Analyzing and Improving Attainable Accuracy for the Communication Hiding Pipelined Conjugate Gradient Method

SIAM Conference on Computational Science and Engineering (CSE’19)
Spokane Convention Center, WA, US, February 25 - March 1, 2019

University of Antwerp* [Belgium], LBNL‡ [USA], INRIA Bordeaux† [France]

S. Cools*, J. Cornelis*, W. Vanroose*, P. Ghysels‡, E. F. Yetkin†, E. Agullo†, L. Giraud†

*E-mail: siegfried.cools@uantwerp.be

Universiteit Antwerpen
Motivation

Communication is hard for humans

Introducing some buzzwords you will frequently hear during this talk ...

“synchronization bottleneck”

“communication latency” / “error propagation”
Motivation

Communication is hard for computers

Data movement (communication) is much more time consuming than flops (computations), so reducing time spent communicating data is essential for HPC

⇒ Communication avoiding / Communication hiding
High communication cost has motivated several approaches to reducing the global synchronization bottleneck in Krylov subspace methods:

Avoiding communication:  **s-step Krylov subspace methods** *
[A. Chronopoulous, E. de Sturler, J. Demmel, M. Hoemmen, E. Carson, L. Grigori, J. Erhel, ...]
- Compute iterations in blocks of $s$, allows use of matrix power kernels
- Reduces number of synchronizations per iteration by a factor of $O(s)$

Hiding communication:  **Pipelined Krylov subspace methods** *
[P. Ghysels, W. Vanroose, S. C., P. Sanan, B. Gropp, I. Yamazaki, P. Luszczek, ...]
- Introduce auxiliary (basis) vectors to decouple SpMV and inner products
- Enables overlapping of communication and computations

* All methods are equivalent to their corresponding Krylov subspace methods in exact arithmetic
Krylov subspace methods
General concepts

Iteratively improve an approximate solution of the linear system $Ax = b$, with

$$x_i \in x_0 + \mathcal{K}_i(A, r_0) = x_0 + \text{span}\{r_0, Ar_0, A^2r_0, \ldots, A^{i-1}r_0\}, \quad r_i = b - Ax_i.$$  

- minimize certain error measure over Krylov subspace $\mathcal{K}_i(A, r_0)$
- Krylov subspace methods:
  - Conjugate Gradients (CG), Lanczos, GMRES, MinRES, BiCG, BiCGStab, CGLS, ...
- Preconditioners:
  - AMG & GMG, Domain Decomposition Methods, FETI, BDDC, Incomplete factorization, Physics based preconditioners, ...
- usually in combination with sparse linear algebra/stencil application
- three algorithmic building blocks:
  i. dot-product
     - $O(N)$ flops
     - global synchronization (MPI_Allreduce)
  ii. SpMV
     - $O(\text{nnz})$ flops
     - neighbor communication only
  iii. axpy
     - $O(N)$ flops
     - no communication
Krylov subspace methods

Classic Conjugate Gradients (CG)

i. dot-products
   - 2 global reductions: latency dominated
   - time scales as $\log_2(\#\text{partitions})$

ii. SpMV
   - computationally expensive
   - good scaling (minor communication)

iii. axpy's
   - vector operations (recurrences)
   - perfect scaling (no communication)

Essentially sequential operations (line-per-line execution)
Krylov subspace methods
Pipelined Conjugate Gradients

i. Communication avoiding:
dot-products grouped in one global reduction phase per iteration

ii. Communication hiding:
overlap global synchronization with SpMV (+ Prec) computation

iii. No free lunch: Additional recurrence relations (axpy's) for the auxiliary vectors $s_i = Ap_i$, $w_i = Ar_i$, $z_i = As_i$

Algorithm Pipelined CG

1: procedure PIPE-CG($A$, $b$, $x_0$)
2: $r_0 := b - Ax_0$; $w_0 := Ar_0$
3: for $i = 0, \ldots$ do
4: $\gamma_i := (r_i, r_i)$
5: $\delta := (w_i, r_i)$
6: $q_i := Aw_i$
7: if $i > 0$ then
8: $\beta_i := \gamma_i / \gamma_{i-1}$; $\alpha_i := (\delta / \gamma_i - \beta_i / \alpha_{i-1})^{-1}$
9: else
10: $\beta_i := 0$; $\alpha := \gamma_i / \delta$
11: end if
12: $z_i := q_i + \beta_i z_{i-1}$
13: $s_i := w_i + \beta_i s_{i-1}$
14: $p_i := r_i + \beta_i p_{i-1}$
15: $x_{i+1} := x_i + \alpha_i p_i$
16: $r_{i+1} := r_i - \alpha_i s_i$
17: $w_{i+1} := w_i - \alpha_i z_i$
18: end for
19: end procedure

Ghysels & Vanroose (2014)
Krylov subspace methods

Deep \( \ell \)-length pipelined CG

Classic KSM:

\[
\begin{align*}
\text{SpMV} & \quad \text{GiRed} & \quad \text{Prec} & \quad \text{GiRed} \\
\text{SpMV} & \quad \text{Prec} & \quad \text{GiRed} & \quad \text{SpMV}
\end{align*}
\]

Pipelined KSM:

\[
\begin{align*}
\text{GiRed} & \quad \text{SpMV} & \quad \text{Prec} & \quad \text{GiRed} \\
\text{SpMV} & \quad \text{Prec} & \quad \text{GiRed} & \quad \text{SpMV} & \quad \text{Prec} & \quad \text{GiRed}
\end{align*}
\]

Deep pipelined KSM:

\[
\begin{align*}
\text{SpMV} & \quad \text{Prec} & \quad \text{GiRed} \\
\text{SpMV} & \quad \text{Prec} & \quad \text{GiRed} & \quad \text{SpMV} & \quad \text{Prec} & \quad \text{GiRed}
\end{align*}
\]

Pipelined “D-Lanczos” \cite{saad2003}:

Consider the Lanczos relation

\[
AV_i = V_{i+1} T_{i+1,i}
\]

with \( A \) symmetric, \( V_{i+1} = [v_0, v_1, \ldots, v_i] \) the Krylov subspace basis and \( T_{i+1,i} \) a symmetric tridiagonal matrix

\[
T_{i+1,i} = \begin{pmatrix}
\gamma_0 & \delta_0 & & \\
\delta_0 & \gamma_1 & & \\
& \delta_0 & \gamma_2 & \\
& & \ddots & \ddots & \ddots \end{pmatrix}
\begin{pmatrix}
\delta_{i-2} & \\
\gamma_{i-1} & \\
\delta_{i-1}
\end{pmatrix}
\]
Krylov subspace methods

Deep $\ell$-length pipelined CG

Classic KSM:

Pipelined KSM:

Deep pipelined KSM:

Pipelined “D-Lanczos” \( \text{Saad (2003)} \)

Introduce the auxiliary Krylov subspace basis \( Z_{i+1} = [z_0, z_1, \ldots, z_i] \) that runs \( l \) SpMVs ahead of the basis \( V_{i-l+1} \) as

\[
  z_i := \begin{cases} 
    v_0 & j = 0, \\
    P_i(A)v_0 & 0 < i \leq l, \\
    P_i(A)v_{i-l} & i > l,
  \end{cases}
\]

with polynomials \( P_i(t) \) of fixed order \( l \)

\[
P_i(t) := \prod_{j=0}^{l-1} (t - \sigma_j),
\]

where \( l \) is the pipeline length. \( \text{Ghysels et al. (2013)} \)
Krylov subspace methods
Deep $\ell$-length pipelined CG

- Applying $P_l(A)$ to $AV_i = V_{i+1}T_{i+1,i}$ yields a Lanczos-type relation

$$AZ_i = Z_{i+1}B_{i+1,i}$$

with $B_{i+1,i}$ shifted tridiagonal matrix.

- Auxiliary basis vectors are computed using a three-term recurrence relation

$$z_{i+1} = \left( Az_i - \gamma_{i-1}z_i - \delta_{i-1}z_{i-1} \right) / \delta_{i-1}$$

- Original basis vectors are computed using a multi-term recurrence relation

$$v_{i-l+1} = \left( z_{i-l+1} - \sum_{j=i-3l+1}^{i-l} g_{j,i-l+1} v_j \right) / g_{i-l+1,i-l+1}.$$
Algorithm 1 \( l \)-length pipelined \( p(l) \)-CG

\( \textbf{Input:} \ A, \ b, \ x_0, \ l, \ m, \ \tau \)

1: \( r_0 := b - Ax_0; \ v_0 := r_0 / \|r_0\|_2; \ z_0 := v_0; \quad g_{0,0} := 1; \)
2: \textbf{for} \( i = 0, \ldots, m + l \) \textbf{do}
3: \quad \begin{aligned}
& z_{i+1} := \begin{cases} (A - \sigma_i I) z_i, & i < l \\ Az_i, & i \geq l \end{cases} \\
& \text{if} \ i \geq l \text{ then} \\
& \quad g_{j,i-l+1} := (g_{j,i-l+1} - \sum_{k=i-3l+1}^{j-1} g_{j,k} g_{k,i-l+1})/g_{j,j}; \\& \quad g_{i-l+1,i-l+1} := \sqrt{g_{i-l+1,i-l+1} - \sum_{k=i-3l+1}^{i-l} g_{k,i-l+1}^2}; \\
& \quad \text{# Check for breakdown and restart if required} \\
& \quad \text{if} \ i < 2l \text{ then} \\
& \quad \quad \gamma_{i-l} := (g_{i-l,i-l+1} + \sigma_{i-l} g_{i-l,i-l} - g_{i-l-1,i-l-1} \delta_{i-l-1})/g_{i-l,i-l}; \\
& \quad \quad \delta_{i-l} := g_{i-l+1,i-l+1}/g_{i-l,i-l}; \\
& \quad \text{else} \\
& \quad \quad \gamma_{i-l} := (g_{i-l,i-l} \gamma_{i-2l} + g_{i-l,i-l} \delta_{i-2l} - g_{i-l-1,i-l-1} \delta_{i-2l-1})/g_{i-l,i-l}; \\
& \quad \quad \delta_{i-l} := (g_{i-l+1,i-l+1} \delta_{i-2l})/g_{i-l,i-l}; \\
& \quad \text{end if} \\
& \quad \quad \psi_{i-l+1} := (z_{i-l+1} - \sum_{j=i-3l+1}^{i-l} g_{j,i-l+1} v_j)/g_{i-l+1,i-l+1}; \\
& \quad \quad z_{i+1} := (z_{i+1} - \gamma_{i-1} z_i - \delta_{i-1} z_{i-1})/\delta_{i-1}; \\
& \quad \text{end if} \\
& \quad g_{j,i+1} := \begin{cases} (z_{i+1}, v_j); & j = \max(0, i - 2l + 1), \ldots, i + l + 1 \\ (z_{i+1}, z_j); & j = i - l + 2, \ldots, i + 1 \end{cases} \\
& \quad \text{if} \ i = l \text{ then} \\
& \quad \quad \gamma_0 := \gamma_0; \quad \zeta_0 := \|r_0\|_2; \quad p_0 := v_0 / \gamma_0; \\
& \quad \text{else if} \ i \geq l + 1 \text{ then} \\
& \quad \quad \lambda_{i-l} := \delta_{i-l-1}/\eta_{i-l-1}; \\
& \quad \quad \eta_{i-l} := \gamma_{i-l} - \lambda_{i-l} \delta_{i-l-1}; \\
& \quad \quad \zeta_{i-l} := -\lambda_{i-l} \zeta_{i-l-1}; \\
& \quad \quad \pi_{i-l} := (\psi_{i-l} - \delta_{i-l} \pi_{i-l-1})/\eta_{i-l}; \\
& \quad \quad \xi_{i-l} := x_{i-l-1} + \zeta_{i-l} \pi_{i-l-1}; \\
& \quad \quad \text{if} \ |z_{i-l}|/\|r_0\| < \tau \text{ then RETURN; end if} \\
& \quad \text{end if} \\
& \text{end for}
Krylov subspace methods

Deep $\ell$-length pipelined CG

**Algorithm 1** $l$-length pipelined p($l$)-CG

**Input:** $A$, $b$, $x_0$, $l$, $m$, $\tau$

1: $r_0 := b - Ax_0$; $v_0 := r_0/\|r_0\|_2$; $z_0 := v_0$; $g_{0,0} := 1$
2: for $i = 0, \ldots, m + l$ do

3: \[
    z_{i+1} := \begin{cases} 
    (A - \sigma_i I)z_i, & i < l \\
    Az_i, & i \geq l 
    \end{cases}
\]

4: if $i \geq l$ then

5: \[
    g_{j,i-l+1} := (g_{j,i-l+1} - \sum_{k=i-3l+1}^{i-1} g_{k,j} g_{k,i-l+1})/g_{j,j}; \quad j = i - 2l + 2, \ldots, i
\]

6: \[
    g_{i-l+1,i-l+1} := \sqrt{g_{i-l+1,i-l+1} - \sum_{k=i-3l+1}^{i-l} g_{k,i-l+1}^2};
\]

7: $\#$ Check for breakdown and restart if required
8: if $i < 2l$ then

9: \[
    \gamma_{i-l} := (g_{i-l,i-l+1} + \sigma_{i-l} g_{i-l,i-l} - g_{i-l-1,i-l-1} \delta_{i-l-1})/g_{i-l,i-l};
\]

10: \[
    \delta_{i-l} := (g_{i-l+1,i-l+1} - g_{i-l,i-l})/g_{i-l,i-l};
\]

else

11: \[
    \gamma_{i-l} := (g_{i-l,i-l} \gamma_{i-2l} + g_{i-l,i-l+1} \delta_{i-2l} - g_{i-l-1,i-l-1} \delta_{i-l-1})/g_{i-l,i-l};
\]

12: \[
    \delta_{i-l} := (g_{i-l+1,i-l+1} \delta_{i-2l})/g_{i-l,i-l};
\]

end if

15: \[
    v_{i-l+1} := (z_{i-l+1} - \sum_{j=i-3l+1}^{i-l} g_{j,i-l+1} v_j)/g_{i-l+1,i-l+1};
\]

16: \[
    z_{i+1} := (z_{i+1} - \gamma_{i-l} z_i - \delta_{i-l} z_{i-l+1})/\delta_{i-l};
\]

end if

18: \[
    g_{j,i+1} := \begin{cases} 
    (z_{i+1}, v_j); & j = \max(0, i-2l+1), \ldots, i-l+1 \\
    (z_{i+1}, z_j); & j = i-l+2, \ldots, i+1
    \end{cases}
\]

19: if $i = l$ then

20: \[
    \eta_{0} := \gamma_{0}; \quad \zeta_{0} := \|r_0\|_2; \quad p_0 := v_0/\eta_{0};
\]

21: else if $i \geq l + 1$ then

22: \[
    \lambda_{i-l} := \delta_{i-l-1}/\eta_{i-l-1};
\]

23: \[
    \eta_{i-l} := \gamma_{i-l} - \lambda_{i-l} \delta_{i-l-1};
\]

24: \[
    \zeta_{i-l} := -\lambda_{i-l} \zeta_{i-l-1};
\]

25: \[
    p_{i-l} := (v_{i-l} - \delta_{i-l-1} p_{i-l-1})/\eta_{i-l};
\]

26: \[
    x_{i-l} := x_{i-l-1} + \zeta_{i-l-1} p_{i-l-1};
\]

27: if $|\zeta_{i-l}|/\|r_0\| < \tau$ then RETURN; end if

28: end if

29: end for

---

**SpMV (+ preconditioner)**
- 1 SpMV on $z_i$ per iteration

**Recurrence relations for $V_{i-1+l}$ and $Z_{i+1}$ basis vectors**
- computation: $2l + 2$ axpy’s
- storage: $3l + 2$ basis vectors

**Dot-products**
- $2l + 1$ band structure of $G_i$
- one global reduction phase is initiated per iteration
Krylov subspace methods

Deep $\ell$-length pipelined CG

Algorithm 1 $l$-length pipelined p($l$)-CG

\[ \begin{align*}
1: & \quad r_0 := b - Ax_0; \quad v_0 := r_0/\|r_0\|_2; \quad z_0 := v_0; \quad g_{0,0} := 1; \\
2: & \quad \text{for } i = 0, \ldots, m + l \text{ do} \\
3: & \quad z_{i+1} := \begin{cases} 
(A - \sigma_i I)z_i, & i < l \\
Az_i, & i \geq l 
\end{cases} \\
4: & \quad \text{if } i \geq l \text{ then} \\
5: & \quad g_{j,l+1} := (g_{j,l+1} - \sum_{k=1}^{j-1} g_{k,j}g_{k,l+1})/g_{j,l}; \\
6: & \quad g_{i-l+1,l+1} := \sqrt{g_{i-l+1,l+1} - \sum_{k=1}^{l-1} g_{k,l+1}^2}; \\
7: & \quad \# \text{ Check for breakdown and restart if required} \\
8: & \quad \text{if } i < 2l \text{ then} \\
9: & \quad \gamma_{l-i} := (g_{i-l+1} + \sigma_{l-i}g_{i-l+1})/g_{i-l+1}; \\
10: & \quad \delta_{i-l} := g_{i-l+1}; \\
11: & \quad \text{else} \\
12: & \quad \gamma_{l-i} := g_{l-i}; \\
13: & \quad \delta_{i-l} := g_{l-i}; \\
14: & \quad \text{end if} \\
15: & \quad \nu_{i-l+1} := (z_{i-l+1} - \sum_{j} v_j); \\
16: & \quad z_{i+1} := z_{i+1} - \gamma_{l-i}z_i; \\
17: & \quad \text{end if} \\
18: & \quad g_{i+l+1} := \begin{cases} 
(z_{i+1}, v_j); & j < l \\
(z_{i+1}, z_j); & j \geq l 
\end{cases} \\
19: & \quad \text{if } i \geq l \text{ then} \\
20: & \quad \eta_0 := \gamma_0; \quad \zeta_0 := \|r_0\|_2; \quad p_0 := v_0/\eta_0; \\
21: & \quad \text{else if } i \geq l + 1 \text{ then} \\
22: & \quad \lambda_{i-l} := \delta_{i-l}/\eta_{i-l}; \\
23: & \quad \eta_{i-l} := \gamma_{i-l} - \lambda_{i-l}\delta_{i-l}; \\
24: & \quad \zeta_{i-l} := -\lambda_{i-l}\zeta_{i-l}; \\
25: & \quad p_{i-l} := (v_{i-l} - \delta_{i-l}p_{i-l-1})/\eta_{i-l}; \\
26: & \quad x_{i-l} = x_{i-l-1} + \zeta_{i-l}p_{i-l-1}; \\
27: & \quad \text{if } \|z_{i-l}\|/\|r_0\| < \tau \text{ then RETURN; end if} \\
28: & \quad \text{end if} \\
29: & \quad \text{end for}
\end{align*} \]

Results of global sync. are needed $l$ iterations later to update $G_{i-l+1}$.

Each global reduction is overlapped by $\ell$ SpMVs.

Recurrence relations for $V_{i-l+1}$ and $Z_{i+1}$ basis vectors

- computation: $2l + 2$ axpy’s
- storage: $3l + 2$ basis vectors

Dot-products

- $2l + 1$ band structure of $G_i$
- one global reduction phase is initiated per iteration
Krylov subspace methods
Parallel performance of pipelined CG

Strong scaling experiments - PETSc 3.6.3/3.7.6 library - MPICH 3.1.3/3.3a2

Per node: Two 6-core Intel Xeon X5660 Nehalem 2.80 GHz - 2D Poisson (5pt) - 1 million unknowns

Per node: Two 14-core Intel E5-2680v4 Broadwell 2.40 GHz - 2D Poisson (5pt) - 3 million unknowns

Cornelis et al. (2018)
Pipelined Conjugate Gradients
Numerical stability in finite precision

Pipelined CG produces identical iterates to classic CG in exact arithmetic; but ...

Finite precision computations introduce roundoff errors that may lead to
1. *Delayed convergence* due to loss of basis orthogonality
2. *Loss of attainable accuracy* due to propagation of local rounding errors introduced by the recurrence relations
Rounding errors due to recurrence relations for residual and solution update:

\[ \tilde{x}_{i+1} = \bar{x}_i + \bar{\alpha}_i \bar{p}_i + \xi_{i+1}^x, \quad \tilde{r}_{i+1} = \bar{r}_i - \bar{\alpha}_i A \bar{p}_i + \xi_{i+1}^r, \]

Computed residual \( \tilde{r}_i \) deviates from the true residual \( b - A \bar{x}_i \) in finite precision:

\[
(b - A \tilde{x}_{i+1}) - \tilde{r}_{i+1} = b - A(\bar{x}_i + \bar{\alpha}_i \bar{p}_i + \xi_{i+1}^x) - (\bar{r}_i - \bar{\alpha}_i A \bar{p}_i + \xi_{i+1}^r) = \sum_{k=0}^{i+1} (A \xi_k^x + \xi_k^r).
\]

**Matrix notation:** \( \tilde{R}_{i+1} = [\tilde{r}_0, \ldots, \tilde{r}_i] \), \( \tilde{X}_{i+1} = [\bar{x}_0, \ldots, \bar{x}_i] \), \( \Theta_i^x, \Theta_i^r \) rounding errors

\[
(B - A \tilde{X}_{i+1}) - \tilde{R}_{i+1} = (A \Theta_{i+1}^x + \Theta_{i+1}^r) E_{i+1},
\]

with \( E_{i+1} \) an upper triangular matrix with all entries one.

**Accumulation of local rounding errors in classic CG, but no amplification.**
Pipelined Conjugate Gradients
Analyzing pipelined CG by Ghysels et al.

Additional recurrence relations in pipelined CG all introduce local rounding errors:

\[
\bar{x}_{i+1} = \bar{x}_i + \bar{\alpha}_i \bar{p}_i + \xi_{i+1}, \quad \bar{s}_i = \bar{w}_i + \bar{\beta}_i \bar{s}_{i-1} + \xi_s,
\]

\[
\bar{r}_{i+1} = \bar{r}_i - \bar{\alpha}_i \bar{s}_i + \xi_{i+1}, \quad \bar{w}_{i+1} = \bar{w}_i - \bar{\alpha}_i \bar{z}_i + \xi_w,
\]

\[
\bar{p}_i = \bar{r}_i + \bar{\beta}_i \bar{p}_{i-1} + \xi_p, \quad \bar{z}_i = A\bar{w}_i + \bar{\beta}_i \bar{z}_{i-1} + \xi_z,
\]

The gap on the residual is coupled to the gaps on the auxiliary variables:

\[
(B - A\bar{X}_i) - \bar{R}_i = (A\Theta_i^\bar{x} + \Theta_i^\bar{r}) E_i + (A\Theta_i^\bar{p} + \Theta_i^\bar{s}) \bar{B}_i^{-1} \bar{A}_i
\]

\[
+ (A\Theta_i^\bar{q} + \Theta_i^\bar{w}) E_i \bar{B}_i^{-1} \bar{A}_i + (A\Theta_i^\bar{q} + \Theta_i^\bar{z}) \bar{B}_i^{-1} \bar{A}_i \bar{B}_i^{-1} \bar{A}_i
\]

with \( \bar{A}_i = \begin{pmatrix} 0 & \bar{\alpha}_0 & \bar{\alpha}_0 & \cdots & \bar{\alpha}_0 \\ 0 & \bar{\alpha}_1 & \bar{\alpha}_1 & \cdots & \bar{\alpha}_1 \\ \vdots & \ddots & \ddots & \cdots & \vdots \\ 0 & \bar{\alpha}_{i-2} & 0 \\ 0 & 0 & \bar{\alpha}_{i-2} \end{pmatrix} \), \( \bar{B}_i^{-1} = \begin{pmatrix} 1 & \bar{\beta}_1 & \bar{\beta}_1 \bar{\beta}_2 & \cdots & \bar{\beta}_1 \bar{\beta}_2 \cdots \bar{\beta}_{i-1} \\ 1 & \bar{\beta}_2 & \bar{\beta}_2 \bar{\beta}_3 & \cdots & \bar{\beta}_2 \bar{\beta}_3 \cdots \bar{\beta}_{i-1} \\ \vdots & \ddots & \ddots & \cdots & \vdots \\ 1 & \bar{\beta}_{i-1} \\ 1 & 1 \end{pmatrix} \)

Remark: \( \beta_i \beta_i+1 \cdots \beta_j = \| r_j \|^2 / \| r_{i-1} \|^2 \), so entries of \( \bar{B}_i^{-1} \) may be arbitrarily large.
Additional recurrence relations in pipelined CG all introduce local rounding errors:

\[
\begin{align*}
\bar{x}_{i+1} &= \bar{x}_i + \bar{\alpha}_i \bar{p}_i + \xi_{i+1}, \\
\bar{r}_{i+1} &= \bar{r}_i - \bar{\alpha}_i \bar{s}_i + \xi_{i+1}, \\
\bar{p}_i &= \bar{r}_i + \bar{\beta}_i \bar{p}_{i-1} + \xi_i, \\
\bar{s}_i &= \bar{w}_i + \bar{\beta}_i \bar{s}_{i-1} + \xi_i, \\
\bar{w}_{i+1} &= \bar{w}_i - \bar{\alpha}_i \bar{z}_i + \xi_{i+1}, \\
\bar{z}_i &= A\bar{w}_i + \bar{\beta}_i \bar{z}_{i-1} + \xi_i.
\end{align*}
\]

The gap on the residual is coupled to the gaps on the auxiliary variables:

\[
(B - A\bar{X}_i) - \bar{R}_i = (A\Theta_i \bar{\bar{x}}_i + \Theta_i \bar{\bar{r}}_i) E_i + (A\Theta_i \bar{\bar{p}}_i + \Theta_i \bar{\bar{s}}_i) B_i \bar{\bar{B}}^{-1} \bar{\bar{A}}_i \\
+ (A\Theta_i \bar{\bar{w}}_i + \Theta_i \bar{\bar{w}}_i) E_i \bar{\bar{B}}^{-1} \bar{\bar{A}}_i + (A\Theta_i \bar{\bar{q}}_i + \Theta_i \bar{\bar{z}}_i) \bar{\bar{B}}^{-1} \bar{\bar{A}}_i \bar{\bar{B}}^{-1} \bar{\bar{A}}_i
\]

with \( \bar{\bar{A}}_i = \begin{pmatrix}
0 & \bar{\alpha}_0 & \bar{\alpha}_0 & \cdots & \bar{\alpha}_0 \\
0 & \bar{\alpha}_1 & \cdots & \bar{\alpha}_1 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \bar{\alpha}_{i-2} & \cdots & 0 \\
\end{pmatrix} \)

\( \bar{\bar{B}}^{-1} = \begin{pmatrix}
1 & \bar{\beta}_1 & \bar{\beta}_2 & \cdots & \bar{\beta}_{i-1} \\
1 & \bar{\beta}_2 & \cdots & \bar{\beta}_{i-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & \bar{\beta}_{i-1} & \cdots & 1 \\
\end{pmatrix} \)

**Amplification of local rounding errors possible, depending on values \( \bar{\alpha}_i \) and \( \bar{\beta}_i \).**
Analyzing deep $\ell$-length pipelined CG

The recurrence relations for $\bar{x}_i$ and $\bar{p}_i$ in finite precision $p(l)$-CG are

$$\bar{x}_i = \bar{x}_{i-1} + \zeta_{i-1} \bar{p}_{i-1} + \xi_i$$

$$\bar{p}_i = (\bar{v}_i - \bar{\delta}_{i-1} \bar{p}_{i-1})/\bar{\eta}_i + \xi_i^p$$

with $\bar{T}_i = \bar{L}_i \bar{U}_i$, implying the actual residual equals

$$b - A\bar{x}_i = \bar{r}_0 - A\bar{V}_i \bar{U}_i^{-1} \bar{q}_i + A\Xi_i^\bar{p} \bar{U}_i^{-1} \bar{q}_i - A\Xi_i^\bar{x} \mathbf{1} + \xi_0$$

$$= \bar{r}_0 - \bar{V}_{i+1} \bar{T}_{i+1,i} \bar{U}_i^{-1} \bar{q}_i - (A\bar{V}_i - \bar{V}_{i+1} \bar{T}_{i+1,i}) \bar{U}_i^{-1} \bar{q}_i + \text{LRE}$$

$$= \bar{r}_i - (A\bar{V}_i - \bar{V}_{i+1} \bar{T}_{i+1,i}) \bar{U}_i^{-1} \bar{q}_i + \text{LRE}$$

Computed residual tends to zero

Inexact Lanczos relation ("gap on $\bar{V}_{i+1}$") determines maximal attainable accuracy
Basis vector recurrences in finite precision \( p(l) \)-CG

\[
\bar{v}_{i+1} = \left( \bar{z}_{i+1} - \sum_{j=i-2l+1}^{i} \bar{g}_{j,i+1} \bar{v}_i \right) / \bar{g}_{i+1,i+1} + \xi_{i+1}^{\bar{v}}, \quad \Leftrightarrow \quad \bar{Z}_i = \bar{V}_i \bar{G}_i + \Xi_i^{\bar{v}} \quad (1)
\]

\[
\bar{z}_{i+1} = (A\bar{Z}_i - \bar{\gamma}_{i-1} \bar{Z}_i - \bar{\delta}_{i-l} \bar{Z}_{i-1}) / \bar{\delta}_{i-l} + \xi_{i+1}^{\bar{z}}, \quad \Leftrightarrow \quad A\bar{Z}_i = \bar{Z}_{i+1} \bar{B}_{i+1,i} + \Xi_i^{\bar{z}} \quad (2)
\]

and the finite precision coefficient relation \( \bar{G}_{i+1} \bar{B}_{i+1,i} = \bar{T}_{i+1,i} \bar{G}_i \) allow to compute the gap on the basis \( \bar{V}_{i+1} \) as

\[
A\bar{V}_i - \bar{V}_{i+1} \bar{T}_{i+1,i} \overset{(1)}{=} A\bar{Z}_i \bar{G}_i^{-1} - \bar{Z}_{i+1} \bar{G}_{i+1,i}^{-1} \bar{T}_{i+1,i} - A\Xi_i^{\bar{v}} \bar{G}_i^{-1} + \Xi_{i+1}^{\bar{v}} \bar{G}_{i+1,i}^{-1} \bar{T}_{i+1,i} \]

\[
\overset{(2)}{=} (A\bar{Z}_i - \bar{Z}_{i+1} \bar{B}_{i+1,i} - A\Xi_i^{\bar{v}} + \Xi_{i+1}^{\bar{v}} \bar{B}_{i+1,i}) \bar{G}_i^{-1} \]

\[
\overset{(3)}{=} (\Xi_i^{\bar{z}} - A\Xi_i^{\bar{v}} + \Xi_{i+1}^{\bar{v}} \bar{B}_{i+1,i}) \bar{G}_i^{-1}.
\]

**Amplification of local rounding errors possible, depending on \( \bar{G}_i^{-1} \).**

Cornelis et al. (2018)
Pipelined Conjugate Gradients

Analyzing deep $\ell$-length pipelined CG

- The norm $\|\tilde{G}^{-1}_i\|_{\text{max}}$ quantifies the impact of rounding error amplification on attainable accuracy in p($\ell$)-CG.

- The Cholesky factorization $Z_i^T Z_i = G_i^T G_i$ relates the conditioning of $G_i$ and the auxiliary basis $Z_i$; numerical stability depends on the polynomial $P_{\ell}(A)$.

Countermeasures against error propagation

Residual replacement in p-CG by Ghysels et al.

- Replace $\bar{r}_i = \text{fl}(b - A\bar{x}_i)$, $\bar{w}_i = \text{fl}(A\bar{r}_i)$, $\bar{s}_i = \text{fl}(A\bar{p}_i)$, $\bar{z}_i = \text{fl}(A\bar{s}_i)$ in selected iterations
- Automated procedure based on estimate $\|b - A\bar{x}_i - \bar{r}_i\|$ (computed inexpensively)
  - Replace sufficiently often such that residual gap remains small
  - Don’t replace if $\|\bar{r}_i\|$ is small, which may cause delay of convergence

Speedup over single-node CG (12-240 cores)  Accuracy vs. total time spent (240 cores)
Countermeasures against error propagation

Stable recurrences for deep $\ell$-length pipelined CG

Introduce $l$ auxiliary bases

$$Z^{(0)}_{i+1} = [v_0, \ldots v_i], \quad Z^{(1)}_{i+1} = [z^{(1)}_0, \ldots z^{(1)}_i], \quad \ldots , \quad Z^{(l)}_{i+1} = [z_0, \ldots z_i],$$

and replace the multi-term recurrence relation for $v_{i-l+1}$ ($\sim 2l$ terms) by $l + 1$ coupled three-term recurrence relations

$$\begin{align*}
v_{i-l+1} &= (z^{(1)}_{i-l+1} + (\sigma_0 - \gamma_{i-l})v_{i-l} - \delta_{i-l-1}v_{i-l-1})/\delta_{i-l}, \\
z^{(1)}_{i-l+2} &= (z^{(2)}_{i-l+2} + (\sigma_1 - \gamma_{i-l})z^{(1)}_{i-l+1} - \delta_{i-l-1}z^{(1)}_{i-l})/\delta_{i-l}, \\
&\vdots \\
z^{(l-1)}_{i} &= (z_{i} + (\sigma_{l-1} - \gamma_{i-l})z^{(l-1)}_{i-1} - \delta_{i-l-1}z^{(l-1)}_{i-2})/\delta_{i-l}, \\
z_{i+1} &= (Az_{i} - \gamma_{i-l}z_{i} - \delta_{i-l-1}z_{i-1})/\delta_{i-l}. & \quad \leftarrow \text{1 SpMV}
\end{align*}$$

This modification causes (almost) no overhead

- the computational cost (#SpMVs and #axpy’s) is identical to before,
- the storage cost increases by only $l - 2$ vectors.
Introduce $l$ auxiliary bases

$$
\bar{Z}_{i+1}^{(0)} = [\bar{v}_0, \ldots, \bar{v}_i], \quad \bar{Z}_{i+1}^{(1)} = [\bar{Z}_0^{(1)}, \ldots, \bar{Z}_i^{(1)}], \quad \ldots, \quad \bar{Z}_{i+1}^{(l)} = [\bar{Z}_0, \ldots, \bar{Z}_i],
$$

and replace the multi-term recurrence relation for $\bar{v}_{i-l+1}$ ($\sim 2l$ terms) by $l + 1$ coupled three-term recurrence relations that all introduce local rounding errors

$$
\begin{align*}
\bar{v}_{i-l+1} &= (\bar{Z}_{i-l+1}^{(1)} + (\sigma_0 - \bar{\gamma}_{i-l})\bar{v}_{i-l} - \bar{\delta}_{i-l-1} \bar{v}_{i-l-1})/\bar{\delta}_{i-l} + \xi_{i-l+1}^{(0)}, \\
\bar{Z}_{i-l+2}^{(1)} &= (\bar{Z}_{i-l+2}^{(2)} + (\sigma_1 - \bar{\gamma}_{i-l})\bar{Z}_{i-l+1}^{(1)} - \bar{\delta}_{i-l-1} \bar{Z}_{i-l}^{(1)})/\bar{\delta}_{i-l} + \xi_{i-l+2}^{(1)}, \\
&\vdots \\
\bar{Z}_{i}^{(l-1)} &= (\bar{Z}_{i} + (\sigma_{l-1} - \bar{\gamma}_{i-l})\bar{Z}_{i-l}^{(l-1)} - \bar{\delta}_{i-l-1} \bar{Z}_{i-2}^{(l-1)})/\bar{\delta}_{i-l} + \xi_{i}^{(l-1)}, \\
\bar{Z}_{i+1} &= (A\bar{Z}_{i} - \bar{\gamma}_{i-l} \bar{Z}_{i} - \bar{\delta}_{i-l-1} \bar{Z}_{i-1})/\bar{\delta}_{i-l} + \xi_{i+1}^{(l)}.
\end{align*}
$$
Countermeasures against error propagation

Stable recurrences for deep $\ell$-length pipelined CG

INFINITE PRECISION ARITHMETIC

Introduce $l$ auxiliary bases

\[
\bar{Z}_{i+1}^{(0)} = [\bar{v}_0, \ldots, \bar{v}_i], \quad \bar{Z}_{i+1}^{(1)} = [\bar{z}_0^{(1)}, \ldots, \bar{z}_i^{(1)}], \quad \ldots, \quad \bar{Z}_{i+1}^{(l)} = [\bar{z}_0, \ldots, \bar{z}_i],
\]

and replace the multi-term recurrence relation for $\bar{v}_{i-l+1}$ ($\sim 2l$ terms) by $l + 1$ coupled three-term recurrence relations that are written in matrix notation as

\[
\begin{align*}
\bar{Z}_{2:i-l+1}^{(1)} &= \bar{Z}_{i-l+1}^{(0)} \bar{T}_{i-l+1,i-1} - \sigma_0 \bar{Z}_{i-l}^{(0)} - \Xi_{i-l+1}^{(0)} \bar{\Delta}_{i-l+1,i-1}, \\
\bar{Z}_{2:i-l+2}^{(2)} &= \bar{Z}_{i-l+2}^{(1)} \bar{T}_{i-l+2,i-1} - \sigma_1 \bar{Z}_{i-l+1}^{(1)} - \Xi_{i-l+2}^{(1)} \bar{\Delta}_{i-l+2,i-1}, \\
&\vdots \\
\bar{Z}_{2:i}^{(l)} &= \bar{Z}_{i-1}^{(l-1)} \bar{T}_{i-1,i-1} - \sigma_{l-1} \bar{Z}_{i-1}^{(l-1)} - \Xi_{i-1}^{(l-1)} \bar{\Delta}_{i,i-1}, \\
A\bar{Z}_{i}^{(l)} &= \bar{Z}_{i+1}^{(l)} \bar{T}_{i+1,i} - \Xi_{i+1}^{(l)} \bar{\Delta}_{i+1,i}.
\end{align*}
\]

This modification causes very limited overhead

$\longrightarrow$ computational cost (= SpMVs and axpy's) is identical to before,

$\longrightarrow$ storage costs increase slightly, but only by $l - 2$ vectors.
Countermeasures against error propagation

Stable recurrences for deep ℓ-length pipelined CG

in finite precision arithmetic

Introduce \( l \) auxiliary bases

\[
\tilde{Z}_{i+1}^{(0)} = [\tilde{v}_0, \ldots \tilde{v}_i], \quad \tilde{Z}_{i+1}^{(1)} = [\tilde{z}_0^{(1)}, \ldots \tilde{z}_i^{(1)}], \quad \ldots \quad \tilde{Z}_{i+1}^{(l)} = [\tilde{z}_0, \ldots \tilde{z}_i],
\]

and replace the multi-term recurrence relation for \( \tilde{v}_{i-l+1} \) (\( \sim 2l \) terms) by \( l+1 \) coupled three-term recurrence relations that are written in matrix notation as

\[
\begin{align*}
\tilde{Z}_{2:i-l+1}^{(1)} &= \tilde{Z}_{i-l+1}^{(0)} \tilde{T}_{i-l+1,i-l} - \sigma_0 \tilde{Z}_{i-l}^{(0)} - \Xi_{i-l+1,i-l}, \\
\tilde{Z}_{2:i-l+2}^{(2)} &= \tilde{Z}_{i-l+2}^{(1)} \tilde{T}_{i-l+2,i-l+1} - \sigma_1 \tilde{Z}_{i-l+1}^{(1)} - \Xi_{i-l+2,i-l+1}, \\
&\vdots \\
\tilde{Z}_{2:i}^{(l)} &= \tilde{Z}_{i}^{(l-1)} \tilde{T}_{i,i-1} - \sigma_{l-1} \tilde{Z}_{i-1}^{(l-1)} - \Xi_{i}^{(l-1)} \Delta_{i,i-1}, \\
A\tilde{Z}_{i}^{(l)} &= \tilde{Z}_{i+1}^{(l)} \tilde{T}_{i+1,i} - \Xi_{i+1}^{(l)} \Delta_{i+1,i}.
\end{align*}
\]

For \( \tilde{Z}_{i+1}^{(l)} \) the gap is given by

\[
A\tilde{Z}_{i}^{(l)} - \tilde{Z}_{i+1}^{(l)} \tilde{T}_{i+1,i} = -\Xi_{i+1}^{(l)} \Delta_{i+1,i}
\]
Introduce $l$ auxiliary bases

$$Z_{i+1}^{(0)} = [\bar{v}_0, \ldots \bar{v}_i], \quad Z_{i+1}^{(1)} = [\bar{z}_0^{(1)}, \ldots \bar{z}_i^{(1)}], \quad \ldots \quad , \quad Z_{i+1}^{(l)} = [\bar{z}_0, \ldots \bar{z}_i],$$

and replace the multi-term recurrence relation for $\bar{v}_{i-1} (\sim 2l$ terms) by $l + 1$ coupled three-term recurrence relations that are written in matrix notation as

$$\begin{cases}
Z_{2:i-1+l+1}^{(1)} = Z_{i-l+1}^{(0)} T_{i-l+1,i-l} - \sigma_0 Z_{i-l+1}^{(0)} - \Xi_{i-l+1,i-l}, \\
Z_{2:i-1+l+2}^{(2)} = Z_{i-l+2}^{(1)} T_{i-l+2,i-l+1} - \sigma_1 Z_{i-l+1}^{(1)} - \Xi_{i-l+2,i-l+1}, \\
\vdots \\
Z_{2:i}^{(l)} = Z_{i-l}^{(l-1)} T_{i,i-1} - \sigma_{l-1} Z_{i-1}^{(l-1)} - \Xi_{i,i-1}, \\
A Z_{i}^{(l)} = Z_{i+1}^{(l)} T_{i+1,i} - \Xi_{i+1,i} \bigl( \Delta_{i+1,i} \bigr).
\end{cases}$$

For $\bar{Z}_{i+1}^{(l-1)}$ the gap is given by $\bar{\Delta}_{i+1,i}$ diagonal matrix

$$A \bar{Z}_{i}^{(l-1)} - \bar{Z}_{i+1}^{(l-1)} \bar{T}_{i+1,i} = (A \bar{Z}_{i}^{(l)} - \bar{Z}_{i+1}^{(l)} \bar{T}_{i+1,i}) \bar{\Delta}_{i}^{+} + \Xi_{i} - \Xi_{i+1}^{(l-1)} \bar{\Delta}_{i+1,i}.$$
Countermeasures against error propagation

Stable recurrences for deep $\ell$-length pipelined CG

INFINITE PRECISION ARITHMETIC

Introduce $l$ auxiliary bases

$$
\bar{Z}^{(0)}_{i+1} = [\bar{v}_0, \ldots, \bar{v}_i], \quad \bar{Z}^{(1)}_{i+1} = [\bar{z}^{(1)}_{i+1}, \ldots, \bar{z}^{(1)}_i], \quad \ldots, \quad \bar{Z}^{(l)}_{i+1} = [\bar{z}_0, \ldots, \bar{z}_i],
$$

and replace the multi-term recurrence relation for $\bar{v}_{i-l+1}$ ($\sim 2l$ terms) by $l+1$ coupled three-term recurrence relations that are written in matrix notation as

$$
\begin{aligned}
\bar{Z}^{(1)}_{2:i-l+1} &= \bar{Z}^{(0)}_{i-l+1} \bar{T}_{i-l+1,i-l} - \sigma_0 \bar{Z}^{(0)}_{i-l} - \Xi^{(0)}_{i-l+1} \bar{\Delta}_{i-l+1,i-l}, \\
\bar{Z}^{(2)}_{2:i-l+2} &= \bar{Z}^{(1)}_{i-l+2} \bar{T}_{i-l+2,i-l+1} - \sigma_1 \bar{Z}^{(1)}_{i-l+1} - \Xi^{(1)}_{i-l+2} \bar{\Delta}_{i-l+2,i-l+1}, \\
&\vdots \\
\bar{Z}^{(l)}_{2:i} &= \bar{Z}^{(l-1)}_{i,i-1} \bar{T}_{i,i-1} - \sigma_{l-1} \bar{Z}^{(l-1)}_{i-1} - \Xi^{(l-1)}_{i} \bar{\Delta}_{i,i-1}, \\
A \bar{Z}^{(l)}_{i+1} &= \bar{Z}^{(l)}_{i+1} \bar{T}_{i+1,i} - \Xi^{(l)}_{i+1} \bar{\Delta}_{i+1,i}.
\end{aligned}
$$

For general $\bar{Z}^{(k)}_{i+1}$ the gap is given by $k \in \{0, 1, \ldots, l - 1\}$

$$
A \bar{Z}^{(k)}_{i} - \bar{Z}^{(k)}_{i+1} \bar{T}_{i+1,i} = (A \bar{Z}^{(k+1)}_{i} - \bar{Z}^{(k+1)}_{i+1} \bar{T}_{i+1,i}) \bar{\Delta}^+_{i,i} + \Xi^{(k+1)}_{i} - \Xi^{(k)}_{i+1} \bar{\Delta}_{i+1,i}.
$$
Introduce \( l \) auxiliary bases

\[
\bar{Z}_{i+1}^{(0)} = [\bar{v}_0, \ldots, \bar{v}_i], \quad \bar{Z}_{i+1}^{(1)} = [\bar{z}_0^{(1)}, \ldots, \bar{z}_i^{(1)}], \quad \ldots, \quad \bar{Z}_{i+1}^{(l)} = [\bar{z}_0, \ldots, \bar{z}_i],
\]

and replace the multi-term recurrence relation for \( \bar{v}_{i-l+1} \) (\( \sim 2l \) terms) by \( l + 1 \) coupled three-term recurrence relations that are written in matrix notation as

\[
\begin{align*}
\bar{Z}_{2:i-l+1}^{(1)} & = \bar{Z}_{i-l+1}^{(0)} \bar{T}_{i-l+1,i-l} - \sigma_0 \bar{Z}_{i-l+1}^{(0)} - \Xi_{i-l+1,i-l+1}^{(0)} \Delta_{i-l+1,i-l}, \\
\bar{Z}_{2:i-l+2}^{(2)} & = \bar{Z}_{i-l+2}^{(1)} \bar{T}_{i-l+2,i-l+1} - \sigma_1 \bar{Z}_{i-l+2}^{(1)} - \Xi_{i-l+2,i-l+1}^{(1)} \Delta_{i-l+2,i-l+1}, \\
& \vdots \\
\bar{Z}_{2:i}^{(l)} & = \bar{Z}_{i}^{(l-1)} \bar{T}_{i,i-1} - \sigma_{l-1} \bar{Z}_{i-1}^{(l-1)} - \Xi_{i}^{(l-1)} \Delta_{i,i-1}, \\
\bar{A} \bar{Z}_{i}^{(l)} & = \bar{Z}_{i+1}^{(l)} \bar{T}_{i+1,i} - \Xi_{i+1,i}^{(l)} \Delta_{i+1,i}.
\end{align*}
\]

**Accumulation of local rounding errors, but no amplification, similar to classic CG. The method thus attains the same accuracy as classic CG!**

---

C. et al. (2019)
Countermeasures against error propagation

Stable recurrences for deep $\ell$-length pipelined CG

Classic CG

![Graph showing residual norm vs iterations for Classic CG](image-url)
Countermeasures against error propagation

Stable recurrences for deep $\ell$-length pipelined CG

Pipelined CG by Ghysels et al.
Countermeasures against error propagation
Stable recurrences for deep $\ell$-length pipelined CG

Pipelined p(1)-CG “unstable”
Countermeasures against error propagation

Stable recurrences for deep $\ell$-length pipelined CG

Pipelined p(2)-CG “unstable”
Countermeasures against error propagation

Stable recurrences for deep $\ell$-length pipelined CG
Countermeasures against error propagation

Stable recurrences for deep $\ell$-length pipelined CG

Pipelined p(5)-CG “unstable”
Countermeasures against error propagation

Stable recurrences for deep $\ell$-length pipelined CG

Pipelined p(1)-CG “stable”
Countermeasures against error propagation

Stable recurrences for deep $\ell$-length pipelined CG
Countermeasures against error propagation

Stable recurrences for deep $\ell$-length pipelined CG

Pipelined p(3)-CG “stable”
Countermeasures against error propagation
Stable recurrences for deep $\ell$-length pipelined CG

Pipelined p(5)-CG “stable”
Numerical experiments

Deep $\ell$-length pipelined CG

- Strong scaling on up to 32 14-core Intel E5-2680v4 Broadwell CPU nodes
- EDR Infiniband, Intel MPI 2018v3, PETSc v3.8.3, KSP ex2
- 2D 5-pt Poisson, 3 million unknowns, 1,500 iterations, no preconditioner

### Speedup (over CG on 1 node)

![Speedup graph](image)

### Accuracy (vs. total CPU time)

![Accuracy graph](image)
Numerical experiments

Deep $\ell$-length pipelined CG

- Strong scaling on up to 128 14-core Intel E5-2680v4 Broadwell CPU nodes
- EDR Infiniband, Intel MPI 2018v3, PETSc v3.8.3, SNES ex48
- 3D Hydrostatic Ice Sheet Flow, 2.25 million FE, Newton-Krylov solver, 7 Newton steps, 4,500 total inner iter, block Jacobi preconditioner, inner tolerance: $1.0\times10^{-10}$, outer tolerance: $1.0\times10^{-8}$

### Speedup (over CG on 1 node)

![Graph showing speedup over CG on 1 node](image)
Numerical experiments

Deep $\ell$-length pipelined CG

- Strong scaling on up to 128 14-core Intel E5-2680v4 Broadwell CPU nodes
- EDR Infiniband, Intel MPI 2018v3, PETSc v3.8.3, SNES ex48
- 3D Hydrostatic Ice Sheet Flow, 2.25 million FE, Newton-Krylov solver, 7 Newton steps, 4,500 total inner iter, block Jacobi preconditioner, inner tolerance: $1.0 \times 10^{-10}$, outer tolerance: $1.0 \times 10^{-8}$

Accuracy (vs. total number of inner iterations)
Conclusions

Takeaway messages

- Pipelined Krylov subspace methods are a promising approach
  - *Hide communication latency* behind computational kernels by adding auxiliary variables and recurrence relations
  - \( p(\ell)\)-CG: Deep pipelines allow to hide global reduction phases behind multiple SpMV’s/iterations
  - *Asynchronous implementation*: dot-products can take multiple iterations to complete; global reductions are implemented in an overlapping manner
  - *Improved scaling* over classic KSMs in strong scaling limit, where global reduction latencies rise and volume of computations per core diminishes

- The finite precision behavior of communication avoiding- and hiding Krylov subspace algorithms should be carefully monitored
  - *Local rounding error analysis* allows to explain loss of attainable accuracy

- Insights to construct a more stable method are obtained from the analysis
  - *Fully restore attainable accuracy* in \( p(I)\)-CG at *no increase in computational costs or storage costs* through residual replacement-type techniques
  - The issue of *loss of orthogonality* has not been addressed by the modifications to \( p(I)\)-CG proposed in this talk
Conclusions

Contributions to PETSc

Open source HPC linear algebra toolkit: https://www.mcs.anl.gov/petsc/

- KSPPGMRES: pipelined GMRES (thanks to J. Brown)
- KSPPPIPECG: pipelined Conjugate Gradients
- KPPPPPIPECGR: pipelined Conjugate Residuals
- KSPPPIPECGRR: pipelined CG with automated residual replacement
- KSPPPIPELCEG: pipelined CG with deep pipelines
- KSPPGROPPCGB: asynchronous CG variant by W. Gropp and collaborators
- KSPPPIPEBCGS: pipelined BiCGStab

We are soliciting for feedback from your applications!
Related publications

- siegfried.cools@uantwerp.be

- jeffrey.cornelis@uantwerp.be

- wim.vanroose@uantwerp.be

Thank you!


