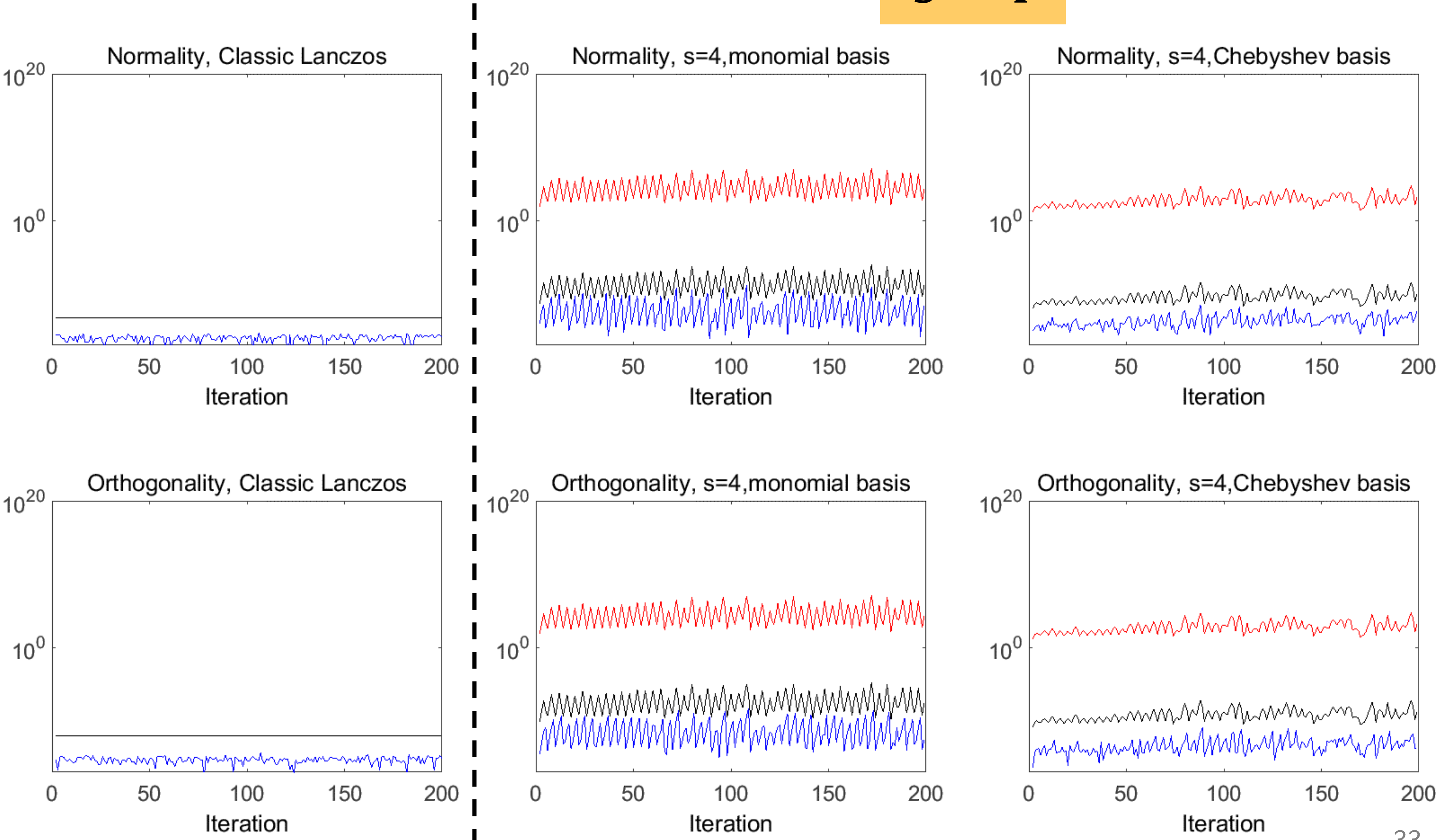
Problem: 2D Poisson,
$n = 256$,
random starting vector

— Computed value
— Bound
— Amplification factor $\Gamma_{k,j}^2$

$$\left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right| \leq \varepsilon_0/2$$

$$\hat{\beta}_{i+1}\left|\hat{v}_i^T \hat{v}_{i+1}\right| \leq 2\varepsilon_0\sigma$$

$s = 4$



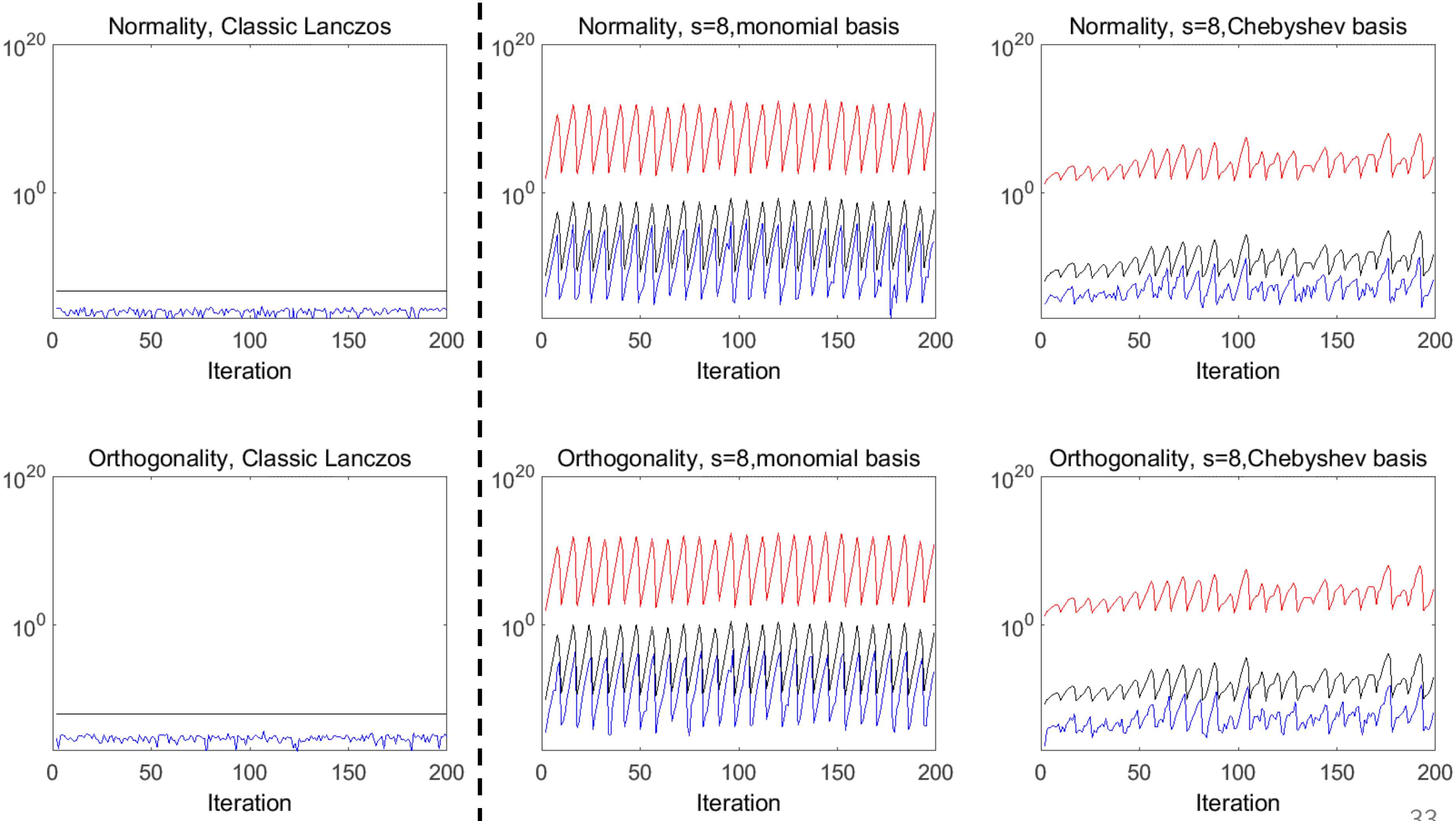
Problem: 2D Poisson,
$n = 256$,
random starting vector

— Computed value
— Bound
— Amplification factor $\Gamma_{k,j}^2$

$$\left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right| \le \varepsilon_0/2$$

$$\hat{\beta}_{i+1}\left|\hat{v}_i^T \hat{v}_{i+1}\right| \le 2\varepsilon_0\sigma$$

$\boldsymbol{s = 8}$



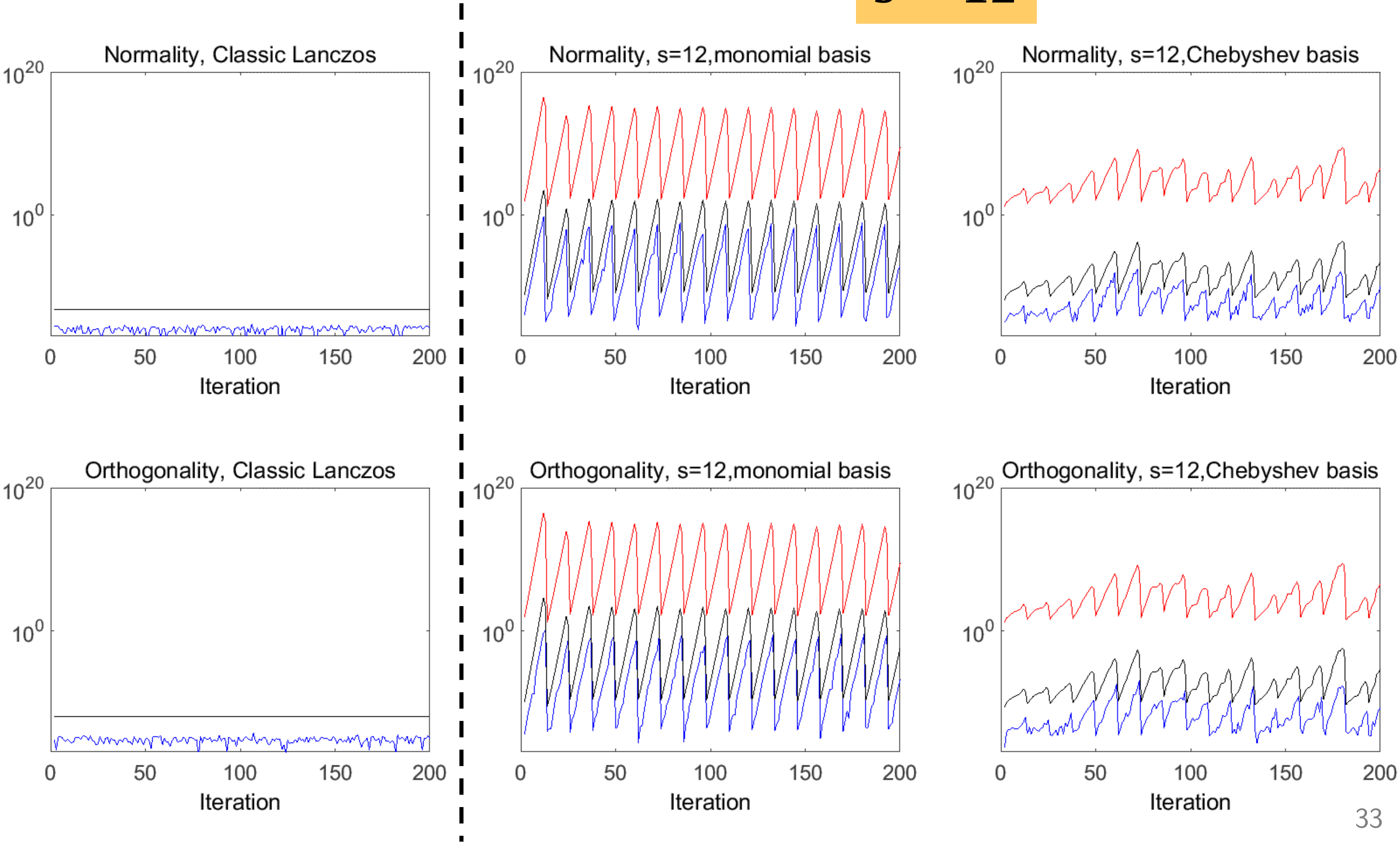
Problem: 2D Poisson,
$n = 256$,
random starting vector

- Computed value
- Bound
- Amplification factor $\Gamma_{k,j}^2$

$$\left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right| \leq \varepsilon_0/2$$

$$\hat{\beta}_{i+1}\left|\hat{v}_i^T \hat{v}_{i+1}\right| \leq 2\varepsilon_0\sigma$$

$\boldsymbol{s = 12}$



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

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

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

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

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

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

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

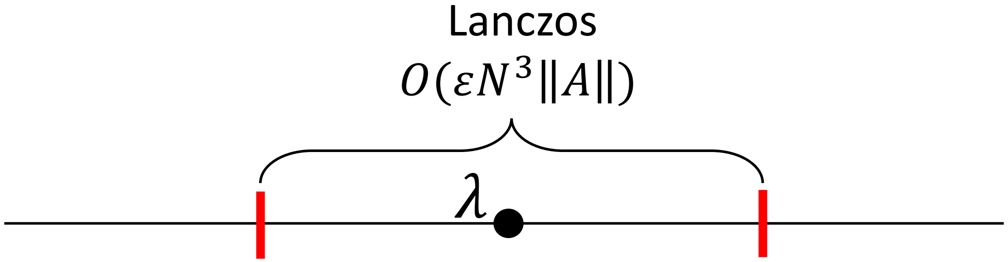
Convergence of Ritz Values in s-step Lanczos

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

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

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

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


Lanczos

$O(\varepsilon N^3 \|A\|)$

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

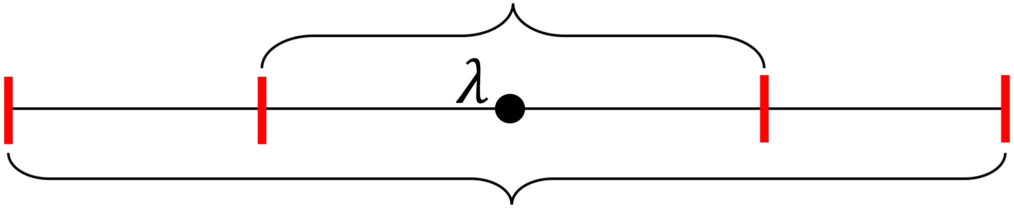
$$\left(\Gamma = c \cdot \max_{\ell \le k} \|\hat{\mathcal{Y}}_\ell^+\| \, \||\hat{\mathcal{Y}}_\ell|\|\right)$$

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

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

Lanczos

$O(\varepsilon N^3 \|A\|)$



$O(\varepsilon N^3 \|A\| \mathbf{\Gamma^2})$

s-step Lanczos

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

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

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

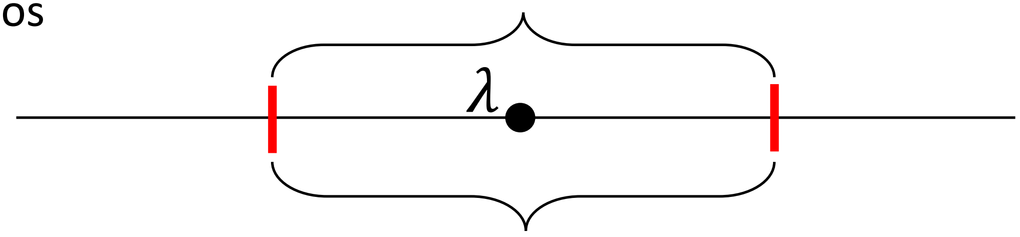
- Bounds on accuracy of Ritz values depend on Γ^2

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

s-step Lanczos behaves the same numerically as classical Lanczos

Lanczos

$O(\varepsilon N^3\|A\|)$



$O(\varepsilon N^3\|A\|)$

s-step Lanczos

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

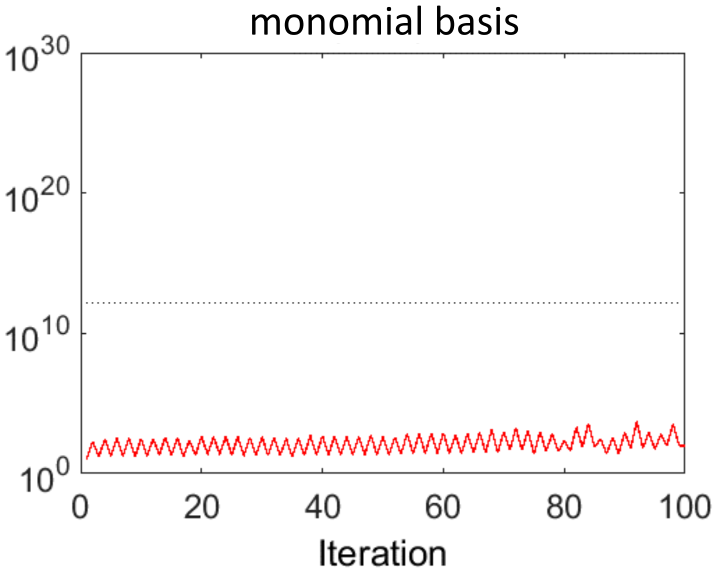
Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

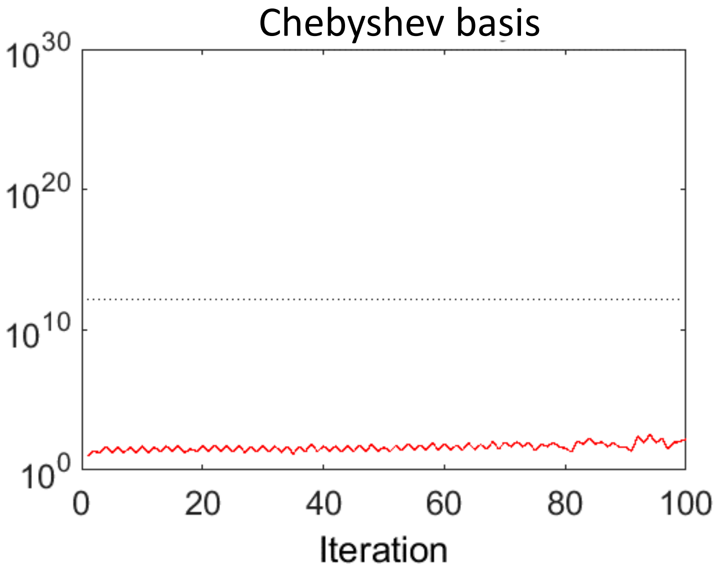
$$\boldsymbol{s = 2}$$

Top plots:

— Computed $\Gamma_{k,j}^2$

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





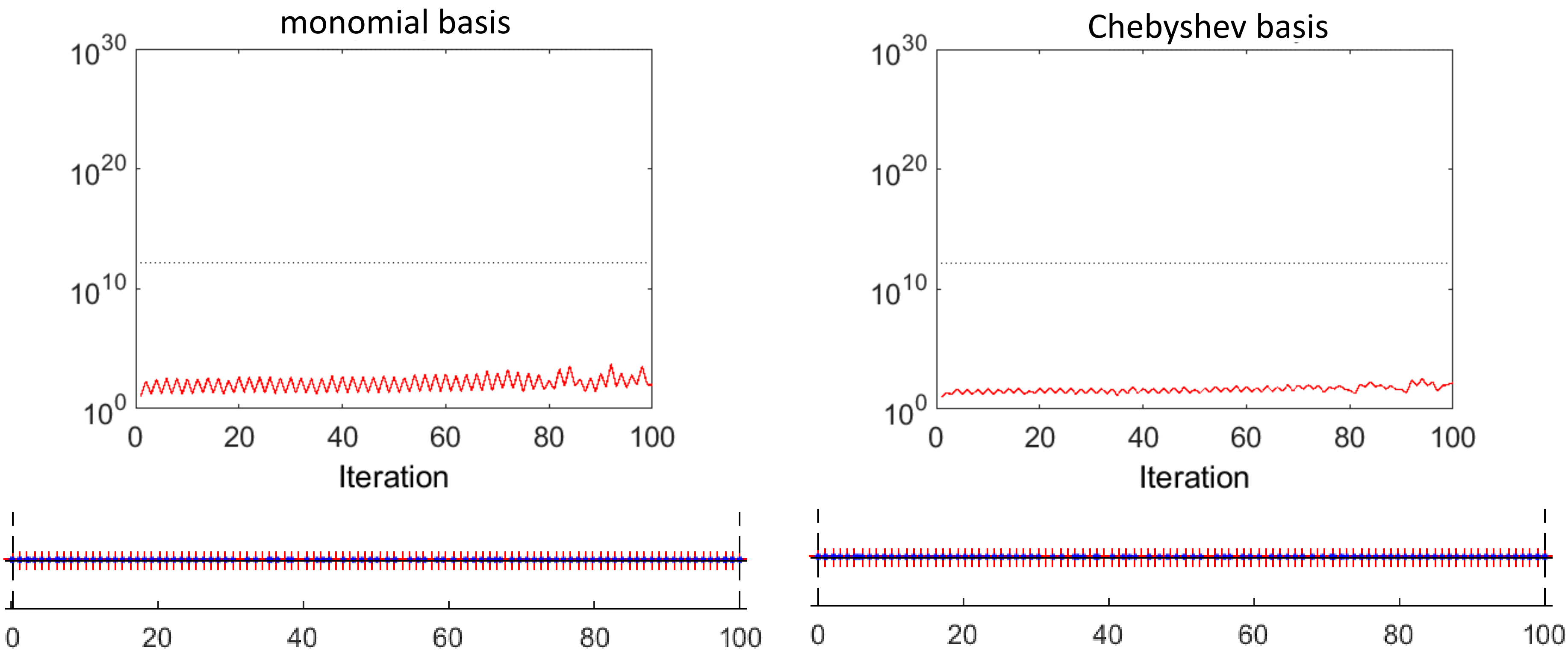
Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

$$\boldsymbol{s = 2}$$

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

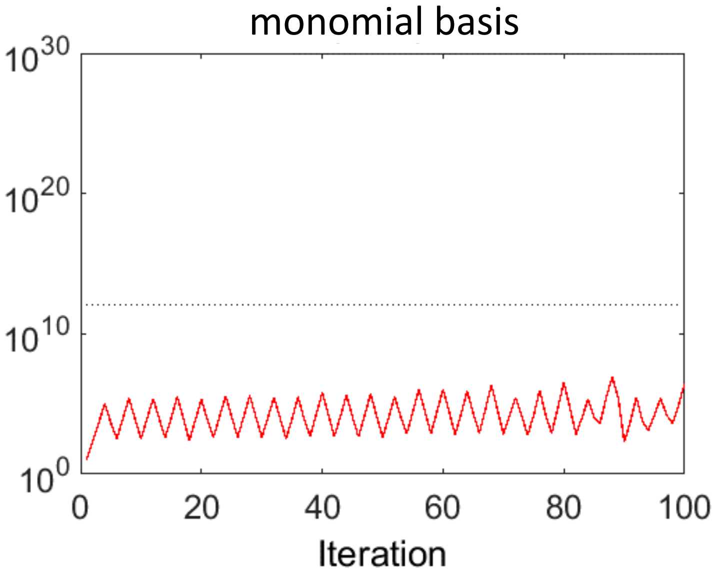
Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

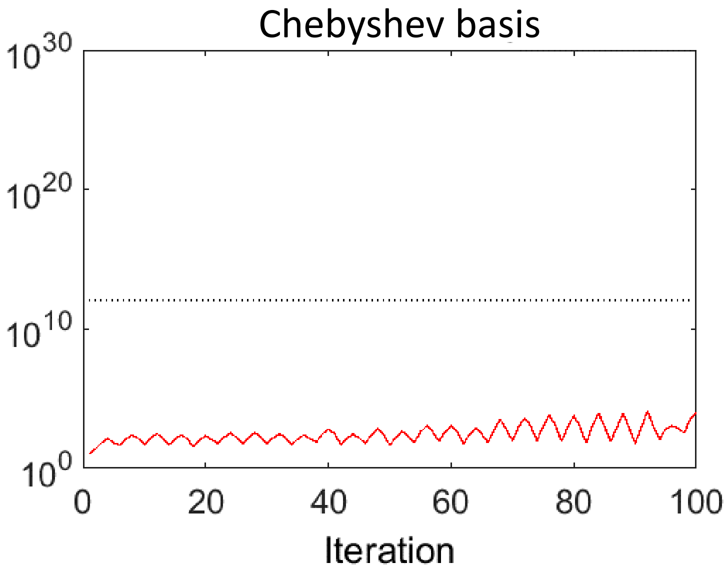
$\boldsymbol{s = 4}$

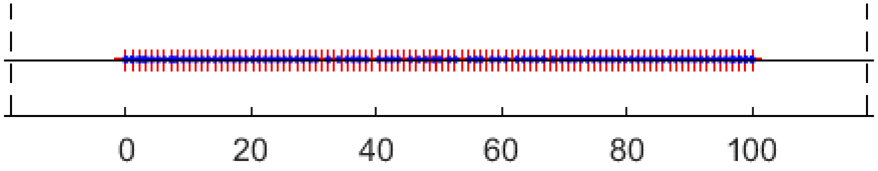
Top plots:

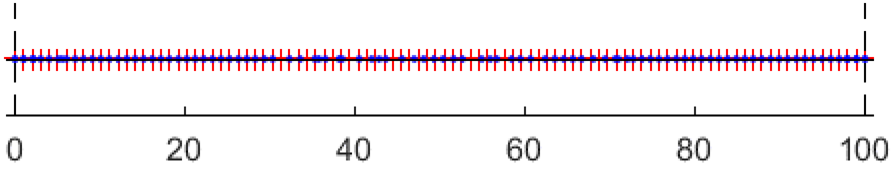
— Computed $\Gamma_{k,j}^2$

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









Bottom Plots:

+ True eigenvalues

● Computed Ritz values

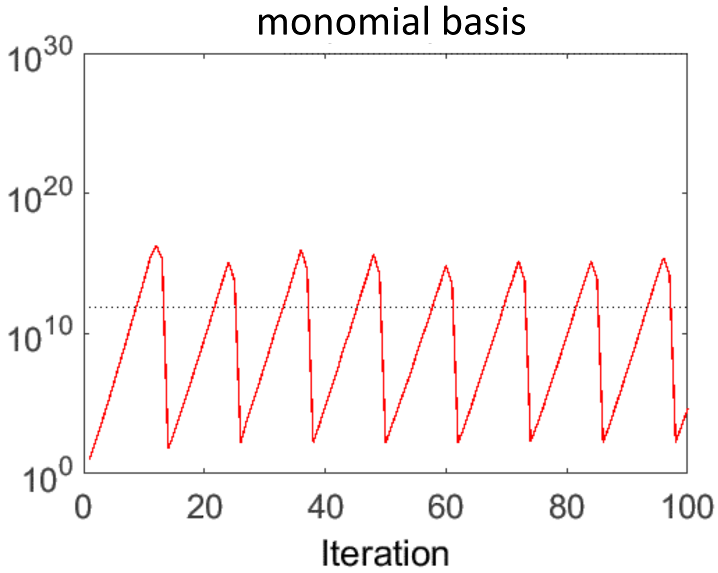
¦ Bounds on range of computed Ritz values

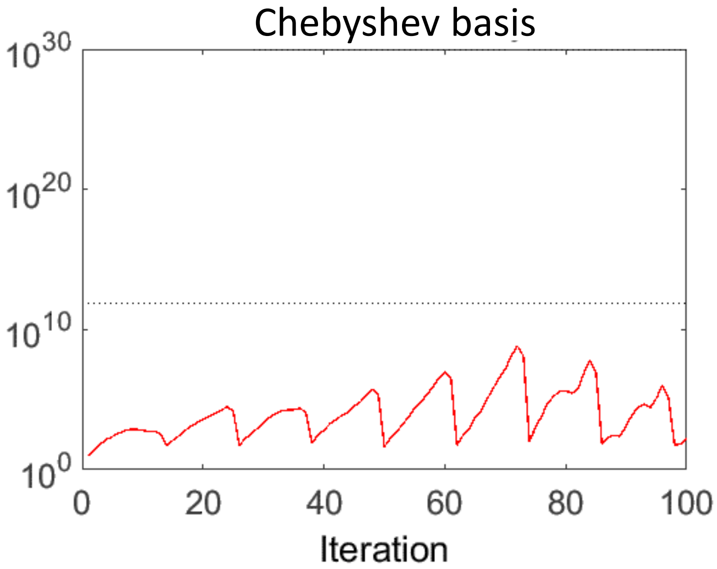
Problem: Diagonal matrix with $n = 100$ with evenly spaced eigenvalues between $\lambda_{min} = 0.1$ and $\lambda_{max} = 100$; random starting vector

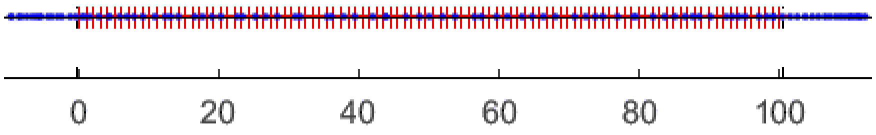
$\boldsymbol{s = 12}$

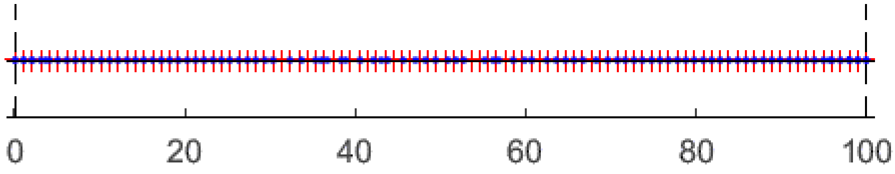
Top plots:

—	Computed $\Gamma_{k,j}^2$
┈	$(24(\varepsilon(n+11s+15))^{-1}$







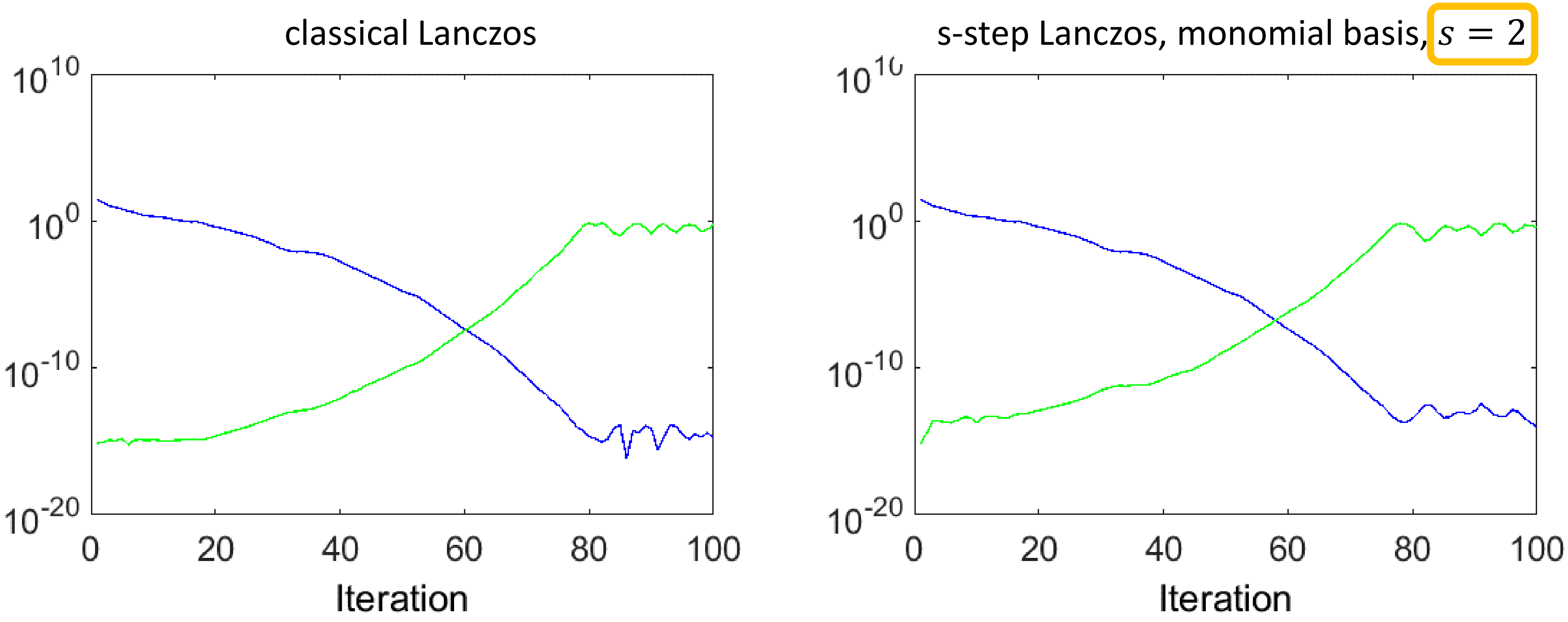


Bottom Plots:

+	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, $\boxed{s = 2}$

Measure of loss of orthogonality ⟵

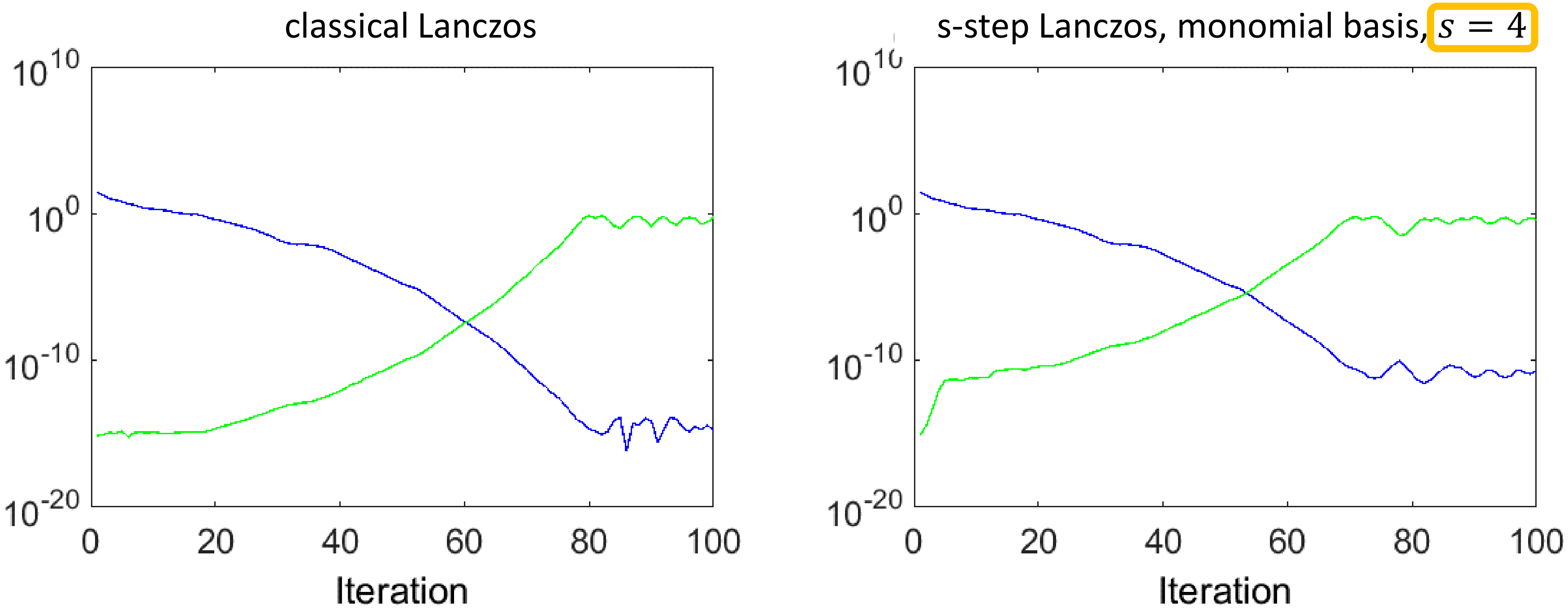
🟩 $\max_i |z_i^{(m)T} \hat{v}_{m+1}|$

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

Measure of Ritz value convergence ⟶

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

$\Gamma \lesssim 3 \times 10^3$

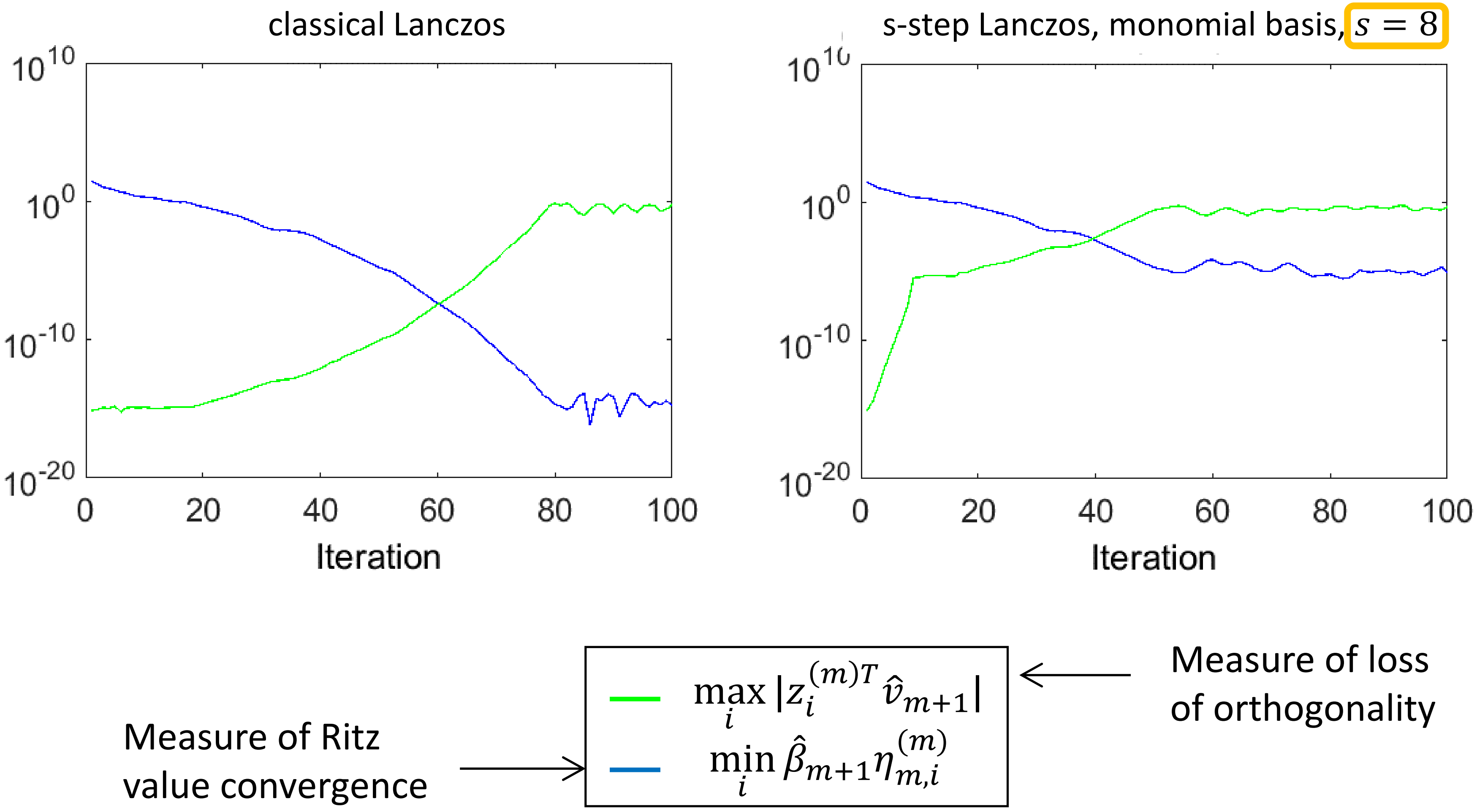


🟩 $\max_i |z_i^{(m)T} \hat{v}_{m+1}|$ ← Measure of loss of orthogonality

Measure of Ritz value convergence → 🟦 $\min_i \hat{\beta}_{m+1} \eta_{m,i}^{(m)}$

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

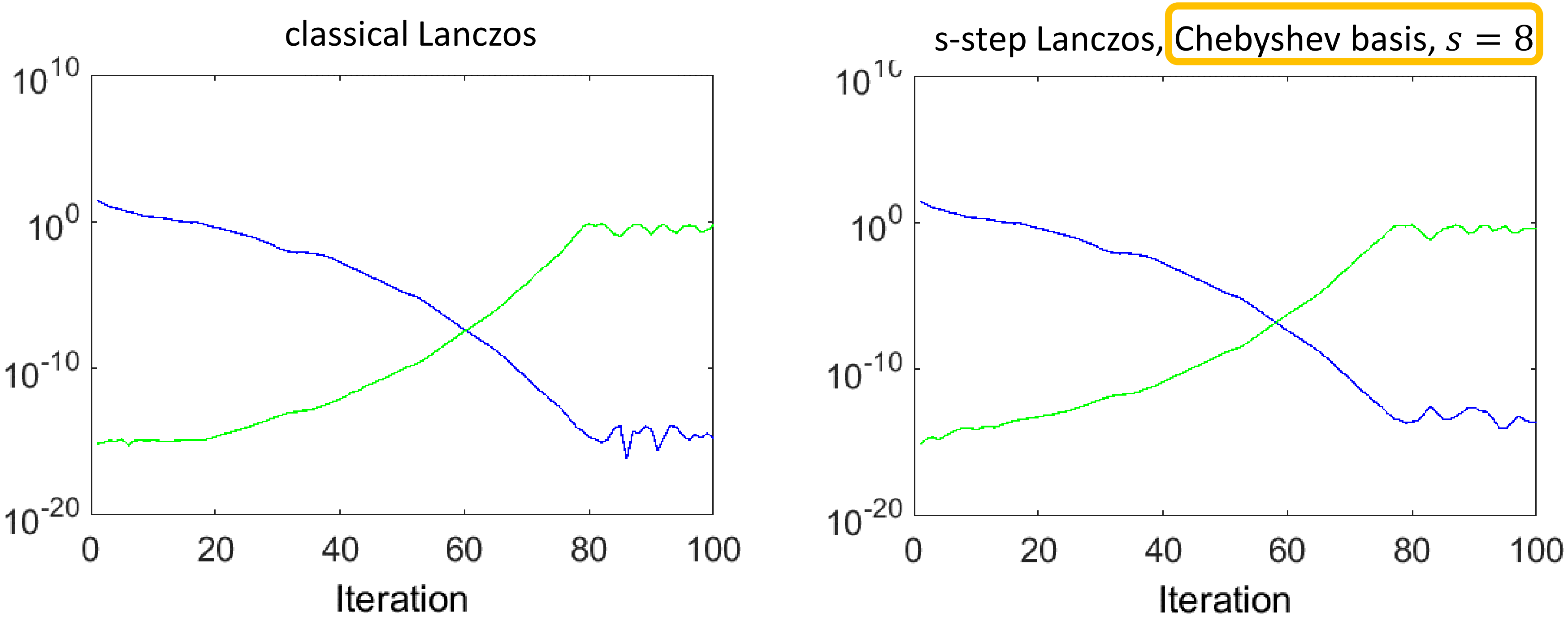
s-step Lanczos, monomial basis, $\boxed{s = 8}$

Measure of loss of orthogonality

Measure of Ritz value convergence

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

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



🟩 $\max_i |z_i^{(m)T} \hat{v}_{m+1}|$ ← Measure of loss of orthogonality

Measure of Ritz value convergence → 🟦 $\min_i \hat{\beta}_{m+1} \eta_{m,i}^{(m)}$

Towards understanding convergence delay

- Coefficients α and β (related to entries of T_i) determine distribution functions $\omega^{(i)}(\lambda)$ which approximate distribution function $\omega(\lambda)$ determined by inputs A, b, x_0 in terms of the ith Gauss-Christoffel quadrature
- CG method = matrix formulation of Gauss-Christoffel quadrature (see, e.g., [Liesen & Strakoš, 2013])
- A-norm of CG error for $f(\lambda) = \lambda^{-1}$ given as scaled quadrature error

$$\int \lambda^{-1} d\omega(\lambda) = \sum_{\ell=1}^{i} \omega_\ell^{(i)} \left\{\theta_\ell^{(i)}\right\}^{-1} + \frac{\|x - x_i\|_A^2}{\|r_0\|^2}$$

Towards understanding convergence delay

- Coefficients α and β (related to entries of T_i) determine distribution functions $\omega^{(i)}(\lambda)$ which approximate distribution function $\omega(\lambda)$ determined by inputs A, b, x_0 in terms of the ith Gauss-Christoffel quadrature
- CG method = matrix formulation of Gauss-Christoffel quadrature (see, e.g., [Liesen & Strakoš, 2013])
- A-norm of CG error for $f(\lambda) = \lambda^{-1}$ given as scaled quadrature error

$$\int \lambda^{-1} d\omega(\lambda) = \sum_{\ell=1}^{i} \omega_\ell^{(i)} \left\{\theta_\ell^{(i)}\right\}^{-1} + \frac{\|x - x_i\|_A^2}{\|r_0\|^2}$$

- For particular CG implementation, can the computed $\widehat{\omega}^{(i)}(\lambda)$ be associated with some distribution function $\widehat{\omega}(\lambda)$ related to the distribution function $\omega(\lambda)$, i.e.,

$$\int \lambda^{-1} d\omega(\lambda) \approx \int \lambda^{-1} d\widehat{\omega}(\lambda) = \sum_{\ell=1}^{i} \widehat{\omega}_\ell^{(i)} \left\{\hat{\theta}_\ell^{(i)}\right\}^{-1} + \frac{\|x - \hat{x}_i\|_A^2}{\|r_0\|^2} + F_i$$

where F_i is small relative to error term?

Towards understanding convergence delay

- Coefficients α and β (related to entries of T_i) determine distribution functions $\omega^{(i)}(\lambda)$ which approximate distribution function $\omega(\lambda)$ determined by inputs A, b, x_0 in terms of the ith Gauss-Christoffel quadrature
- CG method = matrix formulation of Gauss-Christoffel quadrature (see, e.g., [Liesen & Strakoš, 2013])
- A-norm of CG error for $f(\lambda) = \lambda^{-1}$ given as scaled quadrature error

$$\int \lambda^{-1} d\omega(\lambda) = \sum_{\ell=1}^{i} \omega_\ell^{(i)} \left\{\theta_\ell^{(i)}\right\}^{-1} + \frac{\|x - x_i\|_A^2}{\|r_0\|^2}$$

- For particular CG implementation, can the computed $\widehat{\omega}^{(i)}(\lambda)$ be associated with some distribution function $\widehat{\omega}(\lambda)$ related to the distribution function $\omega(\lambda)$, i.e.,

$$\int \lambda^{-1} d\omega(\lambda) \approx \int \lambda^{-1} d\widehat{\omega}(\lambda) = \sum_{\ell=1}^{i} \widehat{\omega}_\ell^{(i)} \left\{\hat{\theta}_\ell^{(i)}\right\}^{-1} + \frac{\|x - \hat{x}_i\|_A^2}{\|r_0\|^2} + F_i$$

where F_i is small relative to error term?

- For classical CG, yes; proved by Greenbaum [1989]

Towards understanding convergence delay

- Coefficients α and β (related to entries of T_i) determine distribution functions $\omega^{(i)}(\lambda)$ which approximate distribution function $\omega(\lambda)$ determined by inputs A, b, x_0 in terms of the ith Gauss-Christoffel quadrature
- CG method = matrix formulation of Gauss-Christoffel quadrature (see, e.g., [Liesen & Strakoš, 2013])
- A-norm of CG error for $f(\lambda) = \lambda^{-1}$ given as scaled quadrature error

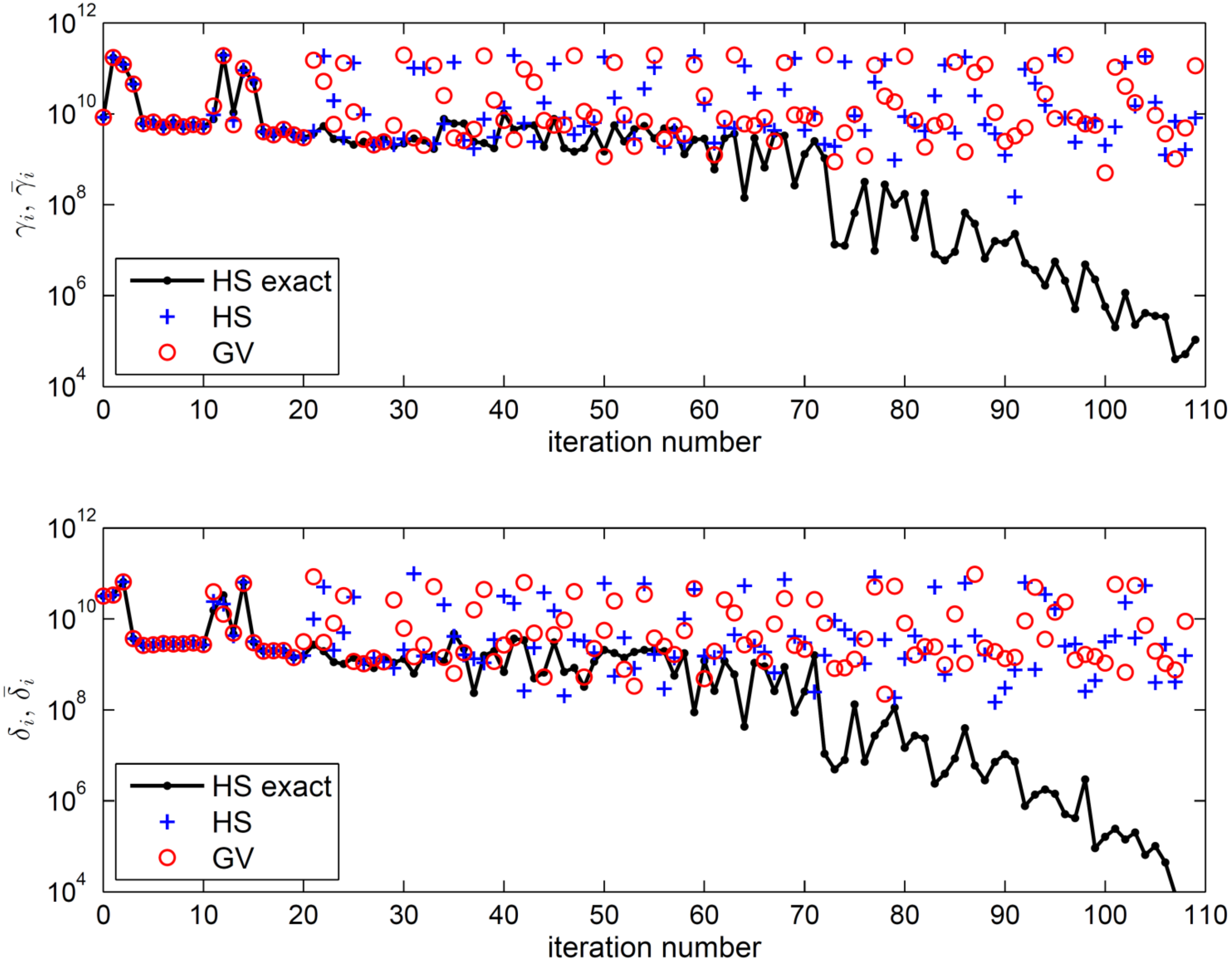
$$\int \lambda^{-1} d\omega(\lambda) = \sum_{\ell=1}^{i} \omega_\ell^{(i)} \left\{\theta_\ell^{(i)}\right\}^{-1} + \frac{\|x - x_i\|_A^2}{\|r_0\|^2}$$

- For particular CG implementation, can the computed $\widehat{\omega}^{(i)}(\lambda)$ be associated with some distribution function $\widehat{\omega}(\lambda)$ related to the distribution function $\omega(\lambda)$, i.e.,

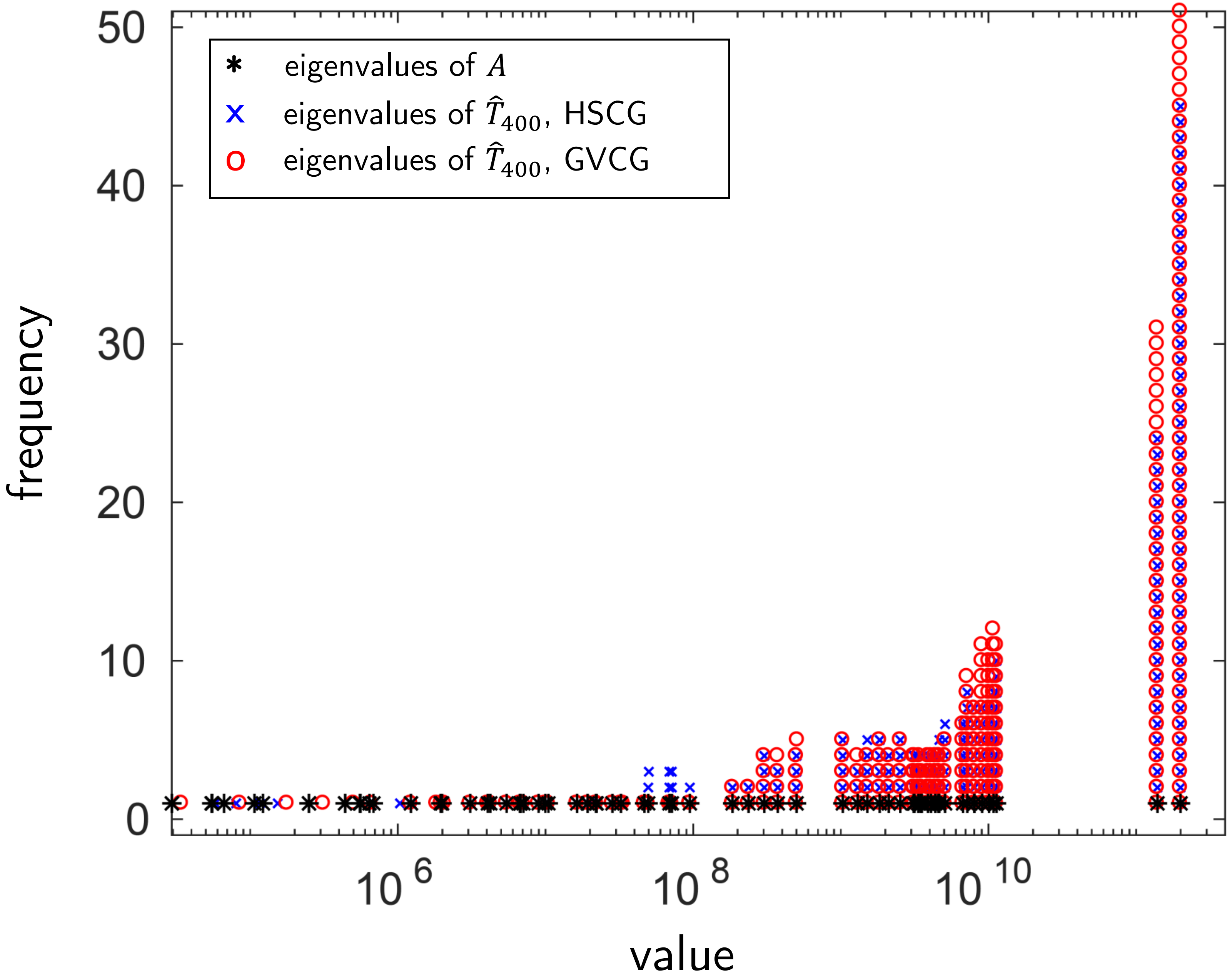
$$\int \lambda^{-1} d\omega(\lambda) \approx \int \lambda^{-1} d\widehat{\omega}(\lambda) = \sum_{\ell=1}^{i} \widehat{\omega}_\ell^{(i)} \left\{\hat{\theta}_\ell^{(i)}\right\}^{-1} + \frac{\|x - \hat{x}_i\|_A^2}{\|r_0\|^2} + F_i$$

  where F_i is small relative to error term?
- For classical CG, yes; proved by Greenbaum [1989]
- For pipelined CG and s-step CG, THOROUGH ANALYSIS NEEDED!

Differences in entries γ_i, δ_i in Jacobi matrices T_i in HSCG vs. GVCG (matrix bcsstk03)



see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]



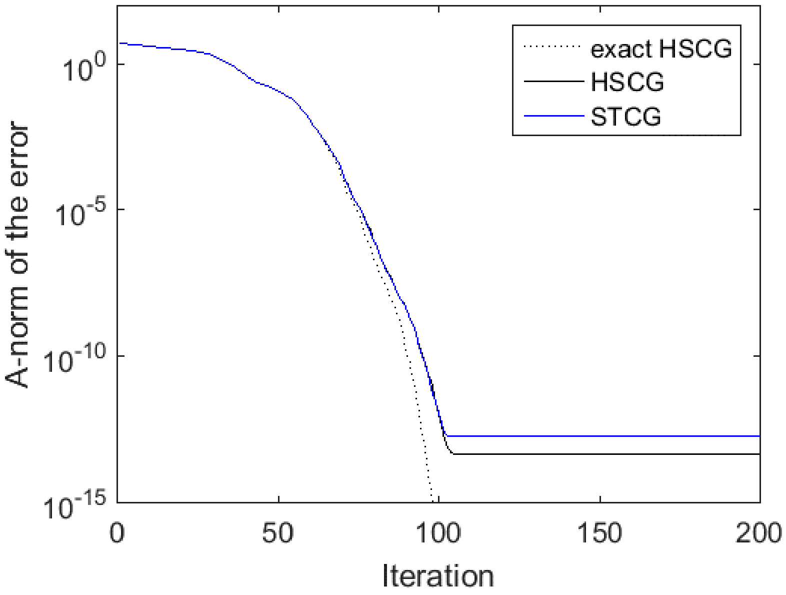
see [C., Rozložník, Strakoš, Tichý, Tůma, 2018]

A different problem…

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

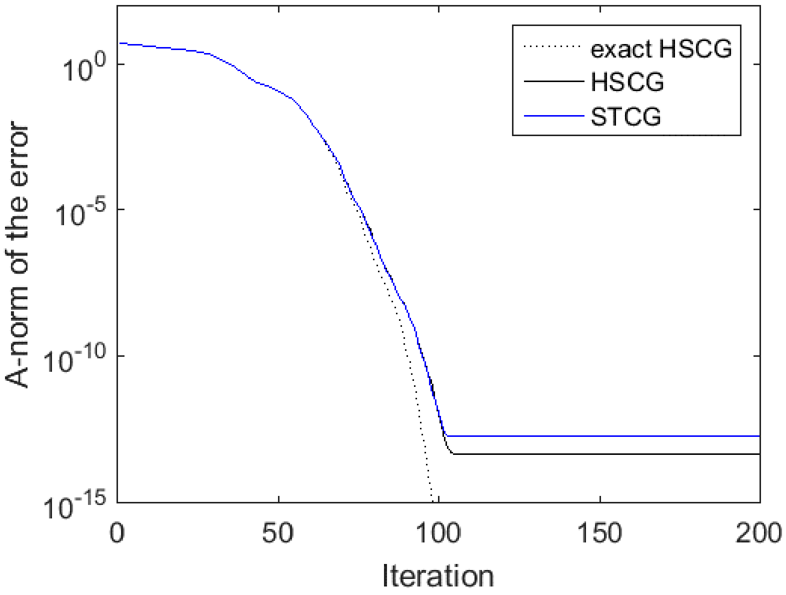
A different problem…

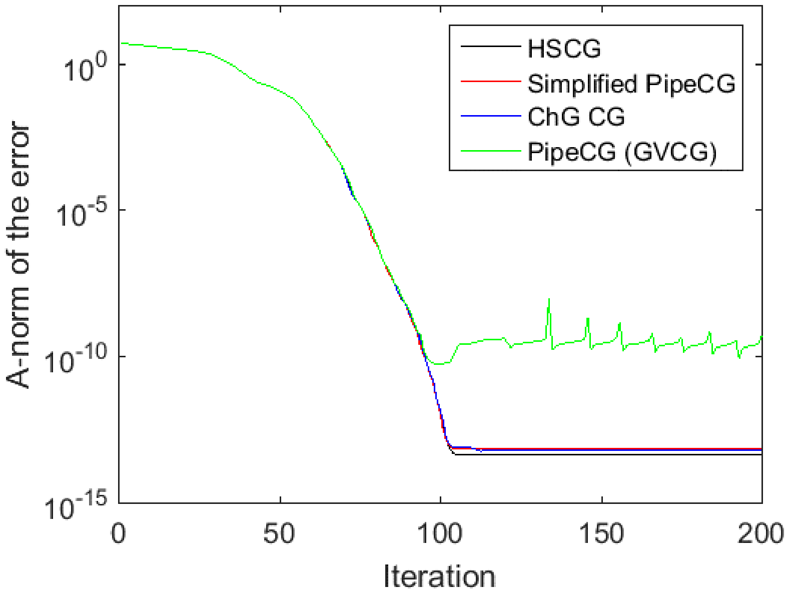
A: **nos4** from UFSMC,
b: equal components in the eigenbasis of A and $||b|| = 1$
$N = 100, \kappa(A) \approx 2e3$



A different problem...

A: **nos4** from UFSMC,
b: equal components in the eigenbasis
of A and $\|b\| = 1$
$N = 100, \kappa(A) \approx 2e3$



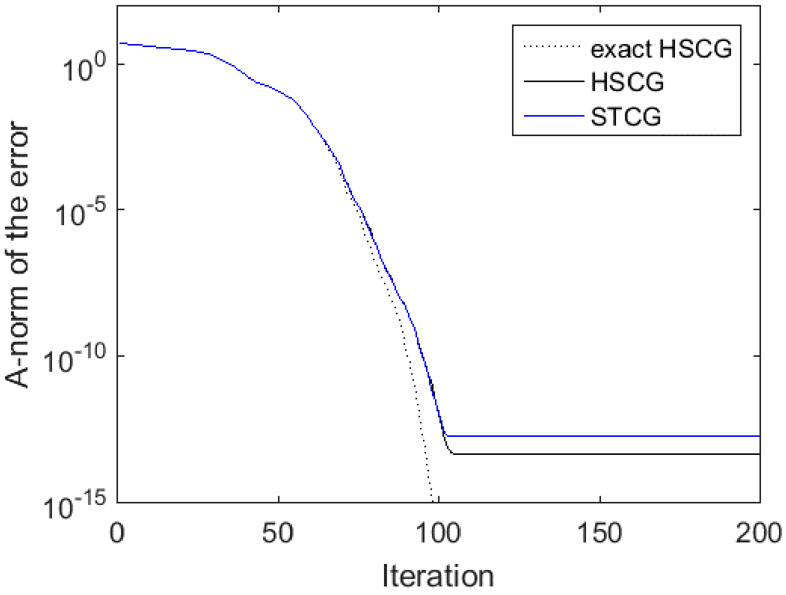


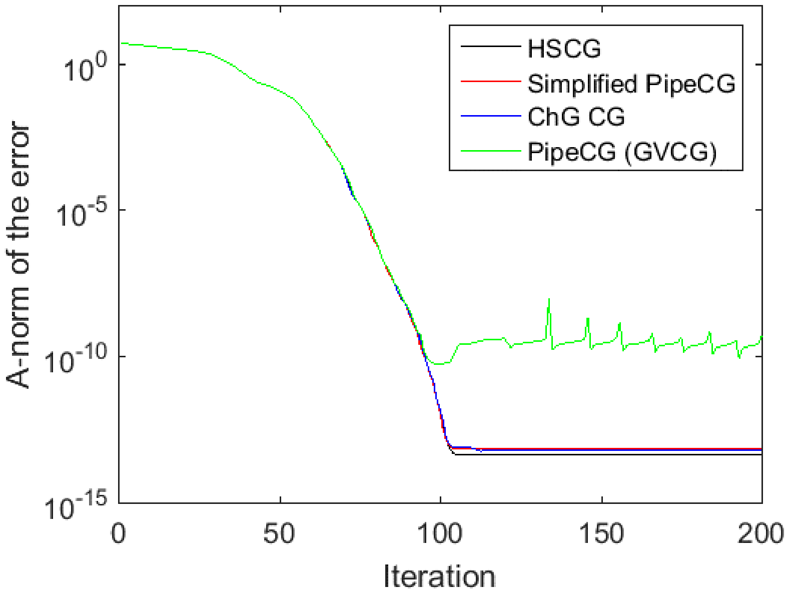
A different problem...

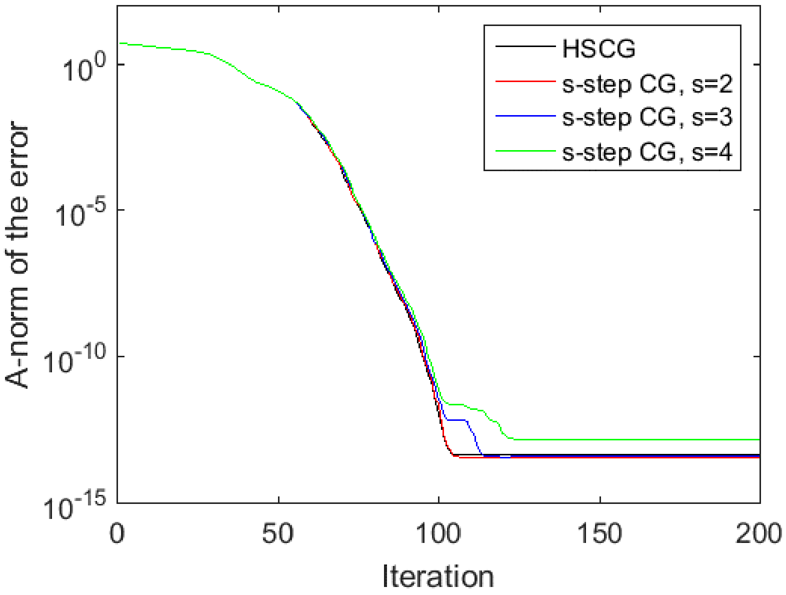
A: **nos4** from UFSMC,

b: equal components in the eigenbasis of A and $\|b\| = 1$

$N = 100, \kappa(A) \approx 2e3$

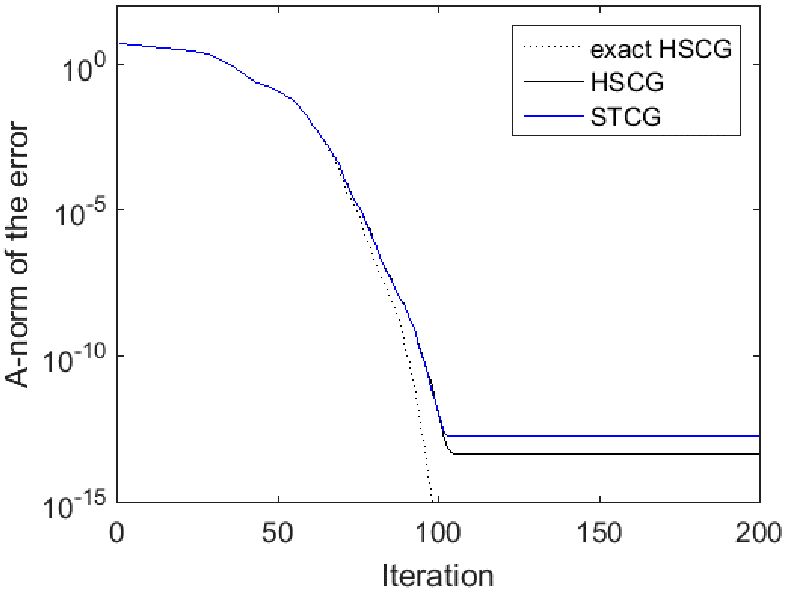




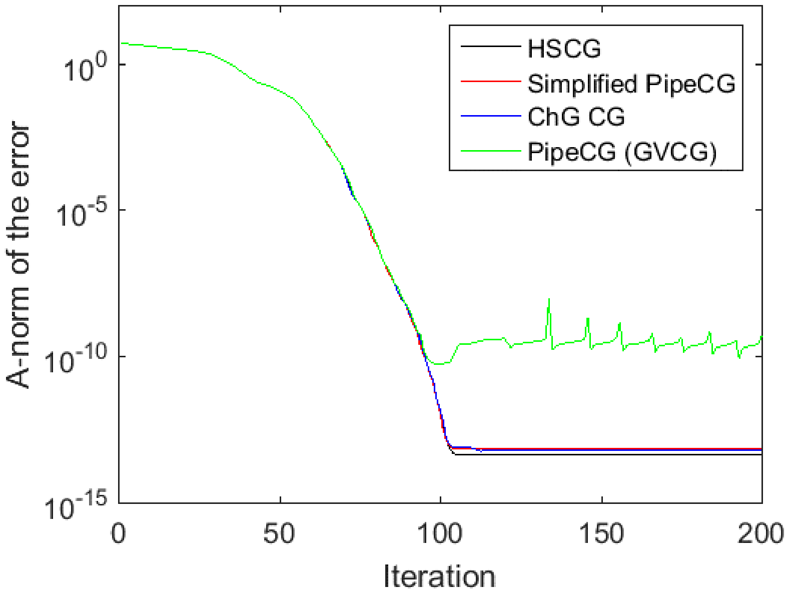


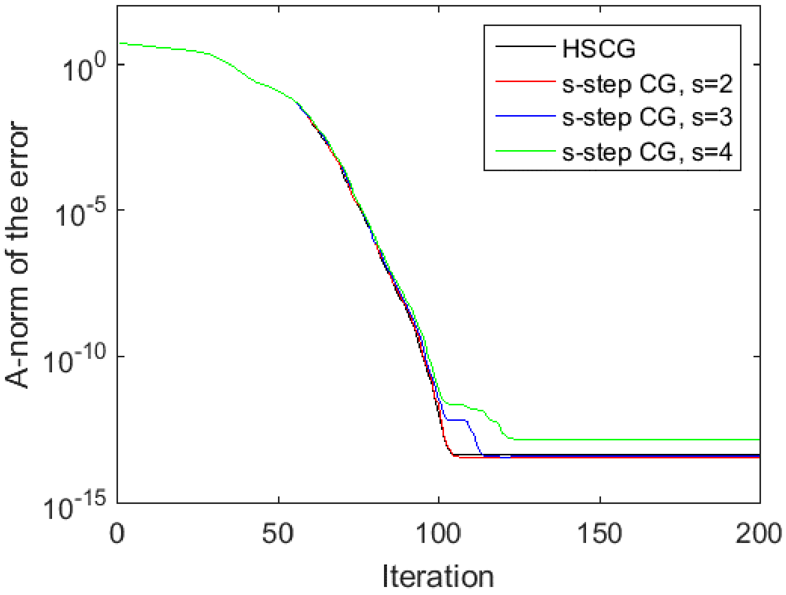
A different problem...

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
$\|x - x_i\|_A \leq 10^{-10}$,
any of these methods will work!

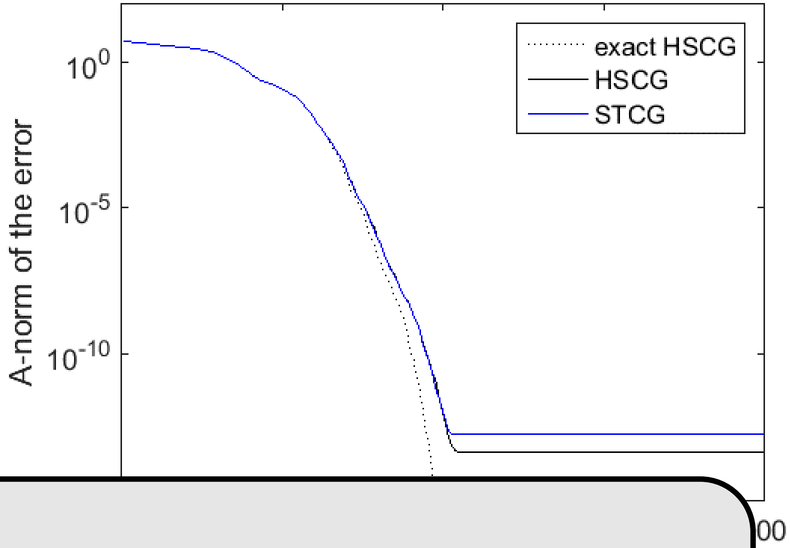




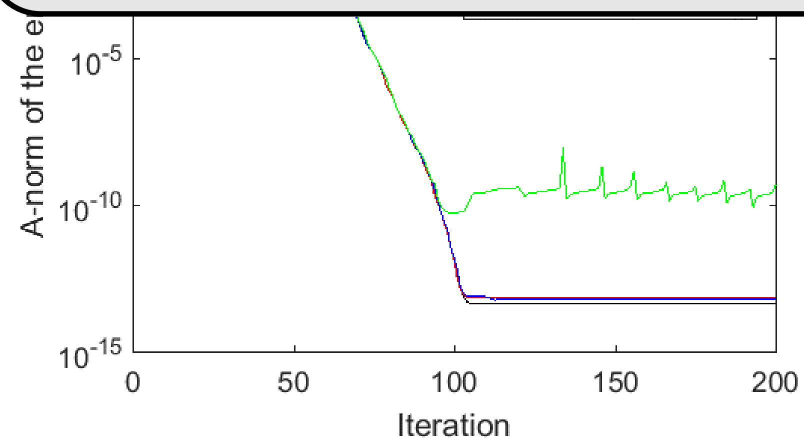
A different problem...

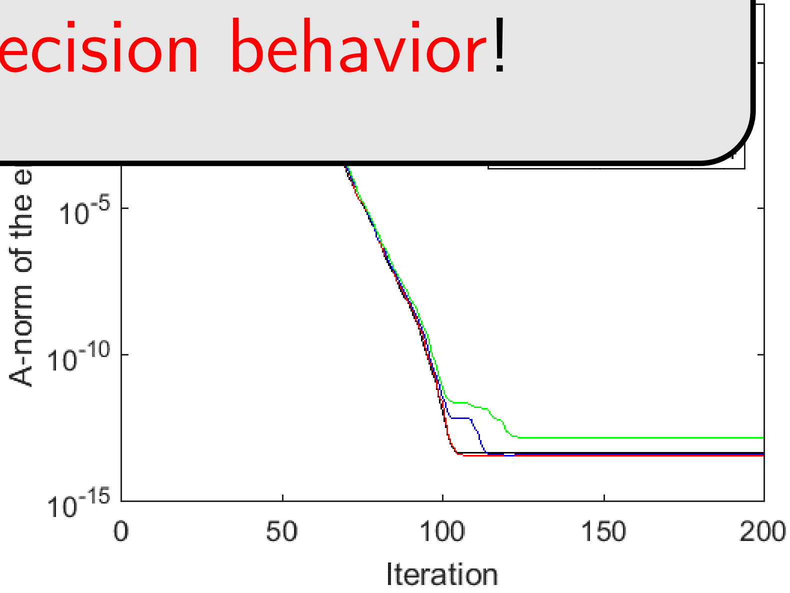
A: **nos4** from UFSMC,
b: equal components in the eigenbasis of A and $\|b\| = 1$
$N = 100, \kappa(A) \approx 2\text{e}3$

If application only requires
$\|x - x_i\|_A \leq 10^{-10}$,
any of these methods will work!



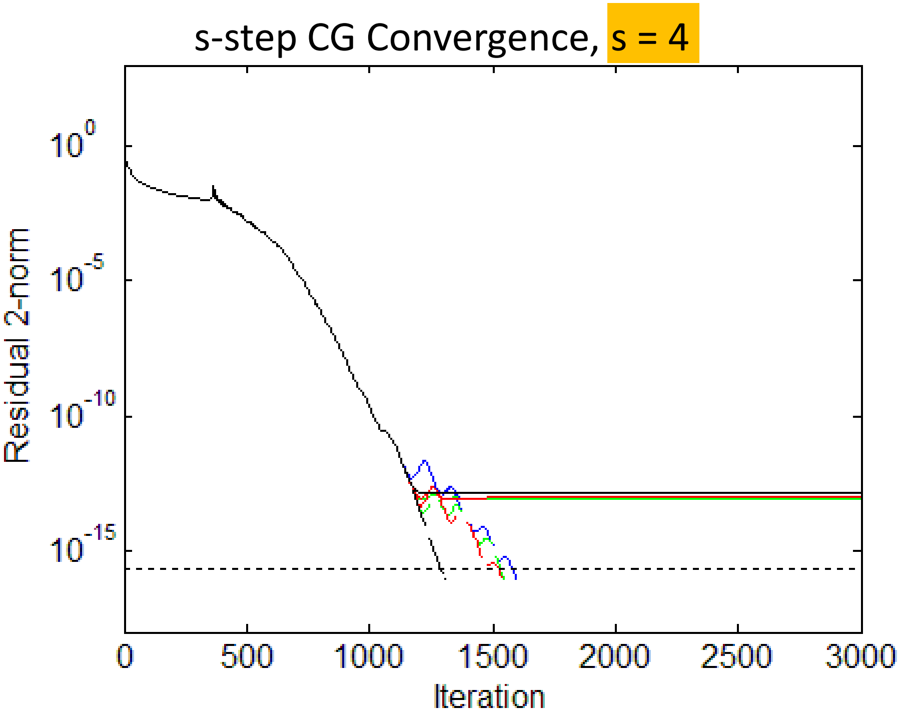
Need adaptive, problem-dependent approach based on understanding of finite precision behavior!



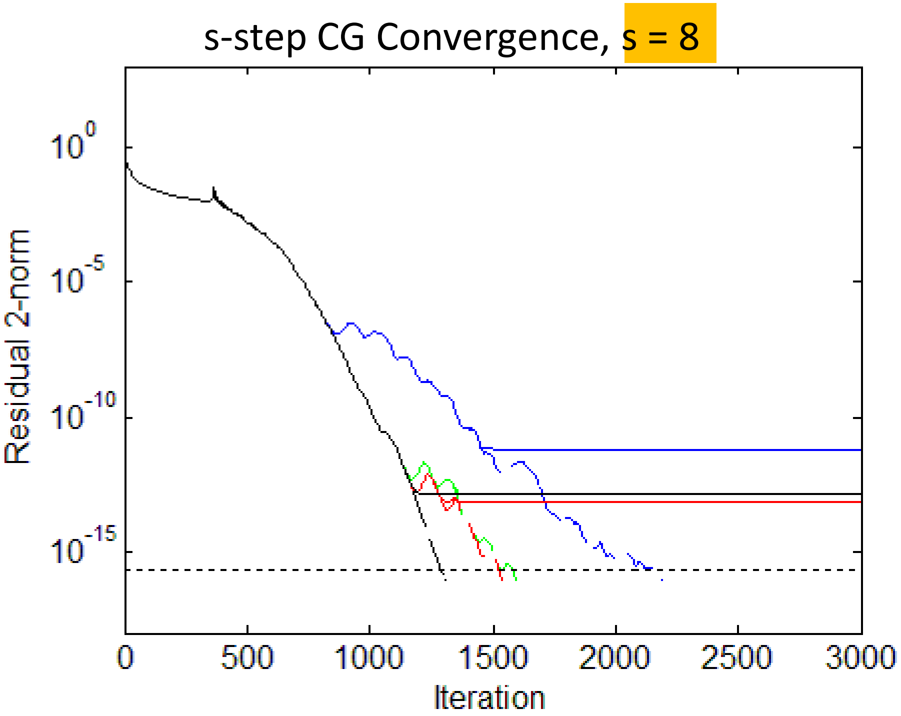


Summary

- Finite precision errors cause loss of attainable accuracy and convergence delay
- In classical CG, attainable accuracy limited only by sum of local rounding errors
- In pipelined CG, sum of many different local rounding errors can be (globally!) amplified
    - Amplification depends on CG recurrence coefficients α and β
        - Not much to do except try to decrease local errors (e.g., by stabilizing shifts)
- In s-step CG, local rounding errors in each outer loop are amplified by a factor related to the condition number of the generated s-step basis matrix
    - Amplification effects are still "local" within an outer loop (block of s iterations)
    - Suggests that basis condition number plays a huge role
- More difficult to precisely characterize convergence delay; further work needed

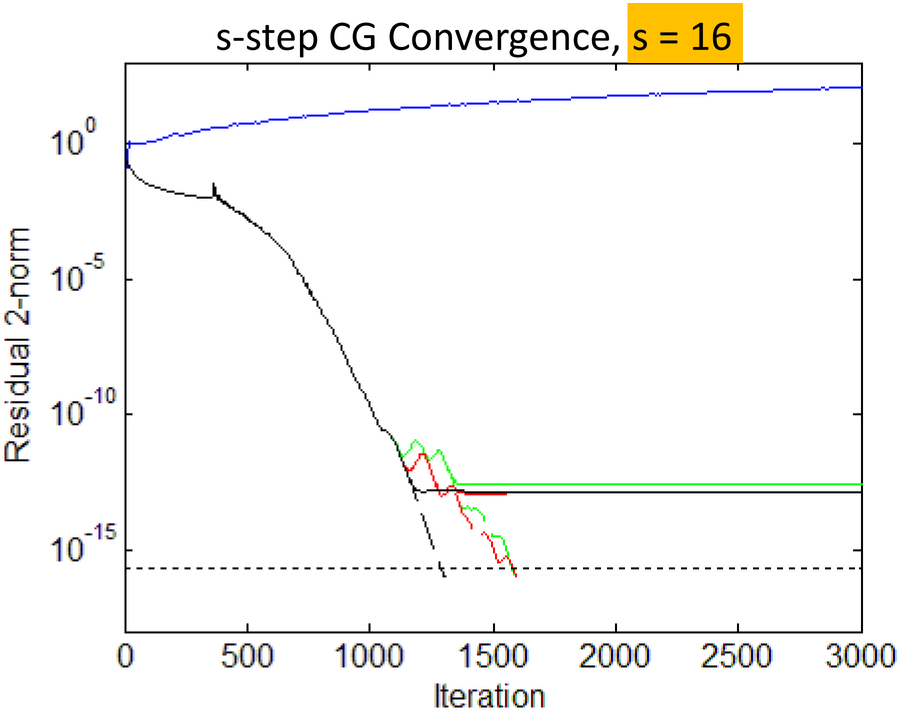


s-step CG Convergence, s = 4



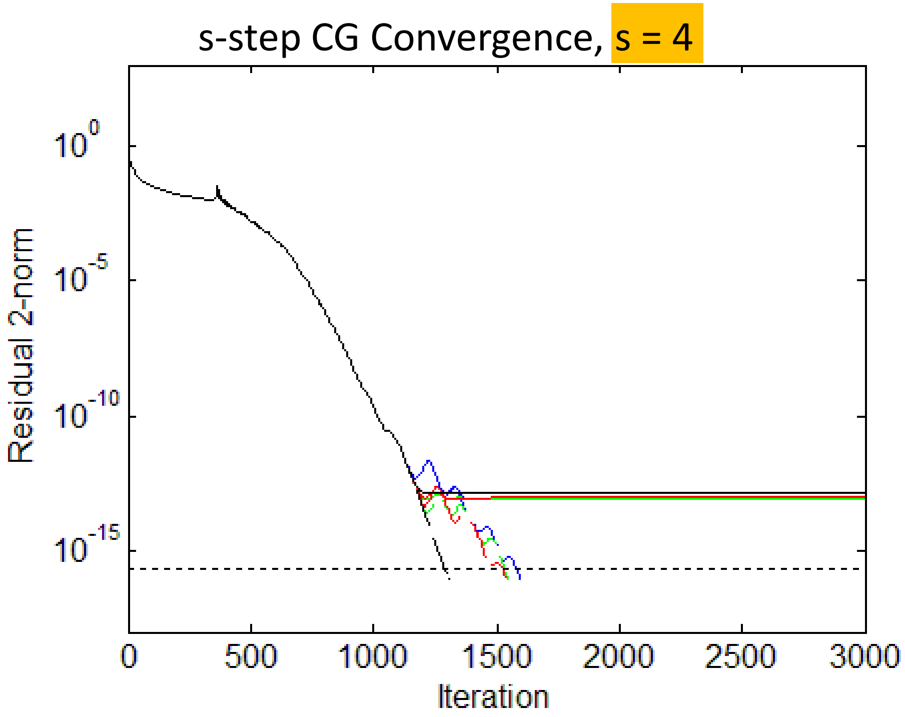
s-step CG Convergence, s = 8

— CG true
- - CG updated
— s-step CG (monomial) true
- - s-step CG (monomial) updated
— s-step CG (Newton) true
- - s-step CG (Newton) updated
— s-step CG (Chebyshev) true
- - s-step CG (Chebyshev) updated

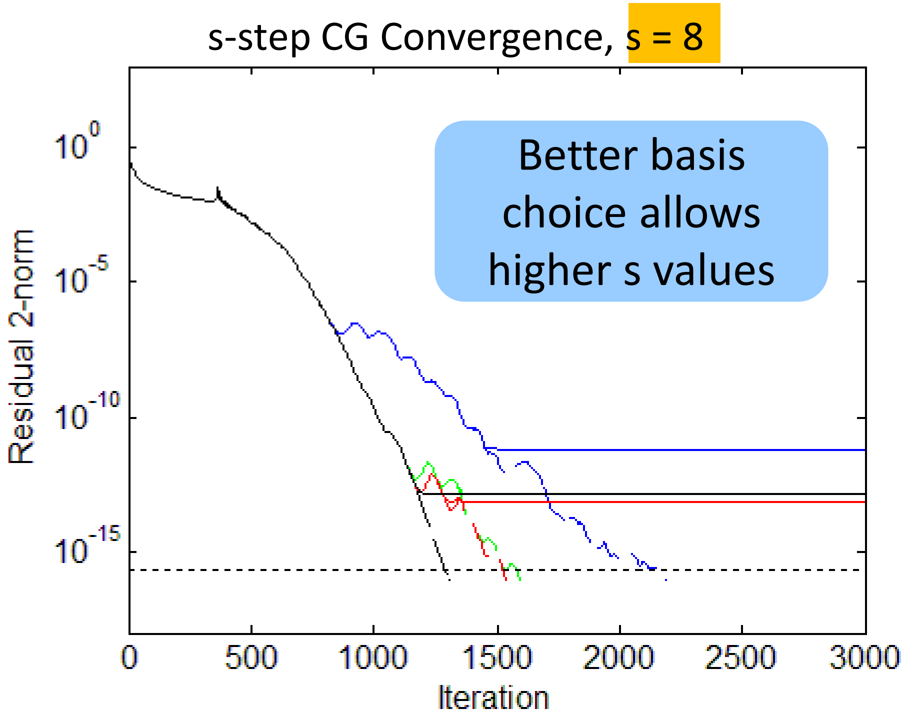


s-step CG Convergence, s = 16

Model Problem: 2D Poisson (5-pt stencil),
$n = 512^2,\ N \approx 10^6, \kappa(A) \approx 10^4$
$b = A(1\sqrt{n} \cdot \mathrm{ones}(n, 1))$

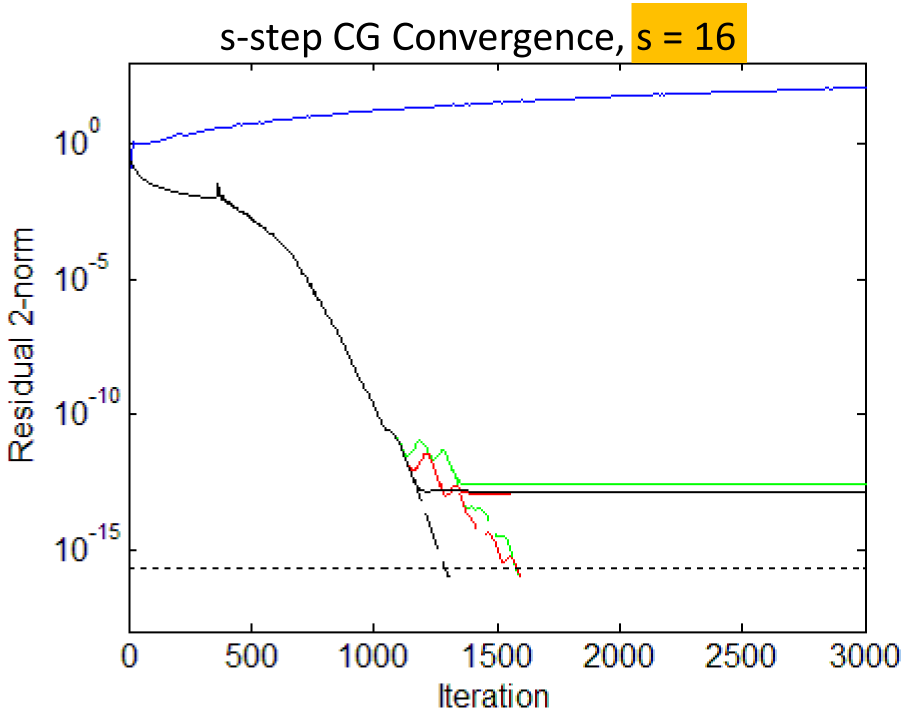


s-step CG Convergence, s = 4



s-step CG Convergence, s = 8

Better basis choice allows higher s values



s-step CG Convergence, s = 16

━ CG true

╌ CG updated

━ s-step CG (monomial) true

╌ s-step CG (monomial) updated

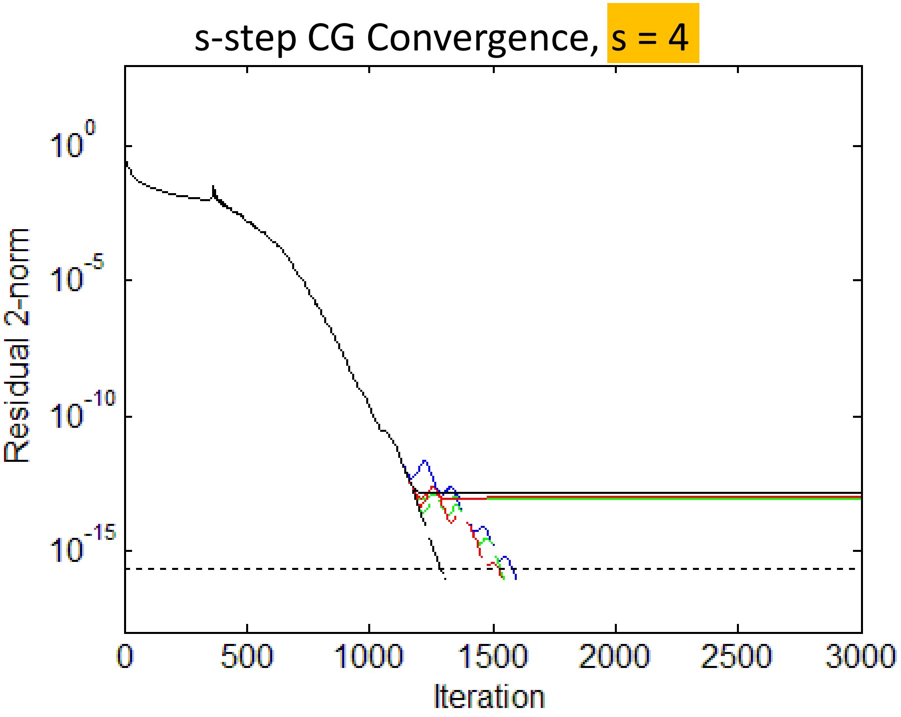
━ s-step CG (Newton) true

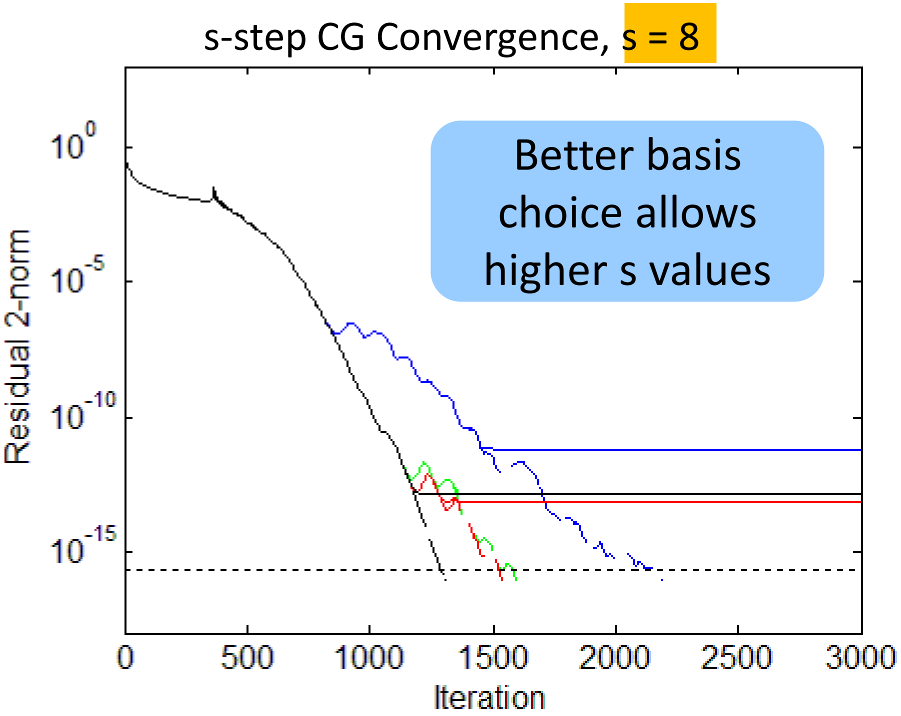
╌ s-step CG (Newton) updated

━ s-step CG (Chebyshev) true

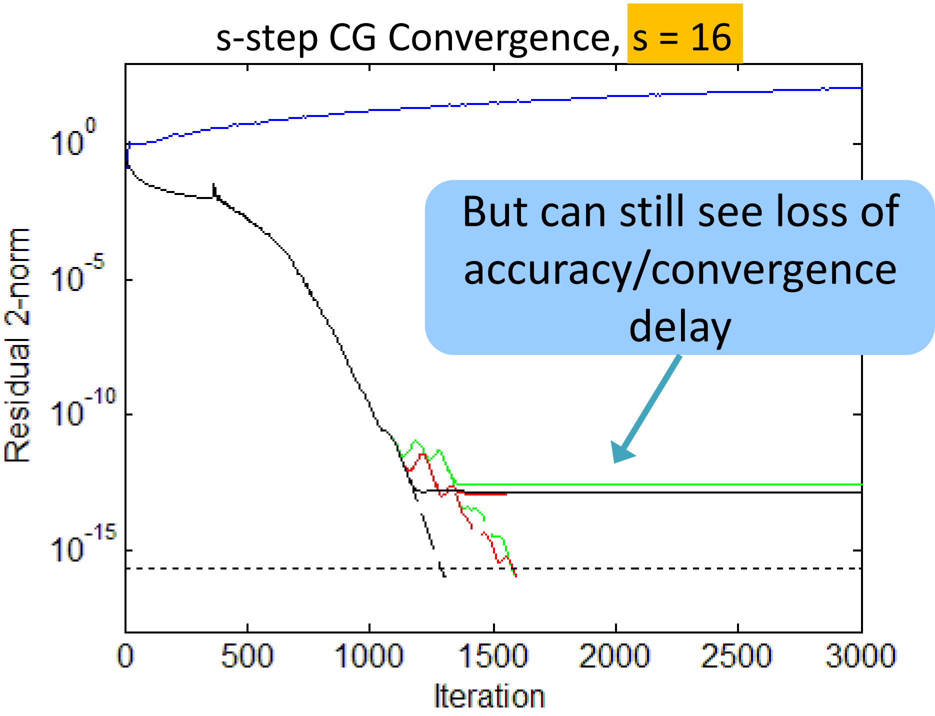
╌ s-step CG (Chebyshev) updated

Model Problem: 2D Poisson (5-pt stencil),
$n = 512^2$, $N \approx 10^6$, $\kappa(A) \approx 10^4$
$b = A(1\sqrt{n} \cdot \text{ones}(n, 1))$





Better basis choice allows higher s values



But can still see loss of accuracy/convergence delay

▬▬	CG true
- - -	CG updated
▬▬	s-step CG (monomial) true
- - -	s-step CG (monomial) updated
▬▬	s-step CG (Newton) true
- - -	s-step CG (Newton) updated
▬▬	s-step CG (Chebyshev) true
- - -	s-step CG (Chebyshev) updated

Model Problem: 2D Poisson (5-pt stencil),
$n = 512^2,\ N \approx 10^6, \kappa(A) \approx 10^4$
$b = A(1\sqrt{n} \cdot \text{ones}(n,1))$

Residual replacement strategy

- Improve accuracy by replacing **computed residual** $\hat{r}_i$ by the **true residual** $\boldsymbol{b} - \boldsymbol{A}\hat{x}_{\boldsymbol{i}}$ in certain iterations
  - Related work for classical CG: van der Vorst and Ye (1999)

Residual replacement strategy

- Improve accuracy by replacing **computed residual** $\hat{r}_i$ by the **true residual** $\boldsymbol{b - A\hat{x}_i}$ in certain iterations
  - Related work for classical CG: van der Vorst and Ye (1999)

- Choose when to replace $\hat{r}_i$ with $b - A\hat{x}_i$ to meet two constraints:
  1. $\|f_i\| = \|b - A\hat{x}_i - \hat{r}_i\|$ is small (relative to $\varepsilon N \|A\| \|\hat{x}_{m+1}\|$)
  2. Convergence rate is maintained (avoid large perturbations to finite precision CG recurrence)

Residual replacement strategy

- Improve accuracy by replacing **computed residual** $\hat{r}_i$ by the **true residual** $\boldsymbol{b - A\hat{x}_i}$ in certain iterations
  - Related work for classical CG: van der Vorst and Ye (1999)

- Choose when to replace $\hat{r}_i$ with $b - A\hat{x}_i$ to meet two constraints:
  1. $\|f_i\| = \|b - A\hat{x}_i - \hat{r}_i\|$ is small (relative to $\varepsilon N \|A\| \|\hat{x}_{m+1}\|$)
  2. Convergence rate is maintained (avoid large perturbations to finite precision CG recurrence)

- Based on derived bound on deviation of residuals, can devise a residual replacement strategy for s-step CG

- Implementation has **negligible cost**

Residual replacement for s-step CG

- Use computable bound for$\|b - A\hat{x}_i - \hat{r}_i\|$ to update d_i, an estimate of error in computing r_i, in each iteration
- Set threshold $\hat{\varepsilon} \approx \sqrt{\varepsilon}$, replace whenever $d_i/\|r_i\|$ reaches threshold

Residual replacement for s-step CG

- Use computable bound for $\|b - A\hat{x}_i - \hat{r}_i\|$ to update d_i, an estimate of error in computing r_i, in each iteration
- Set threshold $\hat{\varepsilon} \approx \sqrt{\varepsilon}$, replace whenever $d_i/\|r_i\|$ reaches threshold

Pseudo-code for residual replacement with group update for s-step CG:

```
if  $d_{i-1} \le \hat{\varepsilon}\|r_{i-1}\|$  and  $d_i > \hat{\varepsilon}\|r_i\|$  and  $d_i > 1.1 d_{init}$
        $z = z + \mathcal{Y}_k\, x'_{k,j} + x_{sk}$
        $x_i = 0$
        $r_i = b - Az$
        $d_{init} = d_i = \varepsilon\big((1 + 2N')\|A\|\|z\| + \|r_i\|\big)$
        $p_i = \mathcal{Y}_k p'_{k,j}$
        break from inner loop and begin new outer loop
end
```

Residual replacement for s-step CG

- Use computable bound for $\|b - A\hat{x}_i - \hat{r}_i\|$ to update d_i, an estimate of error in computing r_i, in each iteration

- Set threshold $\hat{\varepsilon} \approx \sqrt{\varepsilon}$, replace whenever $d_i/\|r_i\|$ reaches threshold

Pseudo-code for residual replacement with group update for s-step CG:

```
if  $d_{i-1} \leq \hat{\varepsilon}\|r_{i-1}\|$  and  $d_i > \hat{\varepsilon}\|r_i\|$  and  $d_i > 1.1 d_{init}$
        $z = z + \mathcal{Y}_k\, x'_{k,j} + x_{sk}$        group update of approximate solution
        $x_i = 0$
        $r_i = b - Az$
        $d_{init} = d_i = \varepsilon\big((1 + 2N')\|A\|\|z\| + \|r_i\|\big)$
        $p_i = \mathcal{Y}_k p'_{k,j}$
        break from inner loop and begin new outer loop
end
```

Residual replacement for s-step CG

- Use computable bound for $\|b - A\hat{x}_i - \hat{r}_i\|$ to update d_i, an estimate of error in computing r_i, in each iteration

- Set threshold $\hat{\varepsilon} \approx \sqrt{\varepsilon}$, replace whenever $d_i / \|r_i\|$ reaches threshold

Pseudo-code for residual replacement with group update for s-step CG:

```
if  $d_{i-1} \le \hat{\varepsilon}\|r_{i-1}\|$  and  $d_i > \hat{\varepsilon}\|r_i\|$  and  $d_i > 1.1 d_{init}$
        $z = z + \mathcal{Y}_k\, x'_{k,j} + x_{sk}$
        $x_i = 0$
        $r_i = b - Az$
        $d_{init} = d_i = \varepsilon\big((1 + 2N')\|A\|\|z\| + \|r_i\|\big)$
        $p_i = \mathcal{Y}_k p'_{k,j}$
        break from inner loop and begin new outer loop
end
```

group update of approximate solution

set residual to true residual

Residual replacement for s-step CG

- Use computable bound for$\|b - A\hat{x}_i - \hat{r}_i\|$ to update d_i, an estimate of error in computing r_i, in each iteration

- Set threshold $\hat{\varepsilon} \approx \sqrt{\varepsilon}$, replace whenever $d_i / \|r_i\|$ reaches threshold

Pseudo-code for residual replacement with group update for s-step CG:

```
if  $d_{i-1} \le \hat{\varepsilon}\|r_{i-1}\|$  and  $d_i > \hat{\varepsilon}\|r_i\|$  and  $d_i > 1.1 d_{init}$
        $z = z + \mathcal{Y}_k\, x'_{k,j} + x_{sk}$
        $x_i = 0$
        $r_i = b - Az$
        $d_{init} = d_i = \varepsilon\big((1 + 2N')\|A\|\|z\| + \|r_i\|\big)$
        $p_i = \mathcal{Y}_k p'_{k,j}$
        break from inner loop and begin new outer loop
end
```

group update of approximate solution

set residual to true residual

A computable bound

- In each iteration, update error estimate d_i $(i \equiv sk + j)$ by:

$$
\begin{aligned}
d_i &\equiv d_{i-1} \\
&+\varepsilon\Big[(4+N')\big(\|A\|\,\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{x}'_{k,j}|\big\| + \big\||\hat{\mathcal{Y}}_k|\cdot|\mathcal{B}_k|\cdot|\hat{x}'_{k,j}|\big\|\big) + \big\||\hat{\mathcal{Y}}_k|\cdot|\hat{r}'_{k,j}|\big\|\Big] \\
&+\varepsilon\begin{cases} \|A\|\|\hat{x}_{sk+s}\|+(2+2N')\|A\|\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{x}'_{k,s}|\big\|+N'\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{r}'_{k,s}|\big\|, & j = s \\ 0, & \text{o.w.} \end{cases}
\end{aligned}
$$

where $N' = \max(N, 2s + 1)$.

A computable bound

- In each iteration, update error estimate d_i $(i \equiv sk + j)$ by:

Estimated only once

$$
\begin{aligned}
d_i &\equiv d_{i-1} \\
&+\varepsilon\Big[(4+N')\Big(\|A\|\ \big\||\hat{\mathcal{Y}}_k|\cdot|\hat{x}'_{k,j}|\big\| + \big\||\hat{\mathcal{Y}}_k|\cdot|\mathcal{B}_k|\cdot|\hat{x}'_{k,j}|\big\|\Big) + \big\||\hat{\mathcal{Y}}_k|\cdot|\hat{r}'_{k,j}|\big\|\Big] \\
&+\varepsilon\begin{cases} \|A\|\|\hat{x}_{sk+s}\|+(2+2N')\|A\|\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{x}'_{k,s}|\big\|+N'\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{r}'_{k,s}|\big\|, & j = s \\ 0, & \text{o.w.} \end{cases}
\end{aligned}
$$

where $N' = \max(N, 2s + 1)$.

A computable bound

- In each iteration, update error estimate d_i $(i \equiv sk + j)$ by:

$O(s^3)$ flops per s iterations; ≤1 reduction per s iterations to compute $\left(|\hat{\mathcal{Y}}_k|^T|\hat{\mathcal{Y}}_k|\right)$

$$d_i \equiv d_{i-1}$$

$$+\varepsilon\left[(4+N')\left(\|A\| \left\| |\hat{\mathcal{Y}}_k| \cdot |\hat{x}'_{k,j}| \right\| + \left\| |\hat{\mathcal{Y}}_k| \cdot |\mathcal{B}_k| \cdot |\hat{x}'_{k,j}| \right\|\right) + \left\| |\hat{\mathcal{Y}}_k| \cdot |\hat{r}'_{k,j}| \right\|\right]$$

$$+\varepsilon\begin{cases} \|A\|\|\hat{x}_{sk+s}\| + (2+2N')\|A\|\left\| |\hat{\mathcal{Y}}_k| \cdot |\hat{x}'_{k,s}| \right\| + N'\left\| |\hat{\mathcal{Y}}_k| \cdot |\hat{r}'_{k,s}| \right\|, & j = s \\ 0, & \text{o.w.} \end{cases}$$

where $N' = \max(N, 2s + 1)$.

A computable bound

- In each iteration, update error estimate d_i $(i \equiv sk + j)$ by:

$O(s^2)$ flops per s iterations; no communication

$$d_i \equiv d_{i-1}$$

$$+\varepsilon\Big[(4+N')\Big(\|A\| \, \big\||\hat{\mathcal{Y}}_k|\cdot|\hat{x}'_{k,j}|\big\| + \big\||\hat{\mathcal{Y}}_k|\cdot|\mathcal{B}_k|\cdot|\hat{x}'_{k,j}|\big\|\Big) + \big\||\hat{\mathcal{Y}}_k|\cdot|\hat{r}'_{k,j}|\big\|\Big]$$

$$+\varepsilon\begin{cases} \|A\|\|\hat{x}_{sk+s}\|+(2+2N')\|A\|\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{x}'_{k,s}|\big\|+N'\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{r}'_{k,s}|\big\|, & j = s \\ 0, & \text{o.w.} \end{cases}$$

where $N' = \max(N, 2s + 1)$.

A computable bound

- In each iteration, update error estimate d_i $(i \equiv sk + j)$ by:

Extra computation all lower order terms, communication only increased by *at most* factor of 2

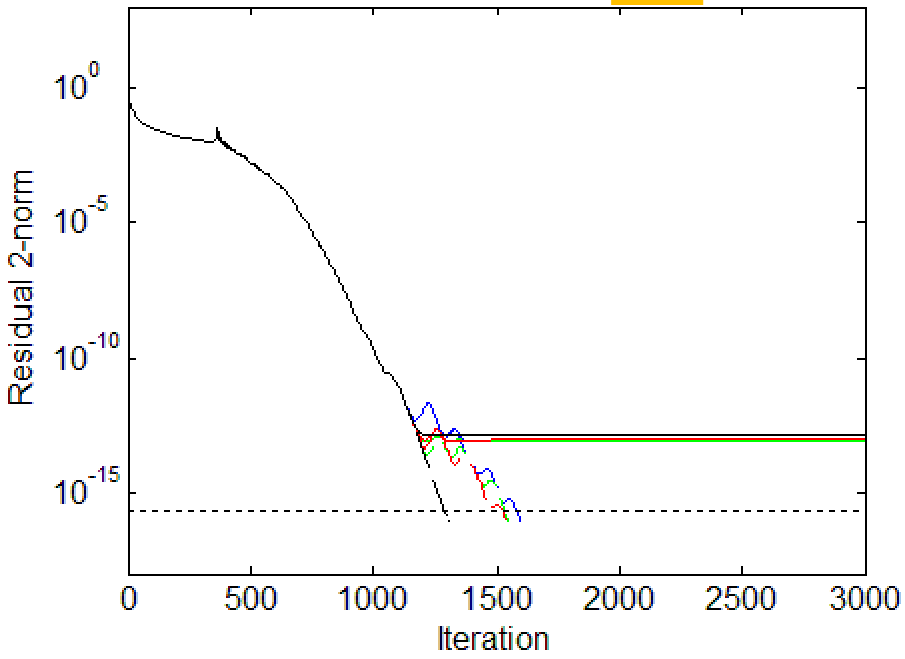
$$d_i \equiv d_{i-1}$$

$$+\varepsilon\Big[(4+N')\Big(\|A\|\,\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{x}'_{k,j}|\big\| + \big\||\hat{\mathcal{Y}}_k|\cdot|\mathcal{B}_k|\cdot|\hat{x}'_{k,j}|\big\|\Big) + \big\||\hat{\mathcal{Y}}_k|\cdot|\hat{r}'_{k,j}|\big\|\Big]$$

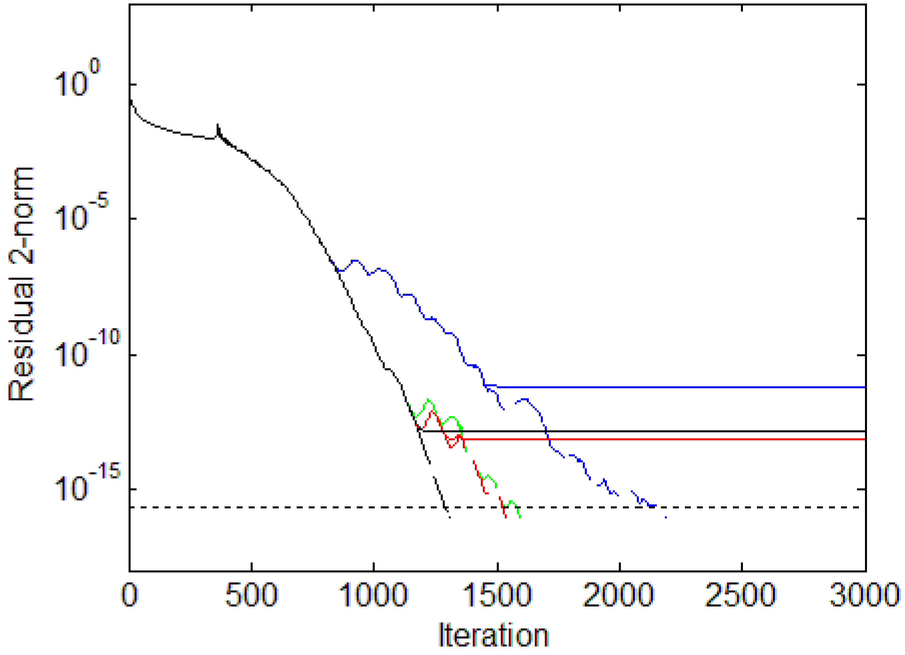
$$+\varepsilon\begin{cases} \|A\|\|\hat{x}_{sk+s}\|+(2+2N')\|A\|\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{x}'_{k,s}|\big\|+N'\big\||\hat{\mathcal{Y}}_k|\cdot|\hat{r}'_{k,s}|\big\|, & j = s \\ 0, & \text{o.w.} \end{cases}$$

where $N' = \max(N, 2s + 1)$.

s-step CG Convergence, s = 4



s-step CG Convergence, s = 8



— CG true

- - CG updated

— s-step CG (monomial) true

- - s-step CG (monomial) updated

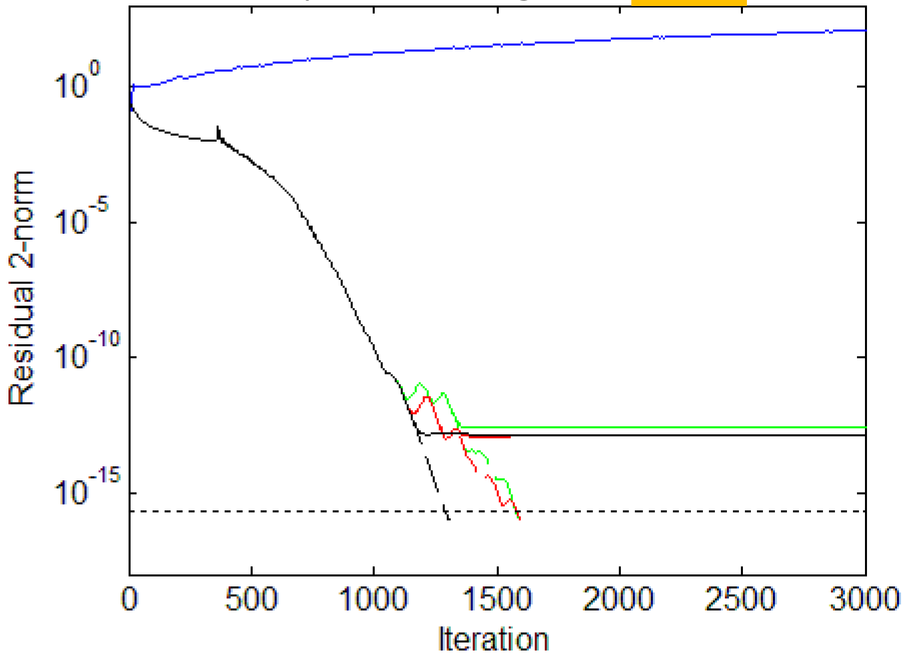
— s-step CG (Newton) true

- - s-step CG (Newton) updated

— s-step CG (Chebyshev) true

- - s-step CG (Chebyshev) updated

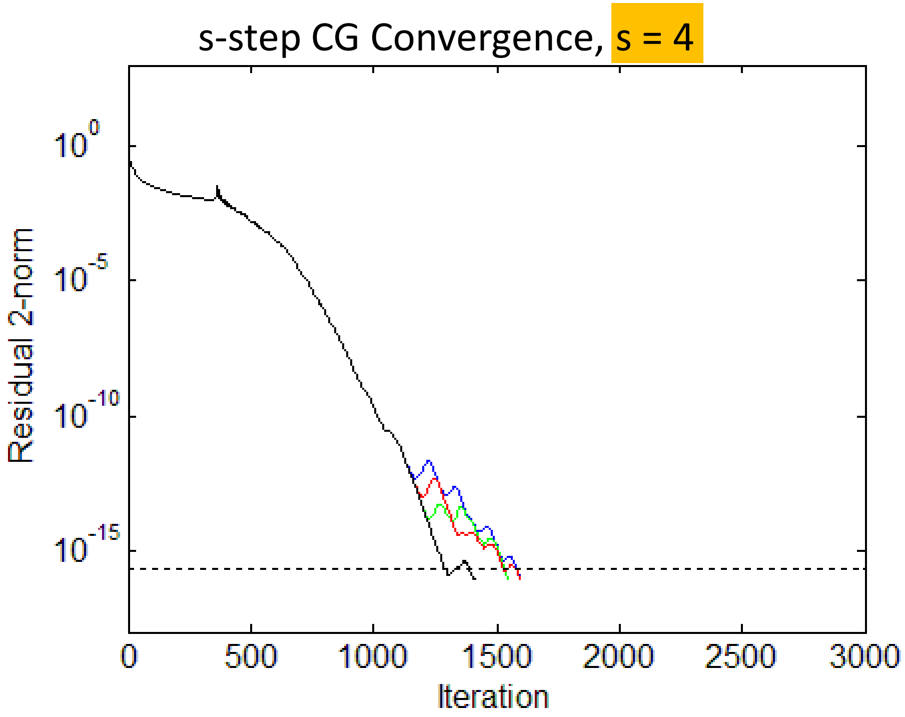
s-step CG Convergence, s = 16

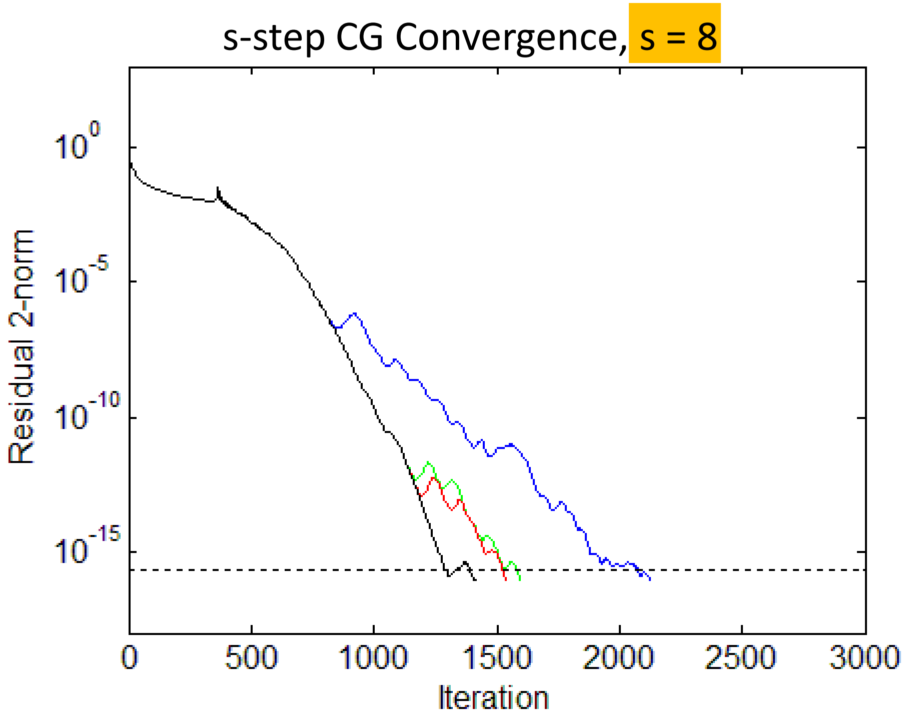


Model Problem: 2D Poisson (5-pt stencil),

$n\ = 512^2,\ N \approx 10^6, \kappa(A) \approx 10^4$

$b = A(1\sqrt{n} \cdot \mathrm{ones}(n,1))$





— CG+**RR** true

- - CG+**RR** updated

— s-step CG+**RR** (monomial) true

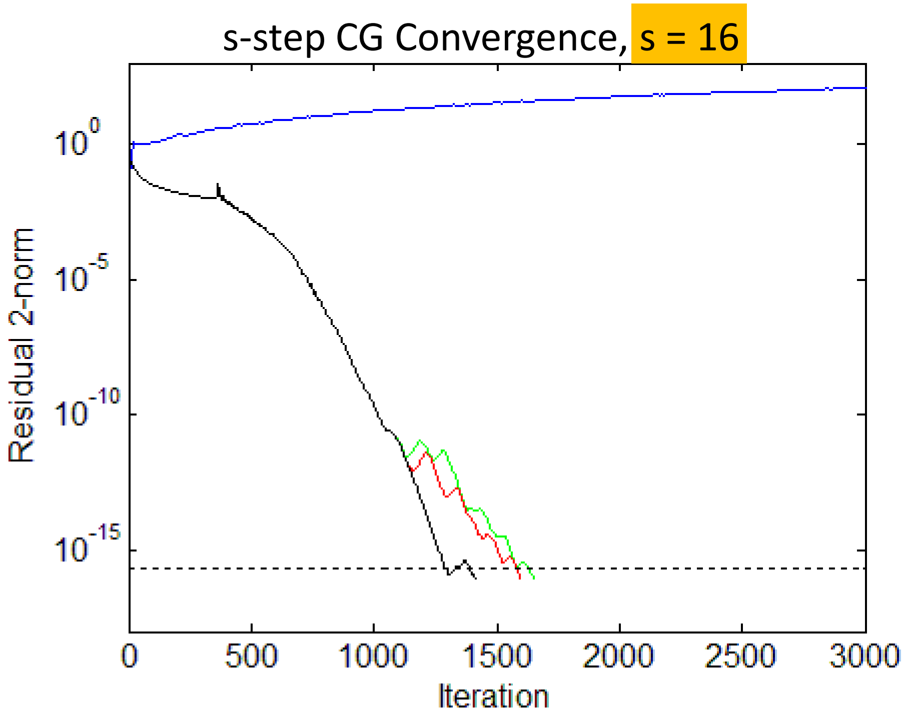
- - s-step CG+**RR** (monomial) updated

— s-step CG+**RR** (Newton) true

- - s-step CG+**RR** (Newton) updated

— s-step CG+**RR**(Chebyshev) true

- - s-step CG+**RR**(Chebyshev) updated

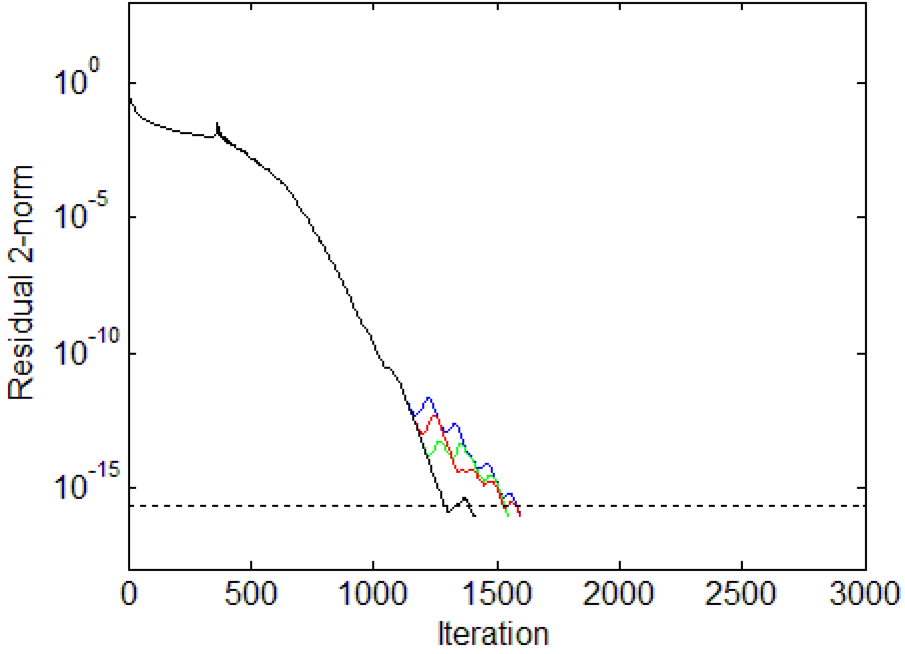


Model Problem: 2D Poisson (5-pt stencil),

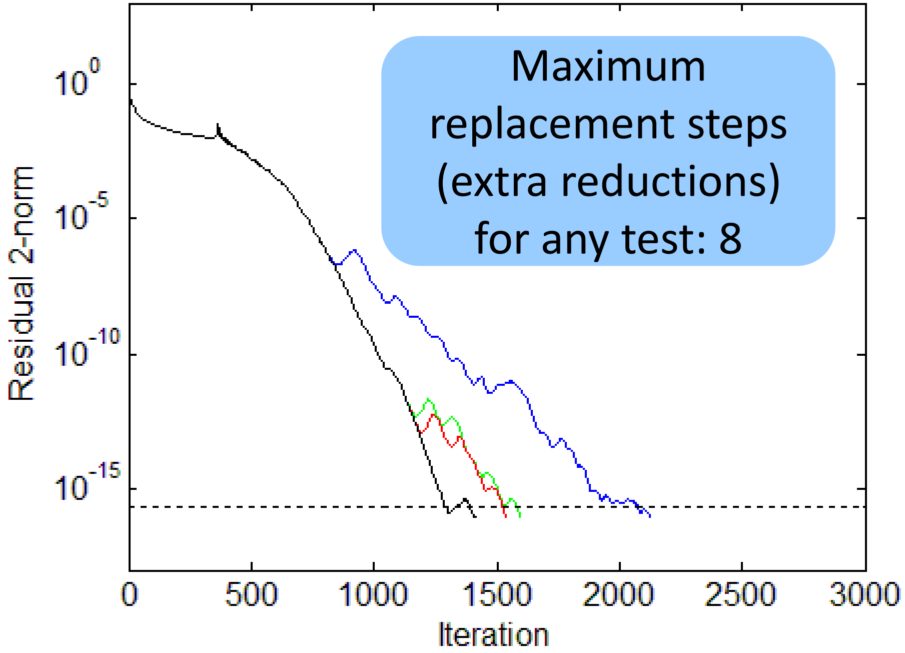
$n = 512^2,\ N \approx 10^6, \kappa(A) \approx 10^4$

$$b = A(1\sqrt{n} \cdot \mathrm{ones}(n,1))$$

s-step CG Convergence, s = 4



s-step CG Convergence, s = 8



Maximum replacement steps (extra reductions) for any test: 8

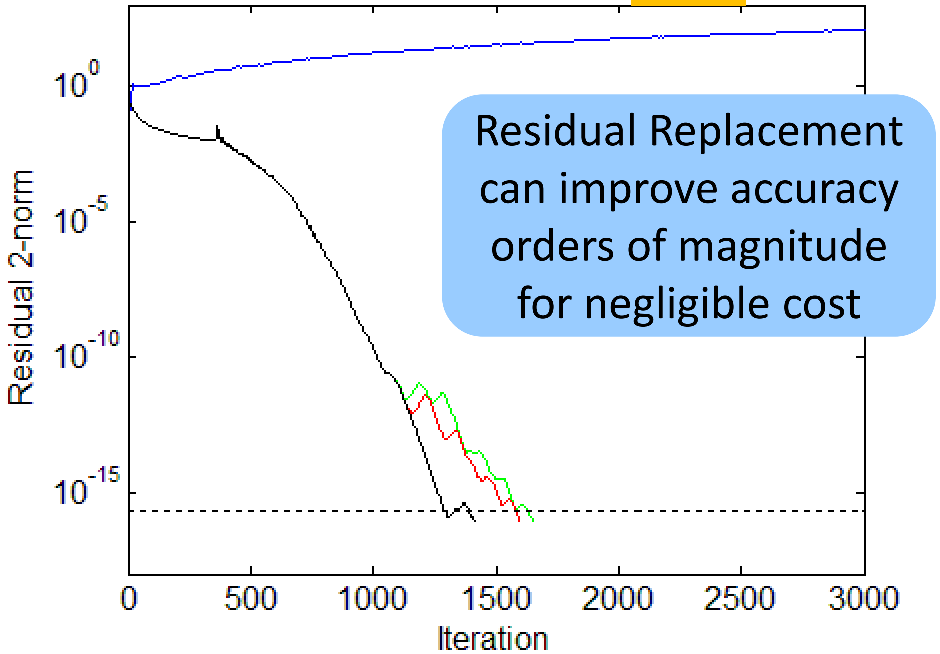
- —— CG+**RR** true
- - - - CG+**RR** updated
- —— s-step CG+**RR** (monomial) true
- - - - s-step CG+**RR** (monomial) updated
- —— s-step CG+**RR** (Newton) true
- - - - s-step CG+**RR** (Newton) updated
- —— s-step CG+**RR**(Chebyshev) true
- - - - s-step CG+**RR**(Chebyshev) updated

Model Problem: 2D Poisson (5-pt stencil),

$n = 512^2,\ N \approx 10^6, \kappa(A) \approx 10^4$

$b = A(1\sqrt{n} \cdot \text{ones}(n,1))$

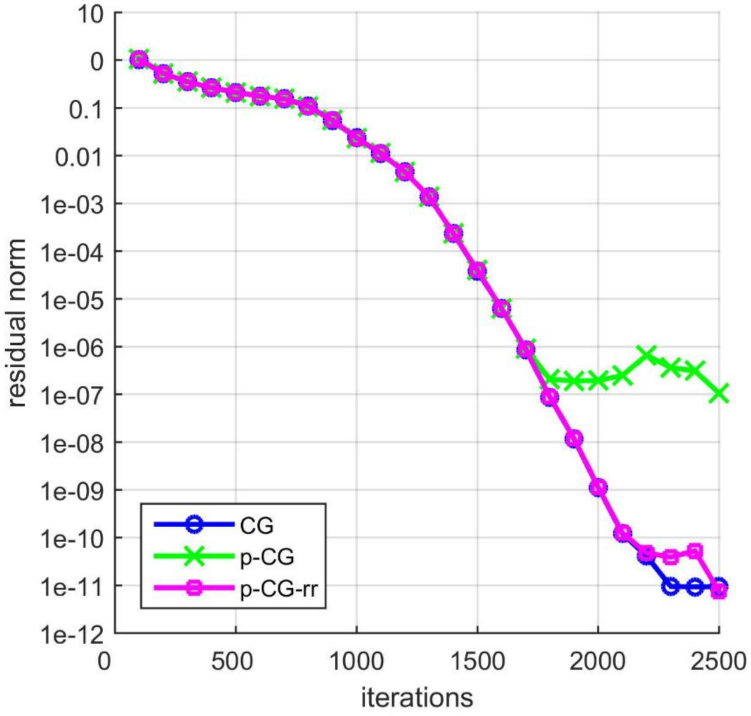
s-step CG Convergence, s = 16



Residual Replacement can improve accuracy orders of magnitude for negligible cost

Pipelined CG with residual replacement

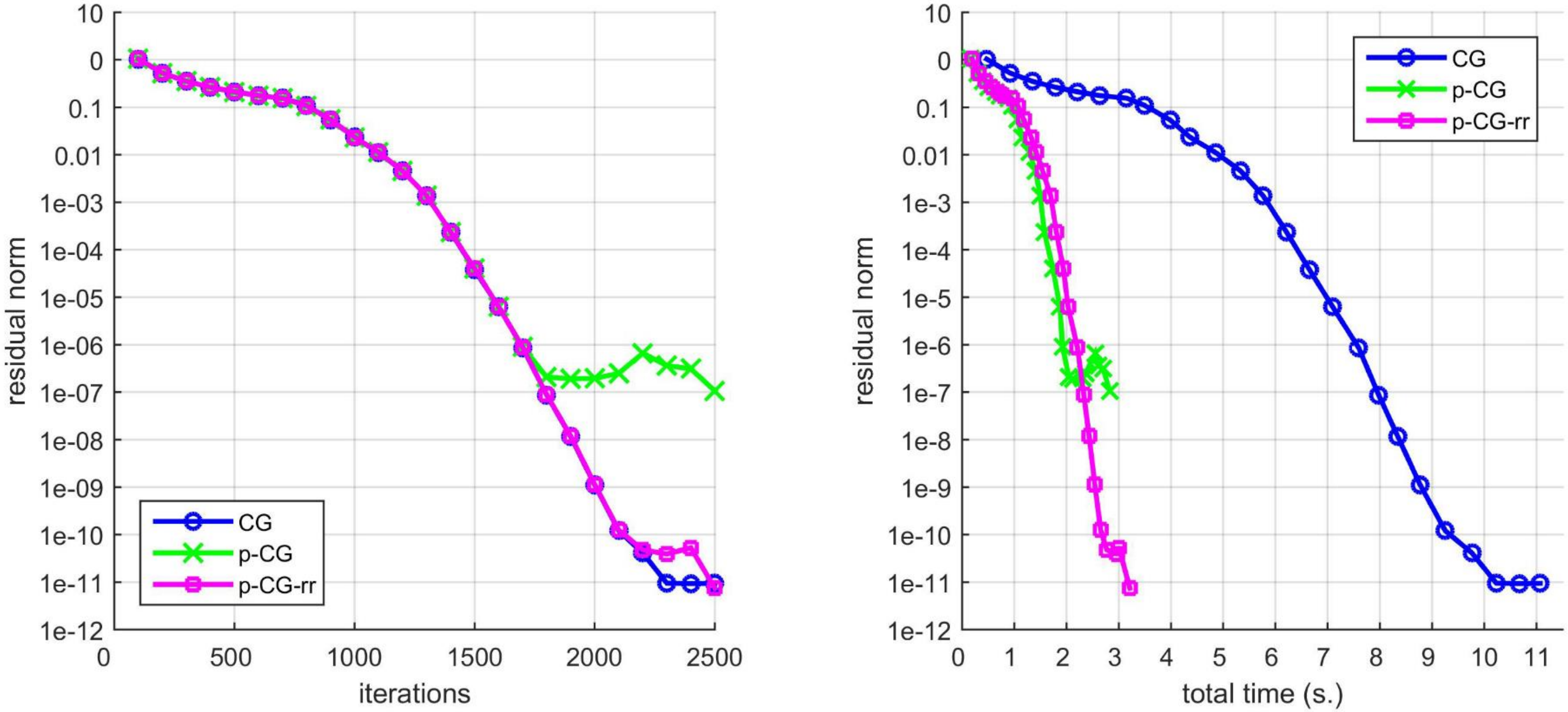
Similar approach possible for pipelined CG; see (Cools et al., 2018)



20 nodes (two 6-core Intel Xeon X5660 Nehalem 2:80-GHz processors per node),
2D Poisson problem with 1e6 unknowns;
in pipelined CG with residual replacement, 39 replacements were performed.

Pipelined CG with residual replacement

Similar approach possible for pipelined CG; see (Cools et al., 2018)



20 nodes (two 6-core Intel Xeon X5660 Nehalem 2:80-GHz processors per node),
2D Poisson problem with 1e6 unknowns;
in pipelined CG with residual replacement, 39 replacements were performed.

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

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

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

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

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

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

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

$$\Gamma_k \equiv c \cdot \|\hat{\mathcal{Y}}_k^+\| \, \||\hat{\mathcal{Y}}_k|\| \lesssim \frac{\varepsilon^*}{\varepsilon \max\limits_{j\in\{0,\dots,s\}} \|\hat{r}_{m+j}\|}$$

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

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

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

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

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

- $\|\hat{r}_i\|$ 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

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

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

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

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

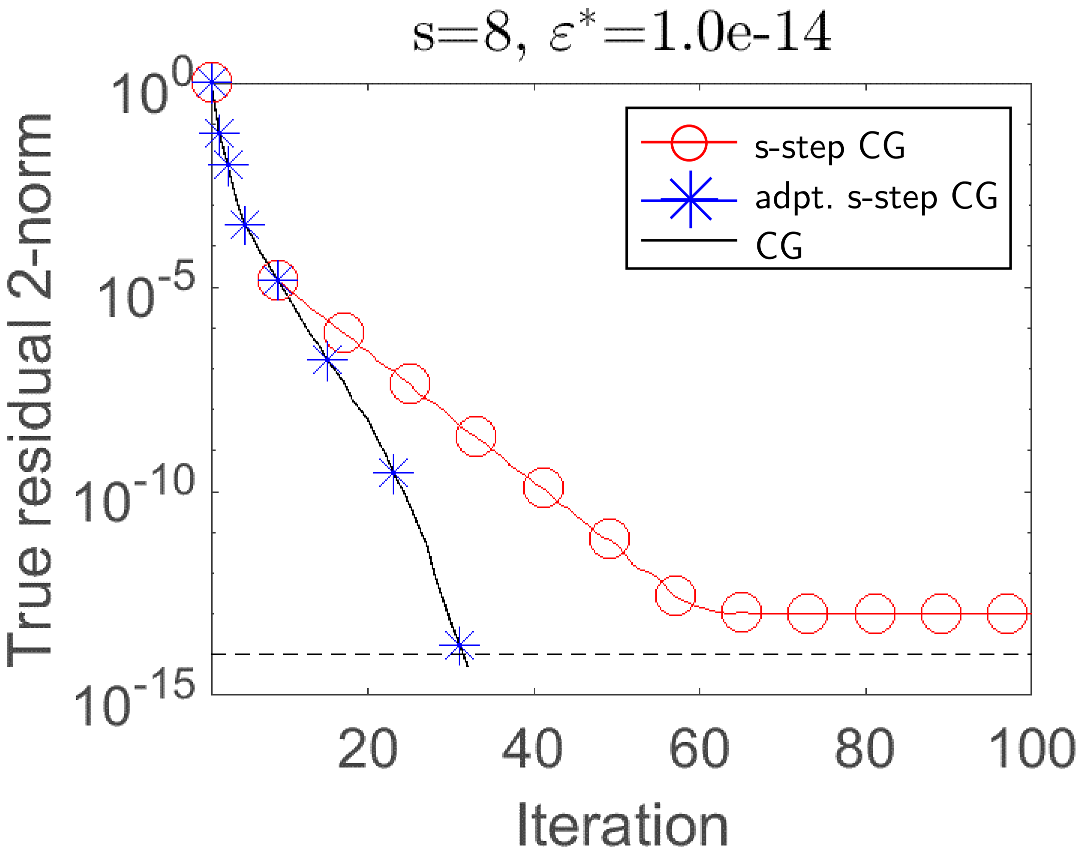
- $\|\hat{r}_i\|$ large → Γ_k must be small; $\|\hat{r}_i\|$ small → Γ_k can grow

$\Rightarrow$ adaptive s-step approach [C., 2018]

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

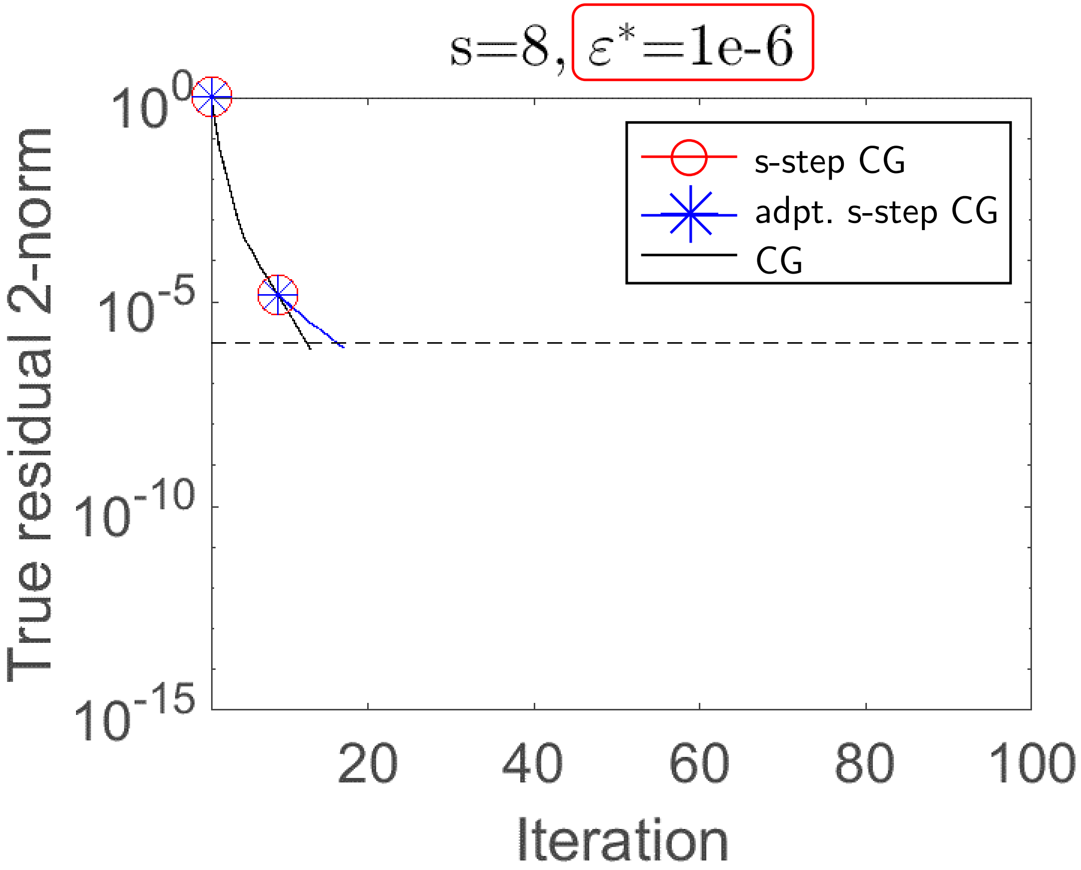
Adaptive s-step CG

mesh3e1 (SuiteSparse)
$n = 289$
$\kappa(A) \approx 10$
$b_i = 1/\sqrt{N}$



Adaptive s-step CG

mesh3e1 (SuiteSparse)
$n = 289$
$\kappa(A) \approx 10$
$b_i = 1/\sqrt{N}$



Extensions to 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

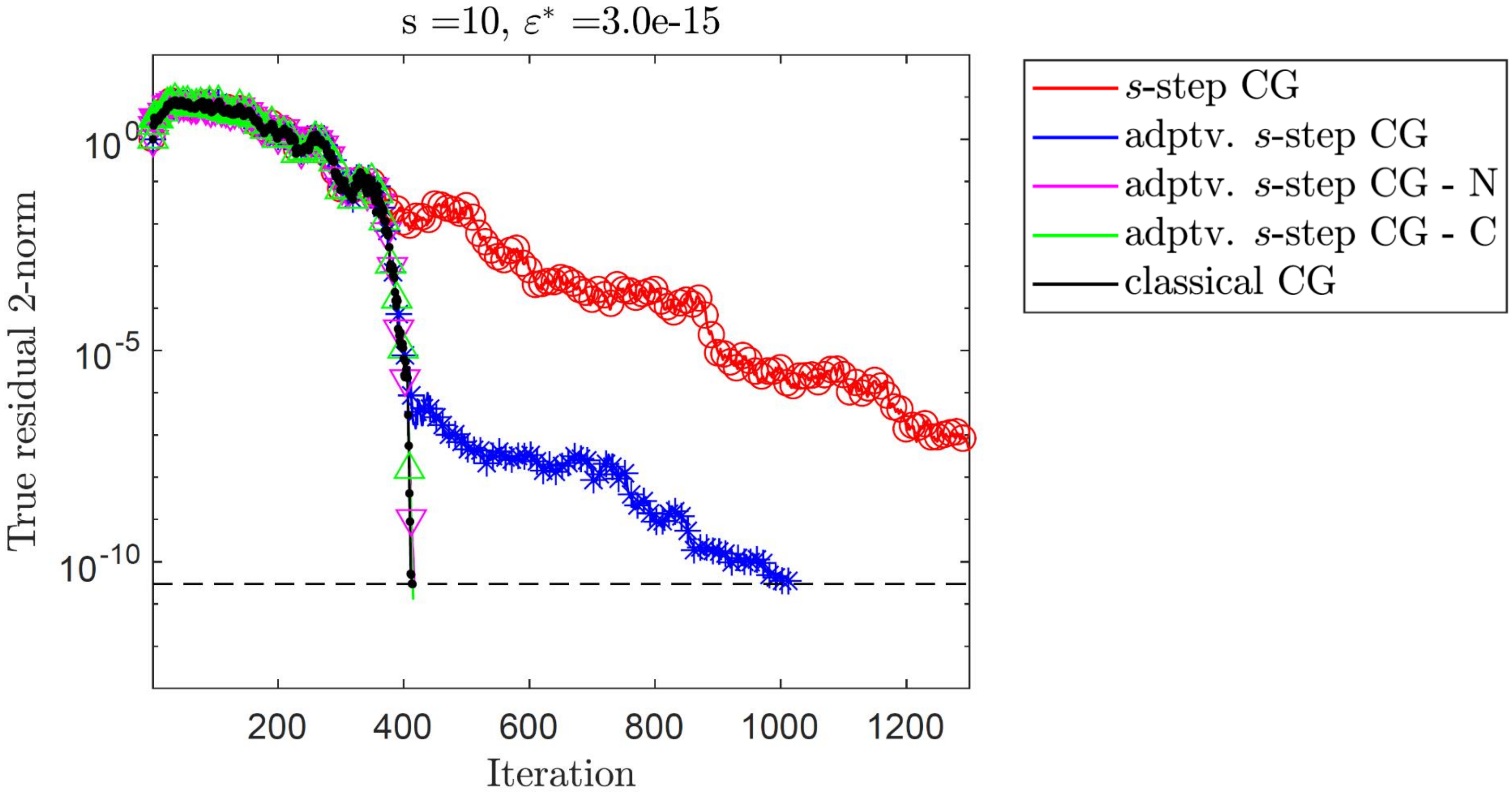
Extensions to adaptive s-step CG

- 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

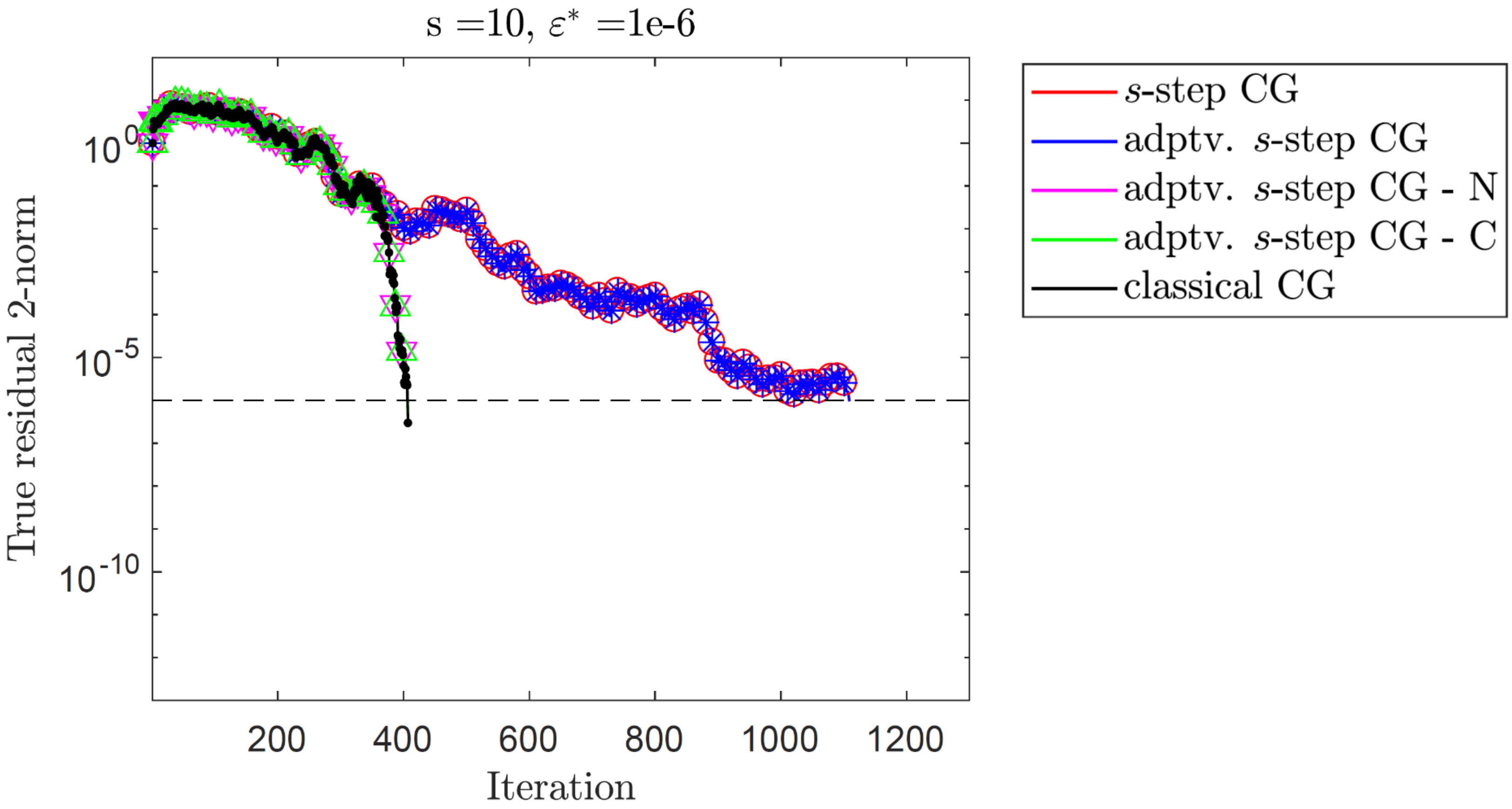
$b_i = 1/\sqrt{N}$



Number of global synchronizations

Fixed s-step	Old adaptive s-step	Improved adaptive s-step w/Newton	Improved adaptive s-step w/Chebyshev	classical CG
-	132	**59**	**53**	414

$A =$ 494bus from SuiteSparse
$b_i = 1/\sqrt{N}$



Number of global synchronizations

Fixed s-step	Old adaptive s-step	Improved adaptive s-step w/Newton	Improved adaptive s-step w/Chebyshev	classical CG
111	111	**43**	**43**	407

When to use an HPC variant

- Solve constitutes a bottleneck within the application (Amdahl's law)

When to use an HPC variant

- Solve constitutes a bottleneck within the application (Amdahl's law)
- Krylov solve is communication-bound (particularly latency bound due to global synchronization)

When to use an HPC variant

- Solve constitutes a bottleneck within the application (Amdahl's law)
- Krylov solve is communication-bound (particularly latency bound due to global synchronization)
- Extremal eigenvalues are known or easy to estimate

When to use an HPC variant

- Solve constitutes a bottleneck within the application (Amdahl's law)
- Krylov solve is communication-bound (particularly latency bound due to global synchronization)
- Extremal eigenvalues are known or easy to estimate
- Accuracy much less than machine epsilon required by the application

When to use an HPC variant

- Solve constitutes a bottleneck within the application (Amdahl's law)
- Krylov solve is communication-bound (particularly latency bound due to global synchronization)
- Extremal eigenvalues are known or easy to estimate
- Accuracy much less than machine epsilon required by the application
- s-step methods
    - The matrix is well-partitioned into domains with low surface-to-volume ratio
    - Simple preconditioning is sufficient/the preconditioner is amenable to communication avoidance
    - The same coefficient matrix (or at least a coefficient matrix with the same nonzero structure) will be reused over multiple solves
    - improvement even for small numbers of nodes (reduces both intra- and inter-processor communication)

When to use an HPC variant

- Solve constitutes a bottleneck within the application (Amdahl's law)
- Krylov solve is communication-bound (particularly latency bound due to global synchronization)
- Extremal eigenvalues are known or easy to estimate
- Accuracy much less than machine epsilon required by the application
- s-step methods
  - The matrix is well-partitioned into domains with low surface-to-volume ratio
  - Simple preconditioning is sufficient/the preconditioner is amenable to communication avoidance
  - The same coefficient matrix (or at least a coefficient matrix with the same nonzero structure) will be reused over multiple solves
  - improvement even for small numbers of nodes (reduces both intra- and inter-processor communication)
- (deep) pipelined methods
  - cost of applying preconditioner + SpMV is less than or the same as a global synchronization
  - improvement only for large numbers of nodes

Looking Forward

- Better understanding of finite precision behavior

- Improved usability
  - More adaptivity, autotuning; less left to the user

- Hybrid methods?
  - stationary iterative method + Krylov subspace method

- Fault tolerance?
  - MTTF=0 on an exascale machine
  - A problem to be handled at the algorithm level, or...?

- Making use of specialized hardware
  - accelerators, GPUs, etc.
  - multiple precisions?
  - new performance model, new programming model, bigger tuning space

Thank you!

carson@karlin.mff.cuni.cz

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