A Residual Replacement Strategy for Improving the Maximum Attainable Accuracy of Communication-Avoiding Krylov Subspace Methods

Erin Carson
James Demmel

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-44
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-44.html

April 20, 2012

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **20 APR 2012**	2. REPORT TYPE	3. DATES COVERED **00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE **A Residual Replacement Strategy for Improving the Maximum Attainable Accuracy of Communication-Avoiding Krylov Subspace Methods**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California, Berkeley,Department of Electrical Engineering and Computer Sciences,Berkeley,CA,94720**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The behavior of conventional Krylov Subspace Methods (KSMs) in finite precision arithmetic is a well-studied problem. The finite precision Lanczos process, which drives convergence of these methods, can lead to a significant deviation between the recursively computed residual and the true residual, b &#56256;&#56320; Axk, decreasing the maximum attainable accuracy of the solution. Van der Vorst and Ye [24] have advocated the use of a residual replacement strategy for KSMs to prevent the accumulation of this error, in which the computed residual is replaced by the true residual at specific iterations chosen such that the Lanczos process is undisturbed. Recent results have demonstrated the performance benefits of Communication-Avoiding Krylov Subspace Methods (CA-KSMs), variants of KSMs which use blocking strategies to perform s computation steps of the algorithm for each communication step. This allows an O(s) reduction in total communication cost, which can lead to significant speedups on modern computer architectures. Despite the potential performance benefits of CA-KSMs, the finite precision error in these variants grows with s, an obstacle for their use in practice. Following the work of Van der Vorst and Ye , we bound the deviation of the true and computed residual in finite precision CA-KSMs, which leads to an implicit residual replacement strategy. We are the first, to our knowledge to perform this analysis for CA-KSMs. We show how to implement our strategy without affecting the asymptotic communication or computation cost of the algorithm. Numerical experiments demonstrate the effectiveness of our residual replacement strategy for both CA-CG and CA-BICG. Specifically, it is shown that accuracy of order O( )jjAjj jjxjj can be achieved with a small number of residual replacement steps for an appropriately chosen polynomial basis, which demonstrates the potential for practical use of CA-KSMs.

15. SUBJECT TERMS

<table>
<tr><td colspan="3">16. SECURITY CLASSIFICATION OF:</td><td rowspan="2">17. LIMITATION OF ABSTRACT
Same as Report (SAR)</td><td rowspan="2">18. NUMBER OF PAGES
28</td><td rowspan="2">19a. NAME OF RESPONSIBLE PERSON</td></tr>
<tr><td>a. REPORT
unclassified</td><td>b. ABSTRACT
unclassified</td><td>c. THIS PAGE
unclassified</td></tr>
</table>

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Acknowledgement

This paper was researched with Government support under and awarded by the Department of Defense, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. This research was also supported by Microsoft (Award #024263) and Intel (Award #024894) funding and by matching funding by U.C. Discovery (Award #DIG07-10227). Additional support comes from Par Lab affiliates National Instruments, Nokia, NVIDIA, Oracle, and Samsung, U.S. DOE grants DE-SC0003959, DE-AC02-05-CH11231, Lawrence Berkeley National Laboratory, and NSF SDCI under Grant Number OCI-1032639.

A Residual Replacement Strategy for Improving the Maximum Attainable Accuracy of Communication-Avoiding Krylov Subspace Methods

Erin Carson and James Demmel

Abstract

The behavior of conventional Krylov Subspace Methods (KSMs) in finite precision arithmetic is a well-studied problem. The finite precision Lanczos process, which drives convergence of these methods, can lead to a significant deviation between the recursively computed residual and the true residual, $b - Ax^k$, decreasing the maximum attainable accuracy of the solution. Van der Vorst and Ye [24] have advocated the use of a residual replacement strategy for KSMs to prevent the accumulation of this error, in which the computed residual is replaced by the true residual at specific iterations chosen such that the Lanczos process is undisturbed.

Recent results have demonstrated the performance benefits of Communication-Avoiding Krylov Subspace Methods (CA-KSMs), variants of KSMs which use blocking strategies to perform s computation steps of the algorithm for each communication step. This allows an $O(s)$ reduction in total communication cost, which can lead to significant speedups on modern computer architectures. Despite the potential performance benefits of CA-KSMs, the finite precision error in these variants grows with s, an obstacle for their use in practice.

Following the work of Van der Vorst and Ye , we bound the deviation of the true and computed residual in finite precision CA-KSMs, which leads to an implicit residual replacement strategy. We are the first, to our knowledge, to perform this analysis for CA-KSMs. We show how to implement our strategy without affecting the asymptotic communication or computation cost of the algorithm. Numerical experiments demonstrate the effectiveness of our residual replacement strategy for both CA-CG and CA-BICG. Specifically, it is shown that accuracy of order $O(\epsilon)||A|| \cdot ||x||$ can be achieved with a small number of residual replacement steps for an appropriately chosen polynomial basis, which demonstrates the potential for practical use of CA-KSMs.

1 Introduction

Krylov subspace methods (KSMs) are a class of iterative algorithms commonly used to solve linear systems. These methods work by iteratively adding a dimension to a Krylov subspace and then choosing the “best” solution from the resulting subspace. In terms of linear algebra, these operations consist of one or more sparse matrix-vector multiplications (SpMVs) and vector operations in each iteration, where the solution x^k and residual r^k are updated as

$$x^k = x^{k-1} + \alpha_{k-1} p^{k-1} \quad r^k = r^{k-1} - \alpha_{k-1} A p^{k-1} \tag{1}$$

or something similar. This encompasses algorithms such as Conjugate Gradient (CG), steepest descent, Biconjugate Gradient (BICG), Conjugate Gradient Squared (CGS), and Stabilized Biconjugate Gradient (BICGSTAB).

It is important to notice that x^k and r^k have different round-off patterns. That is, the expression for x^k does not depend on r^k, nor does the expression for r^k depend on x^k. Therefore, computational errors made in x^k are not self-correcting. Throughout the iteration, these errors accumulate, and cause deviation of the true residual, $b - Ax^k$, and computed residual, r^k. This limits the *maximum attainable accuracy*, which indicates how accurately we can solve the system on a computer with machine precision ϵ. When the algorithm reaches this maximum attainable accuracy, the computed residual will appear to continue decreasing in norm, whereas the norm of the true residual stagnates. This can lead to a very large error in the solution despite the algorithm reporting a very small residual norm.

This has motivated the use of strategies such as restarting and residual replacement to limit the error that accumulates throughout the computation (see, e.g., [20, 24]). The solution is not as simple as using the true residual in every iteration (or even every s iterations). In addition to increasing both the communication and computation in the method, replacing the recursively computed residual with the true residual can destroy the super-linear convergence properties exhibited by KSMs, as it is these recurrences which drive the Lanczos process [20, 24]. Residual replacement strategies must then carefully select iterations where residual replacement takes place, which requires estimating the accrued rounding error. Van der Vorst and Ye have successfully implemented such a strategy for standard Krylov methods [24].

The computation that occurs in each iteration in standard Krylov methods, namely, the updates of x^k and r^k, consist of one or more SpMV and vector operations. Because there are dependencies between iterations in standard Krylov methods and the main kernels in each iteration have low computation to communication ratios, standard Krylov method implementations are communication-bound on modern computer architectures.

This motivated $s-$step, or, Communication-Avoiding KSMs (CA-KSMs), which are equivalent to the standard KSM implementations in exact arithmetic. These variants use blocking strategies to perform s computation steps of the algorithm for each communication step, allowing an $O(s)$ reduction in total communication cost (see, e.g., [2, 3, 4, 6, 7, 10, 11, 16, 19, 21, 22]). Despite attractive performance benefits, these variants are often considered impractical, as increased error in finite precision can negatively affect stability. The deviation of the true and computed residual observed in standard KSMs is worse for CA-KSMs, with the upper bound depending on s. Although many previous authors have observed this behavior in CA-KSMs (see, e.g. [3, 5, 4, 11, 19, 25]), we are the first, to our knowledge, to provide a quantitative analysis of round-off error in these algorithms which limits the maximum attainable accuracy. Our analysis, which follows the analysis for standard KSMs in [24], leads directly to an implicit residual replacement strategy to reduce such error.

Our numerical experiments suggest that, for solving $Ax = b$, if the corresponding standard method with residual replacement converges such that the norm of the true residual is $O(\epsilon||A|| \cdot ||x||)$, and all $(s + 1)$-dimensional Krylov bases generated in our CA-KSM are numerically full rank, our methods will also converge with norm of the true residual equal to $O(\epsilon||A|| \cdot ||x||)$ when our residual replacement strategy is employed. Furthermore, we note that if we generate the Krylov basis using properly chosen Newton or Chebyshev polynomials, the norm of the basis grows slowly with s. Therefore, the number of residual replacement steps for these bases will generally grow slowly with respect to the total number of iterations, and we claim that stability in CA-KSMs can be achieved with *no asymptotic increase in the communication cost of s steps.*

1.1 Related Work

We briefly discuss related work in the areas of s-step and CA-KSMs, as well as work related to the numerical analysis of standard KSMs.

1.1.1 s-step Krylov Subspace Methods

The first instance of an $s-$step method in the literature is Van Rosendale's conjugate gradient method [19]. Van Rosendale's implementation was motivated by exposing more parallelism using the PRAM model. Chronopoulous and Gear later created an $s-$step GMRES method with the goal of exposing more parallel optimizations [5]. Walker looked into s-step bases as a method for improving stability in GMRES by replacing the modified Gram-Schmidt orthogonalization process with Householder QR [25]. All these authors used the monomial basis, and found that convergence often could not be guaranteed for $s > 5$. It was later discovered that this behavior was due to the inherent instability of the monomial basis, which motivated research into the use of other bases for the Krylov Subspace.

Hindmarsh and Walker used a scaled (normalized) monomial basis to improve convergence [10], but only saw minimal improvement. Joubert and Carey implemented a scaled and shifted Chebyshev basis which provided more accurate results [12]. Bai et al. also saw improved convergence using a Newton basis [1]. Although successively scaling the basis vectors serves to lower the condition number of the basis matrix, hopefully yielding convergence closer to that of the standard method, this computation reintroduces the dependency we sought to remove, hindering communication-avoidance. Hoemmen resolves this problem using a novel matrix equilibration and balancing approach as a preprocessing step, which eliminates the need for scaled basis vectors [11].

Hoemmen et. al [6, 11, 16] have derived Communication-Avoiding variants of Lanczos, Arnoldi, CG and GMRES. The derivation of Communication-Avoiding variants of two-sided Krylov subspace methods, such as BICG, CGS, and BICGSTAB can be found in [2].

1.1.2 Error Analysis of Krylov Subspace Methods

An upper bound on the maximum attainable accuracy for KSMs was provided by Greenbaum [9]. Greenbaum proved that this bound can be given a priori for methods like CG, but can not be predetermined for methods like BICG, which can have large intermediate iterates. Additionally, Greenbaum has shown the backward stability of the CG algorithm, by showing that the Ritz values found lie in small intervals around the eigenvalues of A. There are many other analyses of the behavior of various KSMs in finite precision arithmetic (see, e.g. [14, 15, 23]). The reader is also directed to the bibliography in [17].

Sleijpen and Van der Vorst implemented a technique called "flying restarts" to decrease the amount of round-off error that occurs in KSMs [20]. Their method, which is applicable to many KSMs, iteratively tracks an upper bound for the amount of round-off that has occurred in the iteration so far. Using this upper bound, the algorithm may decide, at each iteration, to perform a "group update", to restart the algorithm (setting the right hand side appropriately), or both. The benefit from using a group update strategy is analogous to grouping to reduce round-off error in finite precision summation. Following this work, Van der Vorst and Ye devised a residual replacement strategy, which, rather than restarting, replaces the residual with the computed value of the true residual, combined with group updates [24]. This residual replacement occurs at iterations chosen such that two objectives are met: 1) the accumulated round-off does not grow so large as to limit the attainable accuracy, and 2) the Lanczos process is not perturbed so much as to slow the rate of convergence. To determine when these conditions are met, the algorithm iteratively updates a bound on the error accrued thus far. Our analysis closely parallels that of Van der Vorst and Ye.

1.2 Communication-Avoiding Conjugate Gradient

We briefly review the Communication-Avoiding Conjugate Gradient algorithm (CA-CG), given in Algorithm 1. We chose CG for simplicity, although the same general technique can be applied to other KSMs as well. In the interest of space, we will not derive the algorithm here, but instead refer the reader to numerous other works on the topic, such as [3, 5, 6, 11, 13, 19, 22]. The CA-CG method has both an inner loop, which iterates from $j = 1 : s$, and k outer loop iterations, where k depends on the number of steps until convergence (or some other termination condition). Therefore, we will index quantities as $sk + j$ for clarity.

In CA-CG, we do not update x^{sk+j}, r^{sk+j}, and p^{sk+j} directly within the inner loop, but instead update their coefficients in the Krylov basis

$$V^k = [P^k,\, R^k] = [\rho_0(A)p^{sk},\, \rho_1(A)p^{sk}, ...,\, \rho_s(A)p^{sk},\, \rho_0(A)r^{sk},\, \rho_1(A)r^{sk}, ...,\, \rho_{s-1}(A)r^{sk}]$$

where ρ_i is a polynomial of degree i. We assume a three-term recurrence for generating these polynomials, defined by parameters parameters γ_i, θ_i, and σ_i:

$$\rho_i(z) = \gamma_i(A - \theta_i I)\rho_{i-1}(z) + \sigma_i\rho_{i-2}(z)$$

This Krylov basis is generated at the beginning of each outer loop, using the current r and p vectors, by the communication-avoiding matrix powers kernel, denoted $Akx()$ below. We then represent x^{sk+j}, r^{sk+j}, and p^{sk+j} by coefficients e^{sk+j}, c^{sk+j}, and a^{sk+j}, which are vectors of length $O(2s + 1)$, such that

$$\begin{aligned} x^{sk+j} &= V^k e^{sk+j} + x^{sk} \\ r^{sk+j} &= V^k c^{sk+j} \\ p^{sk+j} &= V^k a^{sk+j} \end{aligned}$$

The matrix powers kernel also returns the tridiagonal matrix T, of dimension $(s + 1) \times s$, constructed such that

$$\begin{aligned} AP_i^k &= P_{i+1}^k T_{i+1} \\ AR_i^k &= R_{i+1}^k T_{i+1} \end{aligned}$$

where P_i^k, R_i^k are $n \times i$ matrices containing the first i columns of P^k or R^k, respectively, and T_{i+1} is the matrix containing the first $i + 1$ rows and first i columns of T. Let $V_j^k = [P_j^k,\, R_{j-1}^k]$. This allows us to write

$$AV_j^k = A[P_j^k,\, R_{j-1}^k] = [P_{j+1}^k T_{j+1},\, R_j^k T_j] = V_{j+1}^k \begin{bmatrix} T_{j+1} & \\ & T_j \end{bmatrix}$$

Let $T'_{j+1} = \begin{bmatrix} T_{j+1} & \\ & T_j \end{bmatrix}$. We can now write an expression for Ap^{sk+j} as follows,

$$\begin{aligned} Ap^{sk+j} &= AV^k a^{sk+j} \\ &= (AV_j^k)\bar{a}^{sk+j} \\ &= V_{j+1}^k T'_{j+1} \bar{a}^{sk+j} \\ &= V^k T' a^{sk+j} \end{aligned}$$

where $\bar{a}^{sk+j}$ denotes the nonzero entries of a^{sk+j}, in order to match dimensions with V_j^k. Therefore, $T'a^{sk+j}$ are the Krylov basis coefficients for Ap^{sk+j}. This expression allows us to avoid explicit multiplication by A within the inner loop, and thus allows us to avoid communication.

```
Algorithm 1 CA-CG Method
$x^0$, $r^0 = b - Ax^0$, $p^0 = r^0$
$k = 0$
while (not converged)
    $[P^k, R^k, T] = Akx(A, [p^{sk}, r^{sk}], [s+1, s], [[\rho_0, ..., \rho_s]])$
        //where $\rho_i$ is a polynomial of degree $i$
    Let $V^k = [P^k, R^k]$,$G^k = (V^k)^T V^k$
    //Initialize coefficient vectors, which
    //will be maintained such that $x^{sk+j} = V^k e^{sk+j} + x^{sk}$, $r^{sk+j} = V^k c^{sk+j}$, $p^{sk+j} = V^k a^{sk+j}$
    $a^{sk} = [1, 0_{2s}]^T$,$c^{sk} = [0_{s+1}, 1, 0_s]^T$, $e^{sk} = [0_{2s+1}]$
    for $j = 1 : s$
        $\alpha_{sk+j-1} = ((c^{sk+j-1})^T G^k (c^{sk+j-1}))/((a^{sk+j-1})^T G^k (T' a^{sk+j-1}))$
        $e^{sk+j} = e^{sk+j-1} + \alpha_{sk+j-1} a^{sk+j-1}$
        $c^{sk+j} = c^{sk+j-1} - T' \left(\alpha_{sk+j-1} a^{sk+j-1}\right)$
        $\beta_{sk+j-1} = ((c^{sk+j})^T G^k (c^{sk+j}))/((c^{sk+j-1})^T G^k (c^{sk+j-1}))$
        $a^{sk+j} = c^{sk+j} + \beta_{sk+j-1} a^{sk+j-1}$
    end for
    $x^{sk+s} = [P^k, R^k] e^{sk+s} + x^{sk}$
    $r^{sk+s} = [P^k, R^k] c^{sk+s}$
    $p^{sk+s} = [P^k, R^k] a^{sk+s}$
    $k = k + 1$
end while
return $x^{sk}$
```

2 Error in Finite Precision CA-KSMs

We assume A is a floating point matrix. Throughout this analysis, we use a standard model of floating point arithmetic:

$$\begin{aligned} fl(x+y) &= x + y + \delta \ \text{ with } |\delta| \le \epsilon(|x+y|) \\ fl(Ax) &= Ax + \delta \ \text{ with } |\delta| \le \epsilon m_A |A| \cdot |x| + O(\epsilon^2) \end{aligned}$$

where ϵ is the unit round-off of the machine, $x,\, y \in R^N$, and m_A is a constant associated with the matrix-vector multiplication (for example, the maximal number of nonzero entries in a row of A). All inequalities are componentwise. Using this model, we can also write

$$fl(y + Ax) = y + Ax + \delta \ \text{ with } |\delta| \leq \epsilon(|y + Ax| + m_A|A| \cdot |x|) + O(\epsilon^2)$$

We can now perform an analysis of round-off error in computing the updates in s-step methods.

2.1 Error in Iterate Updates

In the communication-avoiding variant of CG, we represent and symbolically update vectors $x^{sk+j} = V^k e^{sk+j} + x^{sk}$ and $r^{sk+j} = V^k c^{sk+j}$ by their coefficients in the basis $V^k = [P^k,\, R^k]$, where P^k and R^k are the $O(s)$ dimensional Krylov basis vectors with starting vectors p^{sk} and r^{sk}, respectively. These vectors are initialized as

$$e^{sk} = [0_{2s+1}]^T,\ c^{sk} = [0_{s+1}, 1, 0_s]^T$$

In the inner loop, we update these coefficients as

$$\begin{aligned} e^{sk+j} &= e^{sk+j-1} + \alpha_{sk+j-1} a^{sk+j-1} && (2) \\ c^{sk+j} &= c^{sk+j-1} - T'\left(\alpha_{sk+j-1} a^{sk+j-1}\right) && (3) \end{aligned}$$

When Equations (2) and (3) are implemented in finite precision, they become

$$\begin{aligned} \hat{e}^{sk+j} &= fl(\hat{e}^{sk+j-1} + \alpha_{sk+j-1} a^{sk+j-1}) = \hat{e}^{sk+j-1} + \alpha_{sk+j-1} a^{sk+j-1} + \xi^{sk+j} && (4) \\ & \quad |\xi^{sk+j}| \leq \epsilon|\hat{e}^{sk+j}| + O(\epsilon^2) && (5) \\ \hat{c}^{sk+j} &= fl(\hat{c}^{sk+j-1} - T'\left(\alpha_{sk+j-1} a^{sk+j-1}\right)) = \hat{c}^{sk+j-1} - T'\left(\alpha_{sk+j-1} a^{sk+j-1}\right) + \eta^{sk+j} && (6) \\ & \quad |\eta^{sk+j}| \leq \epsilon(|\hat{c}^{sk+j}| + m_T|T'| \cdot |\alpha_{sk+j-1} a^{sk+j-1}|) + O(\epsilon^2) && (7) \end{aligned}$$

Note that the rounding errors in computing $\alpha_{sk+j-1} a^{sk+j-1}$ do not affect the numerical deviation of the true and computed residuals [24]. Rather, the deviation of the two residuals is due to the different round-off patterns that come from different treatment of $\alpha_{sk+j-1} a^{sk+j-1}$ in the recurrences for e^{sk+j} and c^{sk+j}. Therefore, we let the term $\alpha_{sk+j-1} a^{sk+j-1}$ denote the computed quantity.

To avoid confusion, we let $\hat{x}^{sk+j} = \hat{V}^k \hat{e}^{sk+j} + \hat{\hat{x}}^{sk}$ and $\hat{r}^{sk+j} = \hat{V}^k \hat{c}^{sk+j}$ denote the exact values of expressions whose constituents ($\hat{\hat{x}}^{sk}$,$\hat{V}^k$,$\hat{e}^{sk+j}$ and $\hat{c}^{sk+j}$) are computed in finite precision. We use $\hat{\hat{x}}^{sk+j}$, $\hat{\hat{r}}^{sk+j}$ to denote the floating point evaluations of the same expressions. Assuming $x^0 = 0$,

$$\begin{aligned} \hat{\hat{x}}^{sk+j} \quad = fl(\hat{x}^{sk+j}) \quad &= fl(fl(\hat{V}^k \cdot \hat{e}^{sk+j}) + \hat{\hat{x}}^{sk}) = \hat{V}^k \hat{e}^{sk+j} + \psi^{sk+j} + \sum_{i=0}^{k-1} (\hat{V}^i \hat{e}^{si+s} + \psi^{si+s}) && (8) \\ &= \left(\hat{V}^k \hat{e}^{sk+j} + \sum_{i=0}^{k-1} \hat{V}^i \hat{e}^{si+s}\right) + \left(\psi^{sk+j} + \sum_{i=0}^{k-1} \psi^{si+s}\right) \\ & |\psi^{sk+j}| \leq \epsilon(|\hat{\hat{x}}^{sk+j}| + m_V|\hat{V}^k| \cdot |\hat{e}^{sk+j}|) + O(\epsilon^2) && (9) \end{aligned}$$

and

$$\begin{aligned} \hat{\hat{r}}^{sk+j} \quad = fl(\hat{r}^{sk+j}) = \quad & fl(\hat{V}^k \hat{c}^{sk+j}) = \hat{V}^k \hat{c}^{sk+j} + \phi^{sk+j} && (10) \\ & |\phi^{sk+j}| \leq \epsilon(|\hat{\hat{r}}^{sk+j}| + m_V|\hat{V}^k| \cdot |\hat{c}^{sk+j}|) && (11) \end{aligned}$$

We can then write an expression for $\hat{\hat{x}}^{sk+j}$ in terms of $\hat{x}^{sk+j}$:

$$\begin{aligned} \hat{x}^{sk+j} &= \hat{V}^k \hat{e}^{sk+j} + \hat{\hat{x}}^{sk} \\ &= \hat{V}^k \hat{e}^{sk+j} + \sum_{i=0}^{k-1} \left(\hat{V}^i \hat{e}^{si+s} + \psi^{si+s}\right) \\ &= \hat{\hat{x}}^{sk+j} - \psi^{sk+j} \\ \hat{\hat{x}}^{sk+j} &= \hat{x}^{sk+j} + \psi^{sk+j} && (12) \end{aligned}$$

Note that we don't need to explicitly compute $\hat{\hat{x}}^{sk+j}$ or $\hat{\hat{r}}^{sk+j}$ within an inner loop iteration in order to update the representation of the current solution, $\hat{e}^{sk+j}$, and residual, $\hat{c}^{sk+j}$, in the Krylov basis in the next inner loop iteration. Therefore the round-off error in computing $\hat{\hat{x}}^{sk+j}$ and $\hat{\hat{r}}^{sk+j}$ is not cumulative between inner iterations - the only error that accumulates is the error in updating $\hat{e}^{sk+j}$ and $\hat{c}^{sk+j}$.

In the following subsection, we analyze round-off error that occurs in finite precision CA-KSMs. We will obtain an upper bound for the norm of the difference between the true and computed residual at step $sk+j$.

2.2 Deviation of the True and Computed Residual

We can premultiply Equation 4 by $A\hat{V}^k$ to write an expression (in exact arithmetic) for the value of $A\hat{x}^{sk+j}$,

$$A\hat{x}^{sk+j} = A(\hat{V}^k\hat{e}^{sk+j} + \hat{\hat{x}}^{sk}) = A\hat{V}^k\hat{e}^{sk+j-1} + \alpha_{sk+j-1}A\hat{V}^k a^{sk+j-1} + A\hat{\hat{x}}^{sk} + A\hat{V}^k\xi^{sk+j} \tag{13}$$

and we can premultiply Equation (6) by $\hat{V}^k$ to write an expression (in exact arithmetic) for $\hat{r}^{sk+j}$:

$$\hat{r}^{sk+j} = \hat{V}^k\hat{c}^{sk+j} = \hat{V}^k\hat{c}^{sk+j-1} - \alpha_{sk+j-1}\hat{V}^k T' a^{sk+j-1} + \hat{V}^k\eta^{sk+j} \tag{14}$$

We can now write an expression for the difference between the true residual and the computed residual using our recurrences for $A\hat{x}^{sk+j}$ and $\hat{r}^{sk+j}$:

$$\begin{aligned}
b - A\hat{x}^{sk+j} - \hat{r}^{sk+j} &= b - A\hat{V}^k\hat{e}^{sk+j} - A\hat{\hat{x}}^{sk} - \hat{V}^k\hat{c}^{sk+j} \\
&= (b - A\hat{V}^k\hat{e}^{sk+j-1} - A\hat{\hat{x}}^{sk} - \hat{V}^k\hat{c}^{sk+j-1}) - (\alpha_{sk+j-1}A\hat{V}^k a^{sk+j-1} - \alpha_{sk+j-1}\hat{V}^k T' a^{sk+j-1}) \\
&- (A\hat{V}^k\xi^{sk+j} + \hat{V}^k\eta^{sk+j}) \\
&= (b - A\hat{x}^{sk+j-1} - \hat{r}^{sk+j-1}) - (\alpha_{sk+j-1}A\hat{V}^k a^{sk+j-1} - \alpha_{sk+j-1}\hat{V}^k T' a^{sk+j-1}) \\
&- (A\hat{V}^k\xi^{sk+j} + \hat{V}^k\eta^{sk+j}) \\
&= (b - A\hat{\hat{x}}^{sk} - \hat{\hat{r}}^{sk}) - \sum_{i=1}^{j}(\alpha_{sk+i-1}A\hat{V}^k a^{sk+i-1} - \alpha_{sk+i-1}\hat{V}T' a^{sk+i-1} + A\hat{V}\xi^{sk+i} + \hat{V}\eta^{sk+i})
\end{aligned} \tag{15}$$

Then we can bound the 2-norm as:

$$\begin{aligned}
||b - A\hat{x}^{sk+j} - \hat{r}^{sk+j}||_2 &\leq ||b - A\hat{\hat{x}}^{sk} - \hat{\hat{r}}^{sk}||_2 \\
&+ \sum_{i=1}^{j}\Big(||\alpha_{sk+i-1}A\hat{V}^k a^{sk+i-1} - \alpha_{sk+i-1}\hat{V}^k T' a^{sk+i-1}||_2 + ||A\hat{V}^k\xi^{sk+i}||_2 + ||\hat{V}^k\eta^{sk+i}||_2\Big)
\end{aligned} \tag{16}$$

The first term on the right hand side, $||b - A\hat{\hat{x}}^{sk} - \hat{\hat{r}}^{sk}||_2$, gives the norm of the accumulated error at the start of this outer loop iteration. The remaining terms on the right hand side denote the error, or the deviation of the computed from the true residual, accrued in each inner iteration due to finite precision coefficient updates. In order to determine when the true and computed residual have deviated too far, we need to keep track of an estimate of these quantities, and do it in a communication-avoiding way. We will first address the summation term, or, the error in the coefficient updates.

2.2.1 Error in Coefficient Updates within Inner Loop

We will go through and bound each term in the summation: $(1)\alpha_{sk+i-1}A\hat{V}^k a^{sk+i-1} - \alpha_{sk+i-1}\hat{V}^k T' a^{sk+i-1}$, $(2)A\hat{V}^k\xi^{sk+i}$, and $(3)\hat{V}^k\eta^{sk+i}$. Throughout this analysis, we will tend to favor the 2-norm. Although the analysis could be done using any $p-$norm, the 2-norm quantities are easily computable in a communication-avoiding fashion, since the $O(2s+1 \times 2s+1)$ Gram matrix encodes the dot-products with the basis vectors. For the remainder of this paper we will drop terms higher than $O(\epsilon)$ for simplification.

Theorem 2.1. *Let*

$$\hat{V}^k_{i-1} = [\hat{P}^k_{i-1},\, \hat{R}^k_{i-2}] = [\hat{v}_{p,\,sk}, ..., \,\hat{v}_{p,\,sk+i-1},\, \hat{v}_{r,\,sk}, ..., \,\hat{v}_{r,\,sk+i-2}]$$
$$\hat{V}^k_{i} = [\hat{P}^k_{i},\, \hat{R}^k_{i-1}] = [\hat{P}^k_{i-1},\, \hat{v}_{p,\,sk+i},\, \hat{R}^k_{i-2},\, \hat{v}_{r,\,sk+i-1}]$$

be matrices of $2i-1$ *and* $2i+1$ *basis vectors, respectively, for a Krylov subspace with* A*. Then* $||\alpha_{sk+i-1}A\hat{V}^k_{i-1}a^{sk+i-1} - \alpha_{sk+i-1}\hat{V_i}^k T' a^{sk+i-1}||_2$ *is bounded above by*

$$||\alpha_{sk+i-1}A\hat{V}^k_{i-1}a^{sk+i-1} - \alpha_{sk+i-1}\hat{V}^k_i T' a^{sk+i-1}||_2$$

$$\begin{aligned}
&\leq \; ||A\hat{V}^k_{i-1} - \hat{V}^k_i T'||_2 \cdot ||\alpha_{sk+i-1}a^{sk+i-1}||_2 \\
&\leq \; \sqrt{N} \cdot \max\left(||\frac{\zeta^{p,\,sk+1}}{\gamma_1}||_2, ..., \,||\frac{\zeta^{p,\,sk+i}}{\gamma_i}||_2, ||\frac{\zeta^{r,\,sk+1}}{\gamma_1}||_2, ..., \,||\frac{\zeta^{r,\,sk+i-1}}{\gamma_i}||_2 \right) \\
&\quad \cdot \left(||\hat{e}^{sk+i} - \hat{e}^{sk+i-1}||_2\right)
\end{aligned}$$

where

$$\begin{aligned}
||\frac{\zeta^{p,\,sk+t}}{\gamma_t}||_2 &\leq \epsilon \left(\frac{1}{|\gamma_t|} \cdot ||\hat{v}_{p,\,sk+t}||_2 + (2|\theta_t| + 2m_A||A||_2) \cdot ||\hat{v}_{p,\,sk+t-1}||_2 + \frac{2|\sigma_t|}{|\gamma_t|} \cdot ||\hat{v}_{p,\,sk+t-2}||_2 \right) \quad 1 \leq t \leq i \\
||\frac{\zeta^{r,\,sk+t}}{\gamma_t}||_2 &\leq \epsilon \left(\frac{1}{|\gamma_t|} \cdot ||\hat{v}_{r,\,sk+t}||_2 + (2|\theta_t| + 2m_A||A||_2) \cdot ||\hat{v}_{r,\,sk+t-1}||_2 + \frac{2|\sigma_t|}{|\gamma_t|} \cdot ||\hat{v}_{r,\,sk+t-2}||_2 \right) \quad 1 \leq t \leq i
\end{aligned}$$

Proof. We will bound $||\alpha_{sk+i-1}A\hat{V}^k_{i-1}a^{sk+i-1} - \alpha_{sk+i-1}\hat{V}^k_i T' a^{sk+i-1}||_2$, computed exactly. We can rewrite this as

$$||\alpha_{sk+i-1}A\hat{V}^k_{i-1}a^{sk+i-1} - \alpha_{sk+i-1}\hat{V}^k_i T' a^{sk+i-1}||_2 \leq ||A\hat{V}^k_{i-1} - \hat{V}^k_i T'||_2 \cdot ||\alpha_{sk+i-1}a^{sk+i-1}||_2$$

First we will bound $||\alpha_{sk+i-1}a^{sk+i-1}||_2$. Using equation (4), we can write

$$\alpha_{sk+i-1}a^{sk+i-1} = \hat{e}^{sk+i} - \hat{e}^{sk+i-1} - \xi^{sk+i}$$

$$|\alpha_{sk+i-1}a^{sk+i-1}| \;\; \leq \;\; |\hat{e}^{sk+i} - \hat{e}^{sk+i-1}| + |\xi^{sk+i}|$$

$$||\alpha_{sk+i-1}a^{sk+i-1}||_2 \leq ||\hat{e}^{sk+i} - \hat{e}^{sk+i-1}||_2 + \epsilon||\hat{e}^{sk+i}||_2 \tag{17}$$

Now, the term left to bound is $||A\hat{V}^k_{i-1} - \hat{V}^k_i T'||_2$. We know that, computed in finite precision,

$$\hat{V}^k_i = [\hat{P}^k_{i-1},\, \hat{v}_{p,\,sk+i},\, \hat{R}^k_{i-2},\, \hat{v}_{r,\,sk+i-1}]$$

where $v_{p,\,sk+i}$ is a basis vector for the Krylov subspace with starting vector $\hat{p}^{sk}$, defined by the formula (similarly for $v_{r,\,sk+i-1}$):

$$\begin{aligned}
v_{p,\,sk+i} &= \gamma_i(A - \theta_i I)\hat{v}_{p,\,sk+i-1} + \sigma_i\hat{v}_{p,\,sk+i-2} \\
&= \gamma_i A\hat{v}_{p,\,sk+i-1} - \gamma_i\theta_i\hat{v}_{p,\,sk+i-1} + \sigma_i\hat{v}_{p,\,sk+i-2}
\end{aligned} \tag{18}$$

Parameters γ_i, θ_i, and σ_i are coefficients defining the three-term polynomial basis for the Krylov subspace. In the monomial basis, θ_i and σ_i are always 0, and $\gamma_i = 1$. For Newton, $\gamma_i = 1$, $\sigma_i = 0$, and θ_i are chosen to be eigenvalue estimates (Ritz values), ordered according to the (modified) Leja ordering. For Chebyshev, these parameters are chosen based on the bounding ellipse for the estimated eigenvalues of A.

When Equation (18) is implemented in finite precision, we get (See Appendix A), similarly for $v_{r,\,sk+i-1}$:

$$\begin{aligned}
\hat{v}_{p,\,sk+i} = fl(v_{p,\,sk+i}) \quad &= \quad \gamma_i (A - \theta_i I)\hat{v}_{p,\,sk+i-1} + \sigma_i \hat{v}_{p,\,sk+i-2} + \zeta^{sk+i} \\
&= \quad \gamma_i A \hat{v}_{p,\,sk+i-1} - \gamma_i \theta_i \hat{v}_{p,\,sk+i-1} + \sigma_i \hat{v}_{p,\,sk+i-2} + \zeta^{sk+i} && (19) \\
& \quad\ |\zeta^{p,\,sk+i}| \le \epsilon \left(|\hat{v}_{p,\,sk+i}| + 2(|\gamma_i \theta_i| + m_A \cdot |\gamma_i| \cdot |A|) \cdot |\hat{v}_{p,\,sk+i-1}| + 2|\sigma_i \hat{v}_{p,\,sk+i-2}|\right) && (20)
\end{aligned}$$

Now, rearranging Equation (19), we get an expression for $A\hat{v}_{p,\,sk+i-1}$ (or, similarly, $A\hat{v}_{r,\,sk+i-2}$):

$$\begin{aligned}
A\hat{v}_{p,\,sk+i-1} \quad &= \quad \frac{1}{\gamma_i}\left[\hat{v}_{p,\,sk+i} + \gamma_i \theta_i \hat{v}_{p,\,sk+i-1} - \sigma_i \hat{v}_{p,\,sk+i-2} - \zeta^{p,\,sk+i}\right] \\
&= \quad \frac{1}{\gamma_i}\hat{v}_{p,\,sk+i} + \theta_i \hat{v}_{p,\,sk+i-1} - \frac{\sigma_i}{\gamma_i}\hat{v}_{p,\,sk+i-2} - \frac{\zeta^{p,\,sk+i}}{\gamma_i}
\end{aligned}$$

Notice that the right hand side is a multiplication of the finite precision basis vectors by T, our tridiagonal matrix with change-of-basis coefficients, plus the error term, $\frac{\zeta^{p,\,sk+i}}{\gamma_i}$. To write $A\hat{V}^k_{i-1}$, we can write the above as a matrix equation:

$$\begin{aligned}
&A \cdot [\hat{v}_{p,\,sk},\ ...,\ \hat{v}_{p,\,sk+i-1},\ \hat{v}_{r,\,sk},\ ...,\ \hat{v}_{r,\,sk+i-2}] \\
&\qquad = \quad [\hat{v}_{p,\,sk},\ ...,\ \hat{v}_{p,\,sk+i},\ \hat{v}_{r,\,sk},\ ...,\ \hat{v}_{r,\,sk+i-1}] \cdot T' \\
&\qquad\qquad -[\frac{\zeta^{p,\,sk+1}}{\gamma_1},\ ...,\ \frac{\zeta^{p,\,sk+i}}{\gamma_i}, \frac{\zeta^{r,\,sk+1}}{\gamma_1},\ ...,\ \frac{\zeta^{r,\,sk+i-1}}{\gamma_i}] \\
&A\hat{V}^k_{i-1} \quad = \quad \hat{V}^k_i T' - [\frac{\zeta^{p,\,sk+1}}{\gamma_1},\ ...,\ \frac{\zeta^{p,\,sk+i}}{\gamma_i}, \frac{\zeta^{r,\,sk+1}}{\gamma_1},\ ...,\ \frac{\zeta^{r,\,sk+i-1}}{\gamma_i}]
\end{aligned}$$

We can rearrange the above equation to get

$$A\hat{V}^k_{i-1} - \hat{V}^k_i T' = -[\frac{\zeta^{p,\,sk+1}}{\gamma_1},\ ...,\ \frac{\zeta^{p,\,sk+i}}{\gamma_i}, \frac{\zeta^{r,\,sk+1}}{\gamma_1},\ ...,\ \frac{\zeta^{r,\,sk+i-1}}{\gamma_i}]$$

Taking the norm of both sides, we see that

$$\begin{aligned}
||A\hat{V}^k_{i-1} - \hat{V}^k_i T'||_2 &\le ||[\frac{\zeta^{p,\,sk+1}}{\gamma_1},\ ...,\ \frac{\zeta^{p,\,sk+i}}{\gamma_i}, \frac{\zeta^{r,\,sk+1}}{\gamma_1},\ ...,\ \frac{\zeta^{r,\,sk+i-1}}{\gamma_i}]||_2 \\
&\le \sqrt{N}||[\frac{\zeta^{p,\,sk+1}}{\gamma_1},\ ...,\ \frac{\zeta^{p,\,sk+i}}{\gamma_i}, \frac{\zeta^{r,\,sk+1}}{\gamma_1},\ ...,\ \frac{\zeta^{r,\,sk+i-1}}{\gamma_i}]||_1 \\
&= \sqrt{N} \cdot \max\left(||\frac{\zeta^{p,\,sk+1}}{\gamma_1}||_1, ..., ||\frac{\zeta^{p,\,sk+i}}{\gamma_i}||_1, ||\frac{\zeta^{r,\,sk+1}}{\gamma_1}||_1, ..., ||\frac{\zeta^{r,\,sk+i-1}}{\gamma_i}||_1\right) \\
&\le \sqrt{N} \cdot \max\left(||\frac{\zeta^{p,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{p,\,sk+i}}{\gamma_i}||_2, ||\frac{\zeta^{r,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{r,\,sk+i-1}}{\gamma_i}||_2\right)
\end{aligned}$$

Now, we can write the whole bound as

$$\begin{aligned}
&||\alpha_{sk+i-1} A\hat{V}^k_{i-1} a^{sk+i-1} - \alpha_{sk+i-1}\hat{V}^k_i T' a^{sk+i-1}||_2 \\
&\le ||A\hat{V}^k_{i-1} - \hat{V}^k_i T'||_2 \cdot ||\alpha_{sk+i-1} a^{sk+i-1}||_2 \\
&\le \sqrt{N} \cdot \max\left(||\frac{\zeta^{p,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{p,\,sk+i}}{\gamma_i}||_2, ||\frac{\zeta^{r,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{r,\,sk+i-1}}{\gamma_i}||_2\right) \\
&\cdot \left(||\hat{e}^{sk+i} - \hat{e}^{sk+i-1}||_2\right) + O(\epsilon^2)
\end{aligned}$$

Using Equation (20), we get

$$||\frac{\zeta^{p,\,sk+t}}{\gamma_t}||_2 \leq \epsilon \left(\frac{1}{|\gamma_t|} \cdot ||\hat{v}_{p,\,sk+t}||_2 + (2|\theta_t| + 2m_A||A||_2) \cdot ||\hat{v}_{p,\,sk+t-1}||_2 + \frac{2|\sigma_t|}{|\gamma_t|} \cdot ||\hat{v}_{p,\,sk+t-2}||_2 \right)$$
$$1 \leq t \leq i$$
$$||\frac{\zeta^{r,\,sk+t}}{\gamma_t}||_2 \leq \epsilon \left(\frac{1}{|\gamma_t|} \cdot ||\hat{v}_{r,\,sk+t}||_2 + (2|\theta_t| + 2m_A||A||_2) \cdot ||\hat{v}_{r,\,sk+t-1}||_2 + \frac{2|\sigma_t|}{|\gamma_t|} \cdot ||\hat{v}_{r,\,sk+t-2}||_2 \right)$$
$$1 \leq t \leq i-1$$

This proves the Theorem. □

We have two terms left to bound in Equation (16), $A\hat{V}^k\xi^{sk+i}$ and $\hat{V}^k\eta^{sk+i}$. We can bound the 2-norm of $A\hat{V}^k\xi^{sk+i}$ as

$$\begin{aligned} ||A\hat{V}^k\xi^{sk+i}||_2 &\leq ||A||_2 \cdot ||\,|\hat{V}^k| \cdot |\xi^{sk+i}|\,||_2 \\ &\leq \epsilon||A||_2 \cdot ||\,|\hat{V}^k| \cdot |\hat{e}^{sk+i}|\,||_2 \end{aligned} \tag{21}$$

Now, to bound the 2-norm of $\hat{V}^k\eta^{sk+i}$, we plug in and use Equation (7):

$$\begin{aligned} ||\hat{V}^k\eta^{sk+i}||_2 &\leq ||\,|\hat{V}^k| \cdot |\eta^{sk+i}|\,||_2 \\ &\leq \epsilon||\,|\hat{V}^k| \cdot (|\hat{c}^{sk+i}| + m_T|T'| \cdot |\alpha_{sk+i-1}a^{sk+i-1}|)||_2 \\ &\leq \epsilon||\,|\hat{V}^k| \cdot (|\hat{c}^{sk+i}| + m_T|T'| \cdot |\hat{e}^{sk+i} - \hat{e}^{sk+i-1}|)||_2 \\ &\leq \epsilon(||\,|\hat{V}^k| \cdot |\hat{c}^{sk+i}|\,||_2 + m_T||T'||_2 \cdot ||\,|\hat{V}^k| \cdot |\hat{e}^{sk+i} - \hat{e}^{sk+i-1}|\,||_2) \end{aligned} \tag{22}$$

Putting all our terms together, we find

$$\begin{aligned} &||b - A\hat{x}^{sk+j} - \hat{r}^{sk+j}||_2 \\ &\leq ||b - A\hat{\hat{x}}^{sk} - \hat{\hat{r}}^{sk}||_2 \\ &+ \sum_{i=1}^{j} \Big[||\alpha_{sk+i-1}A\hat{V}^k a^{sk+i-1} - \alpha_{sk+i-1}\hat{V}^k T' a^{sk+i-1}||_2 + ||A\hat{V}^k\xi^{sk+i}||_2 + ||\hat{V}^k\eta^{sk+i}||_2\Big] \\ &\leq ||b - A\hat{\hat{x}}^{sk} - \hat{\hat{r}}^{sk}||_2 \\ &+ \sum_{i=1}^{j}[\sqrt{N} \cdot \max\left(||\frac{\zeta^{p,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{p,\,sk+i}}{\gamma_i}||_2, ||\frac{\zeta^{r,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{r,\,sk+i-1}}{\gamma_i}||_2\right) \cdot \big(||\hat{e}^{sk+i} - \hat{e}^{sk+i-1}||_2\big) \\ &+ \epsilon||A||_2 \cdot ||\,|\hat{V}^k| \cdot |\hat{e}^{sk+i}|\,||_2 + \epsilon(||\,|\hat{V}^k| \cdot |\hat{c}^{sk+i}|\,||_2 + m_T||T'||_2 \cdot ||\,|\hat{V}^k| \cdot |\hat{e}^{sk+i} - \hat{e}^{sk+i-1}|\,||_2)] \end{aligned} \tag{23}$$

2.2.2 Error in Basis Change in Outer Loop

Now, we want to bound the term $||b - A\hat{\hat{x}}^{sk} - \hat{\hat{r}}^{sk}||_2$ in Equation (23). We can write, again assuming $x^0 = 0$, $r^0 = b$,

$$
\begin{aligned}
&||b - A\hat{\hat{x}}^{sk} - \hat{\hat{r}}^{sk}||_2 \\
&= ||b - A(\hat{x}^{sk} + \psi^{sk}) - \hat{r}^{sk} - \phi^{sk}||_2 \\
&\leq ||b - A\hat{x}^{sk} - \hat{r}^{sk}||_2 + ||A\psi^{sk} + \phi^{sk}||_2 \\
&= ||b - A\hat{x}^{s(k-1)+s} - \hat{r}^{s(k-1)+s}||_2 + ||A\psi^{s(k-1)+s} + \phi^{s(k-1)+s}||_2 \\
&\leq ||b - A\hat{\hat{x}}^{s(k-1)} - \hat{\hat{r}}^{s(k-1)}||_2 + ||A\psi^{s(k-1)+s} + \phi^{s(k-1)+s}||_2 \\
&+ \sum_{i=1}^{s}[||\alpha_{s(k-1)+i-1}A\hat{V}^{k-1}a^{s(k-1)+i-1} - \alpha_{s(k-1)+i-1}\hat{V}^{k-1}T'a^{s(k-1)+i-1}||_2 \\
&+ ||A\hat{V}^{k-1}\xi^{s(k-1)+i}||_2 + ||\hat{V}^{k-1}\eta^{s(k-1)+i}||_2] \\
&\leq \sum_{l=0}^{k-1}\left(||A\psi^{sl+s} + \phi^{sl+s}||_2 + \sum_{i=1}^{s}\left[||\alpha_{sl+i-1}A\hat{V}^{l}a^{sl+i-1} - \alpha_{sl+i-1}\hat{V}^{l}T'a^{sl+i-1}||_2 + ||A\hat{V}^{l}\xi^{sl+i}||_2 + ||\hat{V}^{l}\eta^{sl+i}||_2\right]\right) \\
&\leq \sum_{l=0}^{k-1}\left(||A\psi^{sl+s}||_2 + ||\phi^{sl+s}||_2\right) \\
&+ \sum_{l=0}^{k-1}\sum_{i=1}^{s}\left(||\alpha_{sl+i-1}A\hat{V}^{l}a^{sl+i-1} - \alpha_{sl+i-1}\hat{V}^{l}T'a^{sl+i-1}||_2 + ||A\hat{V}^{l}\xi^{sl+i}||_2 + ||\hat{V}^{l}\eta^{sl+i}||_2\right) \\
&\leq \sum_{l=0}^{k-1}\left(||A||_2 \cdot ||\psi^{sl+s}||_2 + ||\phi^{sl+s}||_2\right) \\
&+ \sum_{l=0}^{k-1}\sum_{i=1}^{s}\left(||\alpha_{sl+i-1}A\hat{V}^{l}a^{sl+i-1} - \alpha_{sl+i-1}\hat{V}^{l}T'a^{sl+i-1}||_2 + ||A\hat{V}^{l}\xi^{sl+i}||_2 + ||\hat{V}^{l}\eta^{sl+i}||_2\right)
\end{aligned}
\tag{24}
$$

This bound, in words, says that the error at the start of the k^{th} outer loop iteration is the sum of (1) the errors in performing coefficient updates in every inner loop iteration executed so far (iterations 1 through sk) and (2) the errors in computing $\hat{\hat{x}}$ and $\hat{\hat{r}}$ in every outer loop iteration so far (outer loop iterations 1 through k).

2.2.3 Total Error

Putting the terms in the above two sections together, we obtain an upper bound for the error accumulated at iteration $sk + j$:

$$
\begin{align}
&||b - A\hat{x}^{sk+j} - \hat{r}^{sk+j}||_2 \notag\\
&\leq ||(b - A\hat{\hat{x}}^{sk} - \hat{\hat{r}}^{sk})||_2 \notag\\
&+ \sum_{i=1}^{j} \Big(||\alpha_{sk+i-1} A\hat{V}^k a^{sk+i-1} - \alpha_{sk+i-1}\hat{V}^k T' a^{sk+i-1}||_2 + ||A\hat{V}^k \xi^{sk+i}||_2 + ||\hat{V}^k \eta^{sk+i}||_2 \Big) \notag\\
&\leq \sum_{l=0}^{k-1} \big(||A\psi^{sl+s}||_2 + ||\phi^{sl+s}||_2 \big) \notag\\
&+ \sum_{l=0}^{k-1}\sum_{i=1}^{s} \Big(||\alpha_{sl+i-1} A\hat{V}^l a^{sl+i-1} - \alpha_{sl+i-1}\hat{V}^l T' a^{sl+i-1}||_2 + ||A\hat{V}^l \xi^{sl+i}||_2 + ||\hat{V}^l \eta^{sl+i}||_2 \Big) \notag\\
&+ \sum_{i=1}^{j} \Big(||\alpha_{sk+i-1} A\hat{V}^k a^{sk+i-1} - \alpha_{sk+i-1}\hat{V}^k T' a^{sk+i-1}||_2 + ||A\hat{V}^k \xi^{sk+i}||_2 + ||\hat{V}^k \eta^{sk+i}||_2 \Big) \notag\\
&= \sum_{l=0}^{k-1} \big(||A||_2 \cdot ||\psi^{sl+s}||_2 + ||\phi^{sl+s}||_2 \big) \tag{25}\\
&+ \sum_{l=0}^{k-1}\sum_{i=1}^{s} [\sqrt{N} \cdot \max\Big(||\frac{\zeta^{p,\, sl+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{p,\, sl+i}}{\gamma_i}||_2, ||\frac{\zeta^{r,\, sl+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{r,\, sl+i-1}}{\gamma_i}||_2 \Big) \cdot \big(||\hat{e}^{sl+i} - \hat{e}^{sl+i-1}||_2 \big) \notag\\
&\qquad + \epsilon||A||_2 \cdot ||\,|\hat{V}^l| \cdot |\hat{e}^{sl+i}|\,||_2 + \epsilon(||\,|\hat{V}^l| \cdot |\hat{c}^{sl+i}|\,||_2 + m_T||T'||_2 \cdot ||\,|\hat{V}^l| \cdot |\hat{e}^{sl+i} - \hat{e}^{sl+i-1}|\,||_2)] \notag\\
&+ \sum_{i=1}^{j} [\sqrt{N} \cdot \max\Big(||\frac{\zeta^{p,\, sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{p,\, sk+i}}{\gamma_i}||_2, ||\frac{\zeta^{r,\, sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{r,\, sk+i-1}}{\gamma_i}||_2 \Big) \cdot \big(||\hat{e}^{sk+i} - \hat{e}^{sk+i-1}||_2 \big) \notag\\
&\qquad + \epsilon||A||_2 \cdot ||\,|\hat{V}^k| \cdot |\hat{e}^{sk+i}|\,||_2 + \epsilon(||\,|\hat{V}^k| \cdot |\hat{c}^{sk+i}|\,||_2 + m_T||T'||_2 \cdot ||\,|\hat{V}^k| \cdot |\hat{e}^{sk+i} - \hat{e}^{sk+i-1}|\,||_2)] \notag\\
&\leq d_{sk+j} \tag{26}
\end{align}
$$

where we will use d_{sk+j} to denote an upper bound for $||b - A\hat{x}^{sk+j} - \hat{r}^{sk+j}||_2$. By the equation above, we can keep track of this quantity iteratively, by updating this quantity in each iteration as follows:
If $1 \leq j \leq s-1$:

$$
\begin{align}
d_{sk+j} = d_{sk+j-1} &+ \sqrt{N} \cdot \max\Big(||\frac{\zeta^{p,\, sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{p,\, sk+j}}{\gamma_j}||_2, ||\frac{\zeta^{r,\, sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{r,\, sk+j-1}}{\gamma_j}||_2 \Big) \cdot \big(||\hat{e}^{sk+j} - \hat{e}^{sk+j-1}||_2 \big) \notag\\
&+ \epsilon||A||_2 \cdot ||\,|\hat{V}^k| \cdot |\hat{e}^{sk+j}|\,||_2 + \epsilon(||\,|\hat{V}^k| \cdot |\hat{c}^{sk+j}|\,||_2 + m_T||T'||_2 \cdot ||\,|\hat{V}| \cdot |\hat{e}^{sk+j} - \hat{e}^{sk+j-1}|\,||_2) \tag{27}
\end{align}
$$

If $j = s$:

$$
\begin{aligned}
&d_{s(k+1)}\\
&= d_{sk+s} = d_{sk+s-1}\\
&+ \sqrt{N}\cdot\max\left(||\frac{\zeta^{p,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{p,\,sk+s}}{\gamma_s}||_2, ||\frac{\zeta^{r,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{r,\,sk+s-1}}{\gamma_s}||_2\right)\cdot\big(||\hat{e}^{sk+s}-\hat{e}^{sk+s-1}||_2\big)\\
&+ \epsilon||A||_2\cdot||\,|\hat{V}^k|\cdot|\hat{e}^{sk+s}|\,||_2 + \epsilon(||\,|\hat{V}^k|\cdot|\hat{c}^{sk+s}|\,||_2 + m_T||T'||_2\cdot||\,|\hat{V}^k|\cdot|\hat{e}^{sk+s}-\hat{e}^{sk+s-1}|\,||_2)\\
&+ \big(||A||_2\cdot||\psi^{sk+s}||_2 + ||\phi^{sk+s}||_2\big)\\
&= d_{sk+s-1}\\
&+ \sqrt{N}\cdot\max\left(||\frac{\zeta^{p,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{p,\,sk+s}}{\gamma_s}||_2, ||\frac{\zeta^{r,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{r,\,sk+s-1}}{\gamma_s}||_2\right)\cdot\big(||\hat{e}^{sk+s}-\hat{e}^{sk+s-1}||_2\big)\\
&+ \epsilon||A||_2\cdot((1+m_V)\cdot||\,|\hat{V}^k|\cdot|\hat{e}^{sk+s}|\,||_2 + ||\hat{\hat{x}}^{sk+s}||_2) \qquad (28)\\
&+ \epsilon((1+m_V)\cdot||\,|\hat{V}^k|\cdot|\hat{c}^{sk+s}|\,||_2 + ||\hat{\hat{r}}^{sk+s}||_2 + m_T||T'||_2\cdot||\,|\hat{V}^k|\cdot|\hat{e}^{sk+s}-\hat{e}^{sk+s-1}|\,||_2)
\end{aligned}
$$

2.3 Avoiding Communication in Computing the Upper Bound

In each iteration, we will update d^{sk+j}, the deviation of the true and computed residual, given by Equations (27) and (28). Section 3 will discuss how this quantity is used to determine whether or not residual replacement occurs at a given iteration. First, however, we show how to compute the value of d^{sk+j} in a communication-avoiding way, to avoid reintroducing the communication bottlenecks that we sought to remove.

We can assume that we have estimates for $||A||_2$ and $||T'||_2$.These quantities need be computed only once, since their values do not change throughout the iteration. The remaining quantities we must compute are

1. $\max\left(||\frac{\zeta^{p,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{p,\,sk+j}}{\gamma_j}||_2, ||\frac{\zeta^{r,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{r,\,sk+j-1}}{\gamma_j}||_2\right)$,
2. $||\,|\hat{V}^k|\cdot|\hat{e}^{sk+j}|\,||_2$, $||\,|\hat{V}^k|\cdot|\hat{c}^{sk+j}|\,||_2$ and $||\,|\hat{V}|\cdot|\hat{e}^{sk+j}-\hat{e}^{sk+j-1}|\,||_2$,
3. $||\hat{\hat{x}}^{sk+j}||_2$ and $||\hat{\hat{r}}^{sk+j}||_2$, and
4. $||\hat{e}^{sk+j}-\hat{e}^{sk+j-1}||_2$

We can compute $\left\{||\frac{\zeta^{p,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{p,\,sk+s}}{\gamma_j}||_2, ||\frac{\zeta^{r,\,sk+1}}{\gamma_1}||_2, ..., ||\frac{\zeta^{r,\,sk+s-1}}{\gamma_j}||_2\right\}$ at the beginning of the outer loop, and then easily choose the maximum amongst the appropriate $2j-1$ scalar quantities within each inner loop iteration. Computing this set involves computing the norm of each basis vector, which can easily be accomplished by use of the Gram matrix for no additional communication cost. The additional computation cost is $O(s^2)$ per s steps.

For terms in 2. above, we have 2 choices. The first option is to upper bound each quantity as $||\,|\hat{V}^k|\cdot|\hat{e}^{sk+j}|\,||_2 \leq ||\hat{V}^k||_2\cdot||\hat{e}^{sk+j}||_2$. This requires no additional communication, as each processor can compute $||\hat{V}^k||_2$ using the Gram matrix, and computing $||\hat{e}^{sk+j}||_2$, $||\hat{c}^{sk+j}||_2$, and $||\hat{e}^{sk+j}-\hat{e}^{sk+j-1}||_2$ are all local work. The additional computation is $O(s^2)$ per outer loop iteration. The second option is to compute $G_{|\hat{V}|} = |\hat{V}^T|\cdot|\hat{V}|$ in each outer loop. Then $G_{|\hat{V}|}$ can be used to compute the two-norm of these quantities without communication within the inner loop. Although this provides a tighter bound, this option requires an additional global reduction in each outer loop iteration (the asymptotic communication costs remain the same). This option also requires $O(s^2n)$ additional computation per outer loop iteration.

We also use the Gram matrix to compute $||\hat{\hat{x}}^{sk+j}||_2 \leq \sqrt{(\hat{e}^{sk+j})^TG(\hat{e}^{sk+j})}+||\hat{\hat{x}}^{sk}||_2$ and $||\hat{\hat{r}}^{sk+j}||_2 \leq \sqrt{(\hat{c}^{sk+j})^TG(\hat{c}^{sk+j})}$, where $||\hat{\hat{x}}^{sk}||_2$ and $||\hat{\hat{r}}^{sk}||_2$can be computed at the start of the inner loop (since we must communicate to compute and distribute the values of $\hat{\hat{x}}^{sk}$ and $\hat{\hat{r}}^{sk}$ anyway). This requires $O(n+s^2)$ operations per outer loop.

Finally, $||\hat{e}^{sk+j}-\hat{e}^{sk+j-1}||_2$ can be computed locally by each processor, so no additional communication is required. Computing this quantity in each inner loop requires an additional $O(s^2)$ operations per outer loop.

We note that using the Gram matrix to compute inner products locally does result in an extra $O(\epsilon)$ error term. However, since all quantities in 1-3. above are multiplied by ϵ in the bound for d^{sk+j}, these $O(\epsilon)$ error terms become $O(\epsilon^2)$ error terms, and can thus be ignored in our bound.

The communication and computation costs given for computing d^{sk+j} above do not affect the overall computation or communication cost of the algorithm. Therefore, we can keep track of the quantity d^{sk+j} in each iteration without asymptotically increasing the cost of communication or computation in CA-KSMs.

3 Replacement of Residuals in CA-KSMs

In order to improve the maximum attainable accuracy, we want to replace the computed residual with the true residual at certain iterations, according to our calculated d_{sk+j} value. We must choose these iterations carefully to satisfy two constraints: 1) we don't want to let the deviation grow too large, and 2) we don't want to lose super-linear convergence provided by the underlying Lanczos process.

Van der Vorst and Ye [24] have suggested the following condition for residual replacement to satisfy these properties:

$$d_{sk+j-1} \leq \hat{\epsilon}||\hat{r}^{sk+j-1}||_2 \ \&\& \ d_{sk+j} > \hat{\epsilon}||\hat{r}^{sk+j}||_2 \ \&\& \ d_{sk+j} > 1.1 d_{init} \tag{29}$$

where we initially set $d_0 = d_{init} = \epsilon(||r^0||_2 + m_A||A||_2 \cdot ||x^0||_2) = \epsilon||b||_2$, and $\hat{\epsilon}$ is a tolerance parameter. The value $\hat{\epsilon} = \sqrt{\epsilon}$ has been found to be a good value empirically for standard KSMs [24], and we have observed good results for CA-KSMs as well.

In our case, we do not know the actual value of $||\hat{r}^{sk+j-1}||_2$, but we can use the Gram matrix $\hat{G}^k$ to compute $||\hat{\hat{r}}^{sk+j-1}||_2$. We will now argue that we can replace $||\hat{r}^{sk+j-1}||_2$ and $||\hat{r}^{sk+j}||_2$ with $||\hat{\hat{r}}^{sk+j-1}||_2$ and $||\hat{\hat{r}}^{sk+j}||_2$ in Equation (29). Since

$$\hat{r}^{sk+j-1} = \hat{\hat{r}}^{sk+j-1} - \phi^{sk+j-1}$$

we can say

$$\hat{\epsilon}||\hat{r}^{sk+j-1}||_2 \quad \leq \quad \hat{\epsilon}||\hat{\hat{r}}^{sk+j-1}||_2 + \hat{\epsilon}||\phi^{sk+j-1}||_2$$

We know $||\phi^{sk+j-1}||_2 = O(\epsilon)$ and $\hat{\epsilon} = \epsilon^{1/2}$. Since ignore powers of ϵ larger than $O(\epsilon)$, this becomes:

$$\begin{aligned}\hat{\epsilon}||\hat{r}^{sk+j-1}||_2 &\leq \hat{\epsilon}||\hat{\hat{r}}^{sk+j-1}||_2 + O(\epsilon^{3/2}) \\ &\leq \hat{\epsilon}||\hat{\hat{r}}^{sk+j-1}||_2\end{aligned}$$

Therefore, we can use $||\hat{\hat{r}}^{sk+j-1}||_2$ in our replacement condition for CA-KSMs. An analogous argument holds for $||\hat{\hat{r}}^{sk+j}||_2$. Our condition for residual replacement in CA-KSMs will then be

$$d_{sk+j-1} \leq \hat{\epsilon}||\hat{\hat{r}}^{sk+j-1}||_2 \ \&\& \ d_{sk+j} > \hat{\epsilon}||\hat{\hat{r}}^{sk+j}||_2 \ \&\& \ d_{sk+j} > 1.1 d_{init} \tag{30}$$

If this statement is true, we perform a group update by accumulating the current value of $\hat{\hat{x}}^{sk+j}$ into vector $\hat{z}$, as $\hat{z} = fl(\hat{z} + \hat{\hat{x}}^{sk+j})$, and we set $\hat{r}^{sk+j} = fl(b - A\hat{z})$.

To perform a residual replacement in CA-KSMs, all processors must communicate their elements of $\hat{\hat{x}}^{sk+j}$ to compute $b - A\hat{\hat{x}}^{sk+j}$, and we must break out of the inner loop (potentially before completing s steps) and continue with computing the next matrix powers kernel with the new residual in the next outer loop. This means our communication costs could potentially increase if the number of replacements is high (i.e., we compute the true residual every iteration), but our experimental results in the next section indicate that, as long as the generated basis is numerically full-rank and the basis norm growth rate is not too high, the number of replacements is low compared to the total number of iterations. Therefore the communication cost per $s-$steps does not asymptotically increase. A formal round-off analysis of the residual replacement scheme using this condition for KSMs can be found in [24]. Our future work will include a round-off analysis of this replacement scheme for CA-KSMs. The algorithm for residual replacement can be found below in Algorithm 2.

Algorithm 2 Residual Replacement

```
if $d_{sk+j-1} \leq \hat{\epsilon}||\hat{\hat{r}}^{sk+j-1}||_2$ && $d_{sk+j} > \hat{\epsilon}||\hat{\hat{r}}^{sk+j}||_2$ && $d_{sk+j} > 1.1d_{init}$
    $z = z + \hat{\hat{x}}^{sk+j}$
    $r^{sk+j} = b - Az$
    $x^{sk+j} = 0$
    $d_{init} = d_k = \epsilon(||r^{sk+j}||_2 + m_A||A||_2 \cdot ||z||_2)$
    reset$= 1$
    BREAK
end if
```

We can now give the algorithm for CA-CG with residual replacement, shown in Algorithm 3. Blue text denotes new code added to Algorithm 1 for the purpose of residual replacement.

```
Algorithm 3 CA-CG with Residual Replacement
Compute $||A||_2$
$x^0 = 0$, $r^0 = b$, $p^0 = r^0$
$k = 0$
$d_0 = d_{init} = \epsilon(||r^0||_2)$
$z = 0$
reset$= 0$
while (not converged)
    $[P^k, R^k, T] = Akx(A, [p^{sk}, r^{sk}], [s+1, s], [[\rho_0, ..., \rho_s]])$
         //where $\rho_i$ is a polynomial of degree $i$
    if(k==0) Compute $||T'||_2$
    Let $V^k = [P^k, R^k]$,$G^k = (V^k)^T V^k$
    //Initialize coefficient vectors, which
    //will be maintained such that $x^{sk+j} = V^k e^{sk+j} + x^{sk}$, $r^{sk+j} = V^k c^{sk+j}$, $p^{sk+j} = V^k a^{sk+j}$
    $a^{sk} = [1, 0_{2s}]^T$,$c^{sk} = [0_{s+1}, 1, 0_s]^T$, $e^{sk} = [0_{2s+1}]$
    for $j = 1 : s$
        $\alpha_{sk+j-1} = ((c^{sk+j-1})^T G^k (c^{sk+j-1}))/((a^{sk+j-1})^T G^k (T' a^{sk+j-1}))$
        $e^{sk+j} = e^{sk+j-1} + \alpha_{sk+j-1} a^{sk+j-1}$
        $c^{sk+j} = c^{sk+j-1} - \alpha_{sk+j-1} T' a^{sk+j-1}$
        $\beta_{sk+j-1} = ((c^{sk+j})^T G^k (c^{sk+j}))/((c^{sk+j-1})^T G^k (c^{sk+j-1}))$
        $a^{sk+j} = c^{sk+j} + \beta_{sk+j-1} a^{sk+j-1}$
        Update $d_{sk+j}$ using Eq. $(27)$
        if $d_{sk+j-1} \leq \hat{\epsilon}||r^{sk+j-1}||_2$ && $d_{sk+j} > \hat{\epsilon}||r^{sk+j}||_2$ && $d_{sk+j} > 1.1 d_{init}$
            $z = z + \hat{\hat{x}}^{sk+j}$
            $r^{sk+j} = b - Az$
            $x^{sk+j} = 0$
            $d_{init} = d_k = \epsilon(||r^{sk+j}||_2 + m_A ||A||_2 \cdot ||z||_2)$
            reset$= 1$
            BREAK
          end if
    end for
    if reset$! = 1$
        Update $d_{sk+s}$ by Eq. $(28)$
        $x^{sk+s} = [P^k, R^k] e^{sk+s} + x^{sk}$
        $r^{sk+s} = [P^k, R^k] c^{sk+s}$
    end if
    reset=0
    $p^{sk+s} = [P^k, R^k] a^{sk+s}$
    $k = k + 1$
end while
return $z + x^{sk}$
```

4 Experimental Results

We evaluated our residual replacement strategy on a few small matrices (both symmetric and unsymmetric) from the University of Florida Sparse Matrix Collection, using the CA-BICG method (or CA-CG where appropriate). In these experiments, we compare standard (BI)CG with both our CA-(BI)CG method and our CA-(BI)CG method with residual replacement. We ran these tests for $s = [4, 8, 16]$. To lower the 2-norm of the matrix, we used row and column scaling of the input matrix A as a preprocessing equilibration routine, as described in [11]. This process, which only need be performed once, is used in lieu of scaling the basis vectors after they are generated, which reintroduces communication dependencies between iterations. For each matrix, we selected a right hand side b such that $||x||_2 = 1$, $x_i = 1/\sqrt{n}$. We have found empirically that using a replacement tolerance around $\hat{\epsilon} = \sqrt{\epsilon}$, the same value used in Van der Vorst and Ye [24], ensures that the true and computed residual remain the same throughout the computation.

The figures below are organized as follows. The left column shows the convergence of

- standard (BI)CG (black line)
- standard (BI)CG with residual replacement (black dots) [24]
- CA-(BI)CG for all three bases:
  - Monomial (blue line), Newton (green line), Chebyshev (red line)
- CA-(BI)CG with residual replacement for all 3 bases:
  - Monomial (blue dots), Newton (green dots), Chebyshev (red dots).

The right column shows our upper bound estimates, d_{sk+j}, for

- standard (BI)CG with residual replacement (black dashed line) [24]
- CA-(BI)CG with residual replacement for all 3 bases:
  - Monomial (blue dashed line), Newton (green dashed line), and Chebyshev (red dashed line)

vs. the true value of $||r_{true}^{sk+j} - \hat{\hat{r}}^{sk+j}||_2$ for

- standard (BI)CG with residual replacement (black dots) [24]
- CA-(BI)CG with residual replacement for all 3 bases:
  - Monomial (blue dots), Newton (green dots), and Chebyshev (red dots)

Table 1 shows the total number of replacements, and the iterations at which the replacements occurred, for each experiment.

	Naive Method			$s = 4$		$s = 8$		$s = 16$	
Matrix	Total Repl.	Replacement Iterations	Basis	Total Repl.	Replacement Iterations	Total Repl.	Replacement Iterations	Total Repl.	Replacement Iterations
cdde1, cond(A) = 2.4E+03, $\|\|A\|\|_2 = 2.0$	1	88	M	4	76, 88, 107, 121	16	15, 29, 37, 46, 58, 66, 74, 81, ..., 134, 146	34*	6, 10, 13, 16, 20, 25, 27, 30, 33, ..., 115, 120
			N	4	76, 89, 111, 121	4	59, 81, 107, 115	10	13, 26, 41, 55, 65, 76, 87, 102, 111, 121
			C	4	76, 90, 107, 121	4	59, 81, 107, 115	10	13, 26, 41, 55, 65, 76, 87, 102, 111, 121
jpwh991 cond(A) = 93.4, $\|\|A\|\|_2 = 1.9$	1	32	M	2	21, 36	7	8, 16, 23, 32, 41, 49, 56	8*	6, 10, 15, 21, 25, 32, 38, 43
			N	2	23, 41	2	21, 36	4	12, 23, 34, 43
			C	2	23, 41	2	21, 36	3	14, 23, 34
mesh1em1 cond(A) = 11.6, $\|\|A\|\|_2 = 1.8$	1	18	M	1	12	3	7, 15, 22	5*	4, 9, 14, 19, 22
			N	1	15	1	13	2	11, 22
			C	1	15	1	13	2	12, 22
mhdb416 cond(A) = 69.7, $\|\|A\|\|_2 = 2.3$	1	22	M	1	20	4	8, 16, 24, 32	10*	2, 4, 7, 10, 12, 14, 18, 23, 28, 32
			N	1	22	2	20, 30	3	14, 24, 32
			C	1	21	2	19, 29	3	13, 23, 31
nos4 cond(A) = 995.1, $\|\|A\|\|_2 = 2.0$	1	57	M	1	55	6	8, 24, 39, 51, 61, 70	14	4, 9, 14, 20, 25, 30, 35, 40, 46, .., 68, 72
			N	1	56	2	53, 71	3	34, 49, 63
			C	1	58	2	53, 71	4	18, 35, 49, 63
pde900 cond(A) = 119.1, $\|\|A\|\|_2 = 2.0$	1	86	M	4	47, 71, 94, 103	10	28, 35, 49, 58, 71, 79, 88, 96, 110, 121	30	5, 7, 9, 14, 19, 21, 26, 28, 32, 34, ..., 112, 113
			N	5	9, 52, 86, 103, 113	5	48, 71, 94, 103, 115	6	44, 60, 79, 93, 104, 116
			C	3	49, 81, 107	4	48, 71, 94, 103	6	44, 60, 79, 94, 103, 115

Table 1: Residual Replacement Iterations for CA-BICG Experiments. Matrices from the University of Florida Sparse Matrix Collection. Equilibration routine used on A to reduce norm. Right hand side chosen such that $||x|| = 1$. Tolerance used was $\hat{\epsilon} = \sqrt{\epsilon}$. CA-KSM restarts given for monomial basis (M), Newton basis (N), and Chebyshev basis (C). A “*” indicates that a Krylov basis generated by the algorithm was not full rank.

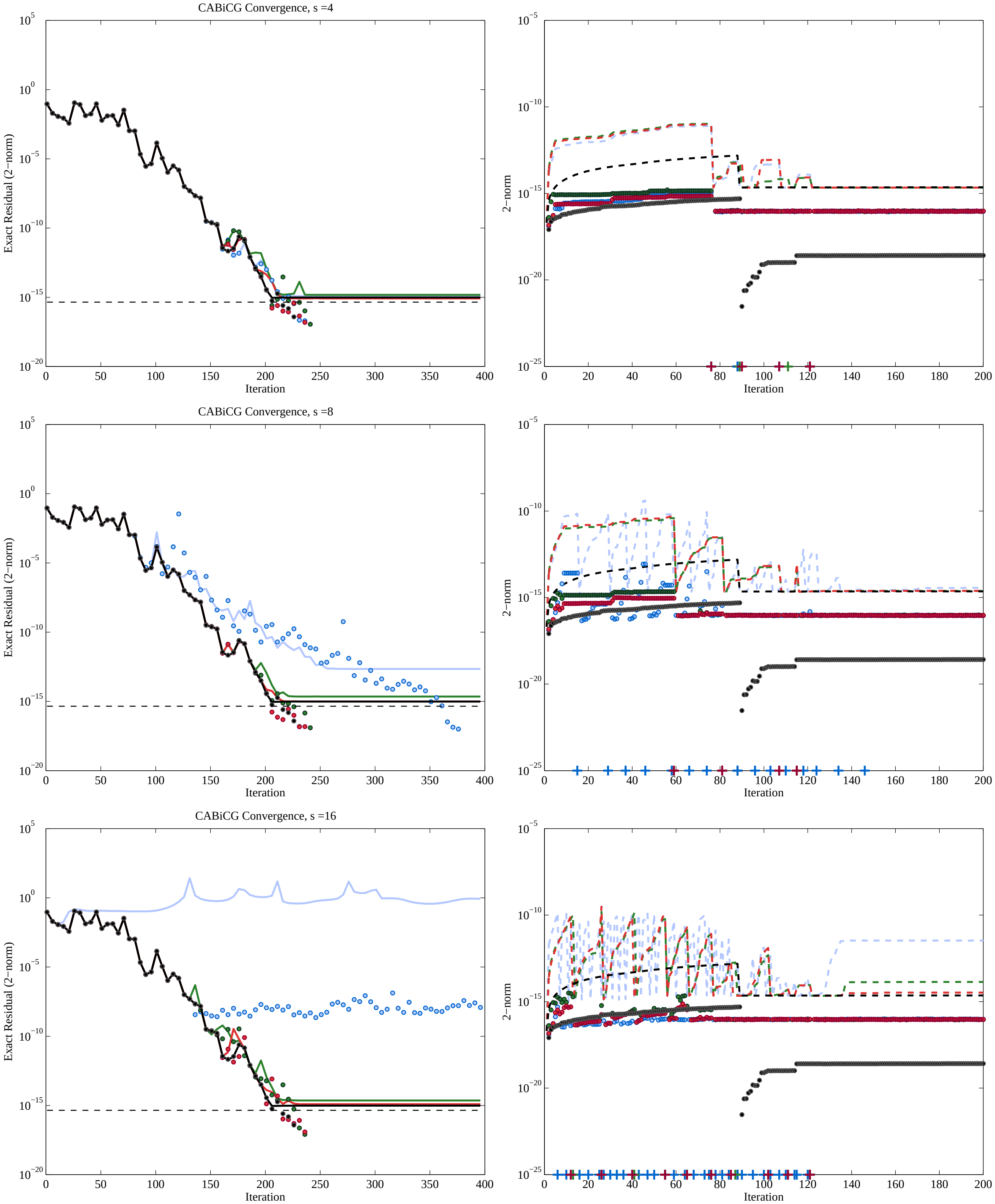

Figure 1: cdde1. Model convection-diffusion differential equations. Non-normal.

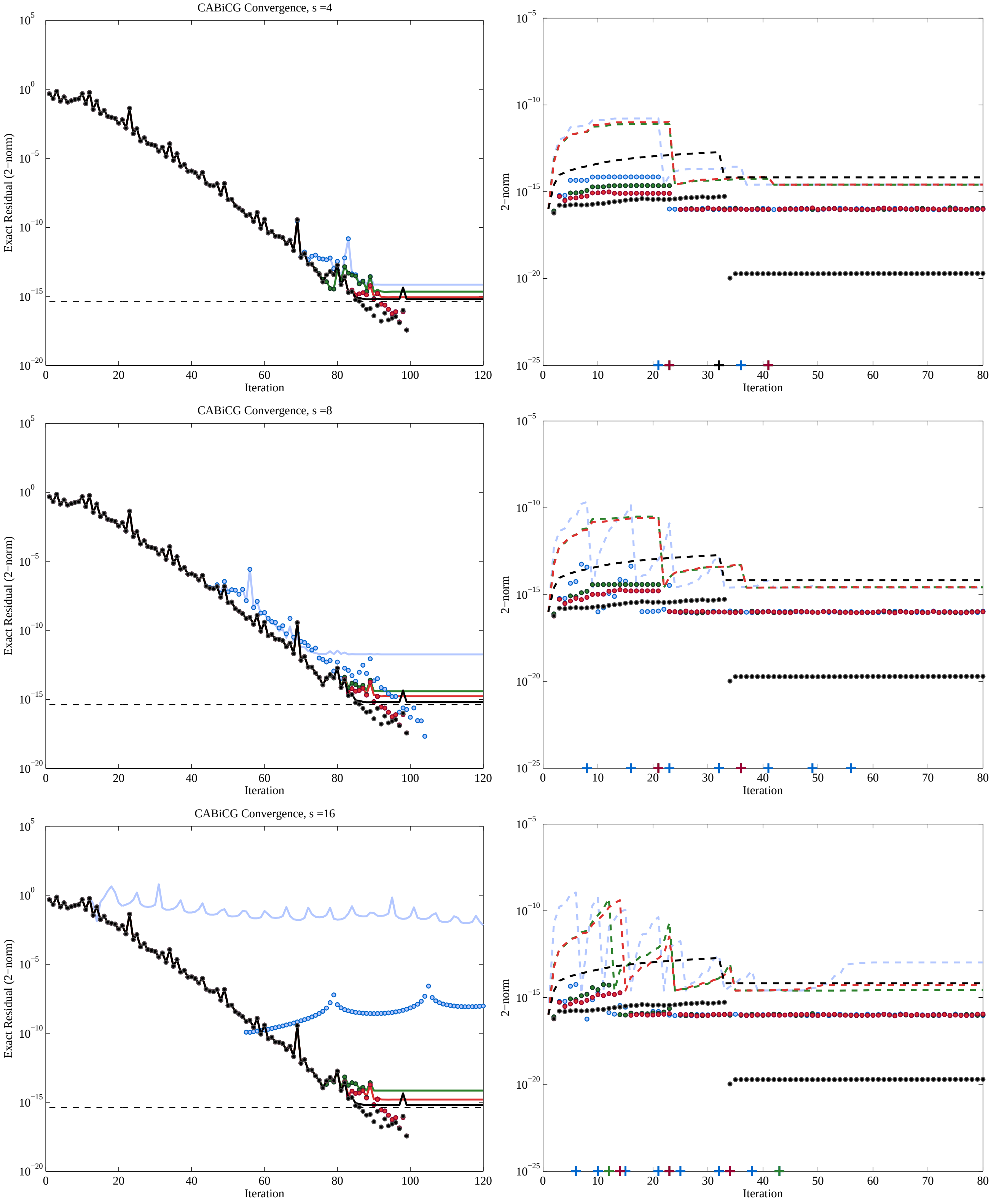

Figure 2: jpwh991. Unsymmetric matrix from Philips, LTD. Semiconductor device problem. Non-normal.

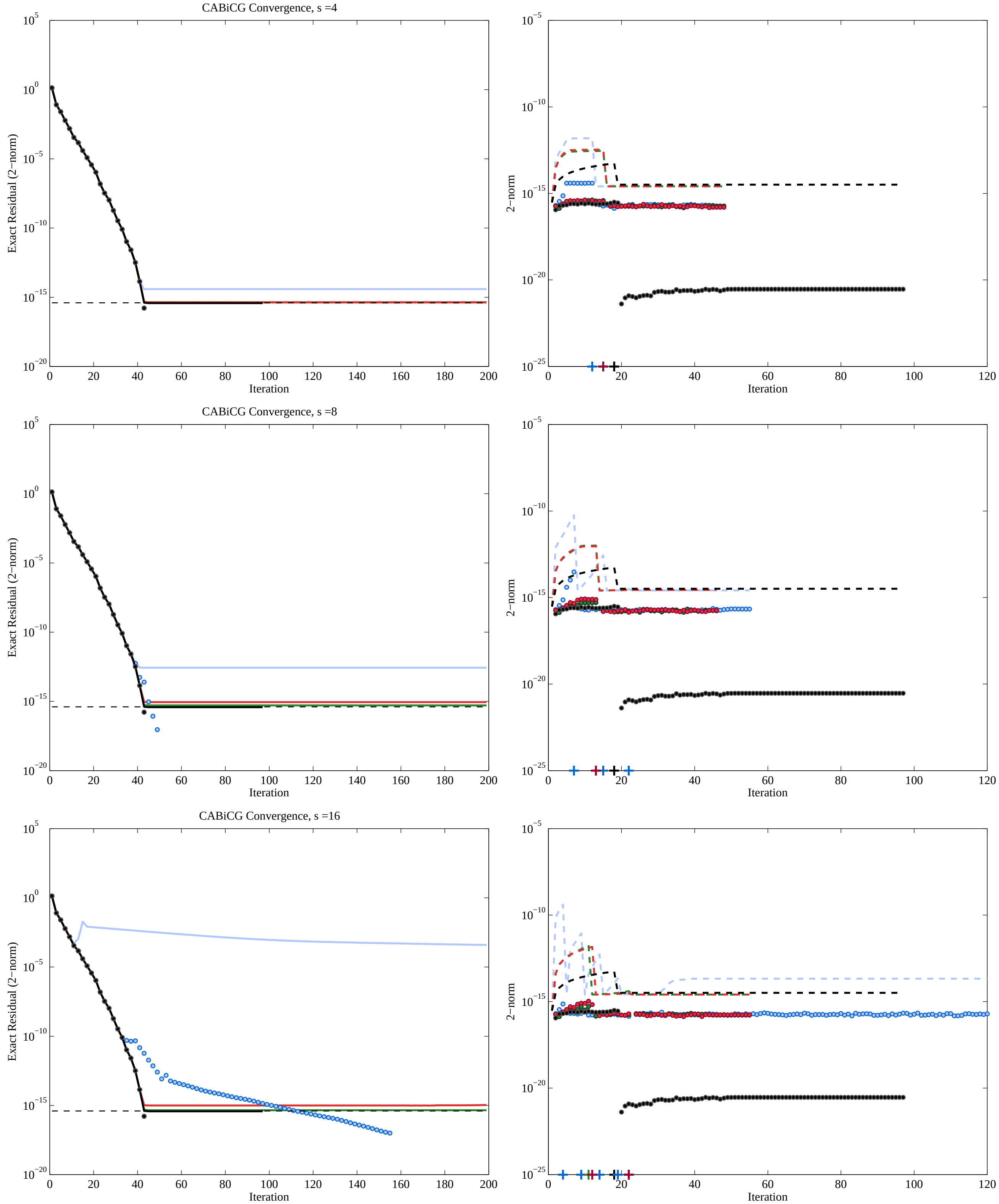

Figure 3: mesh1em1. Structural problem from NASA. SPD.

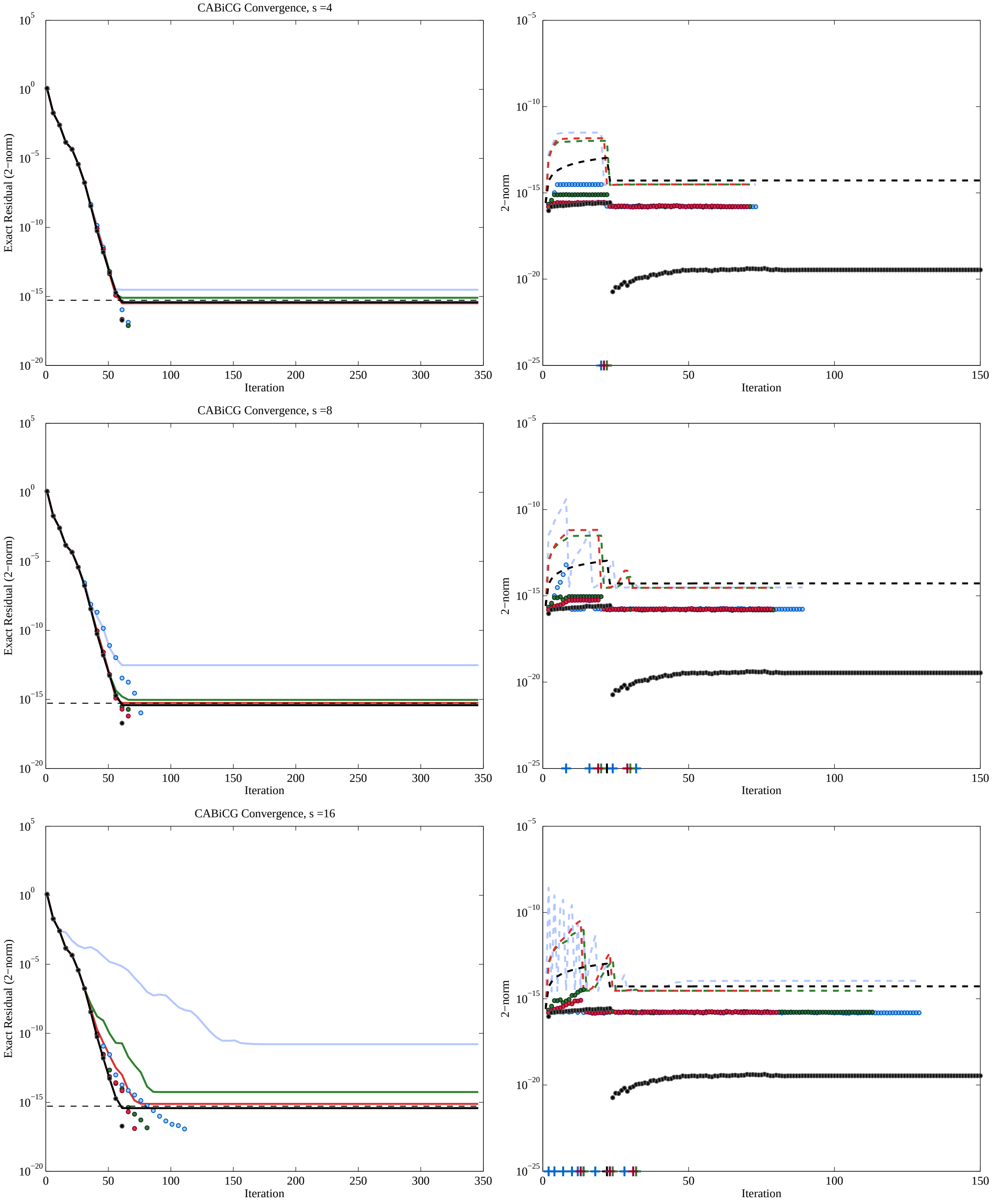

Figure 4: mhdb416. Magneto-hydro-dynamics Alfven spectral problem. SPD.

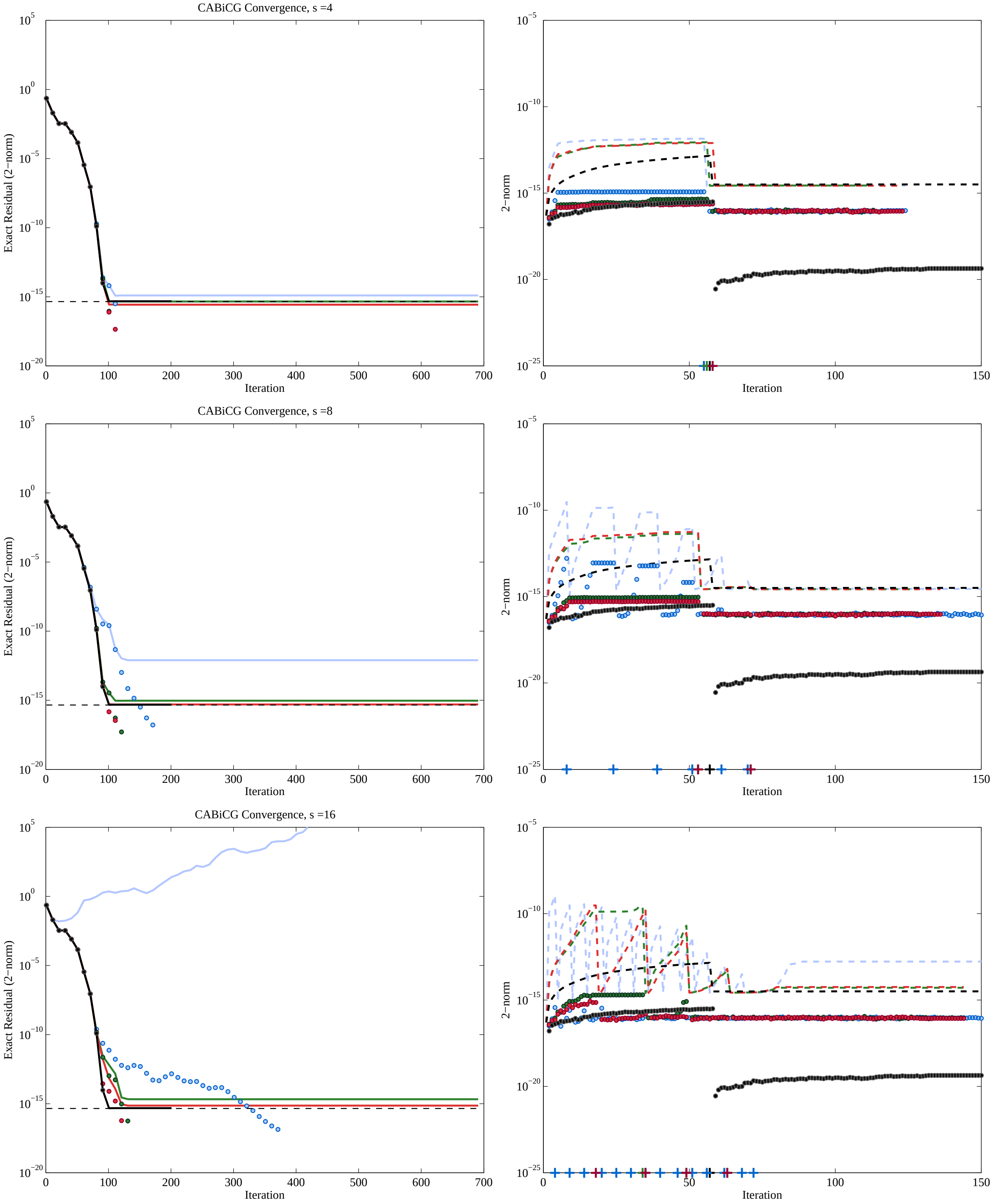

Figure 5: nos4. Beam Structure Model. SPD.

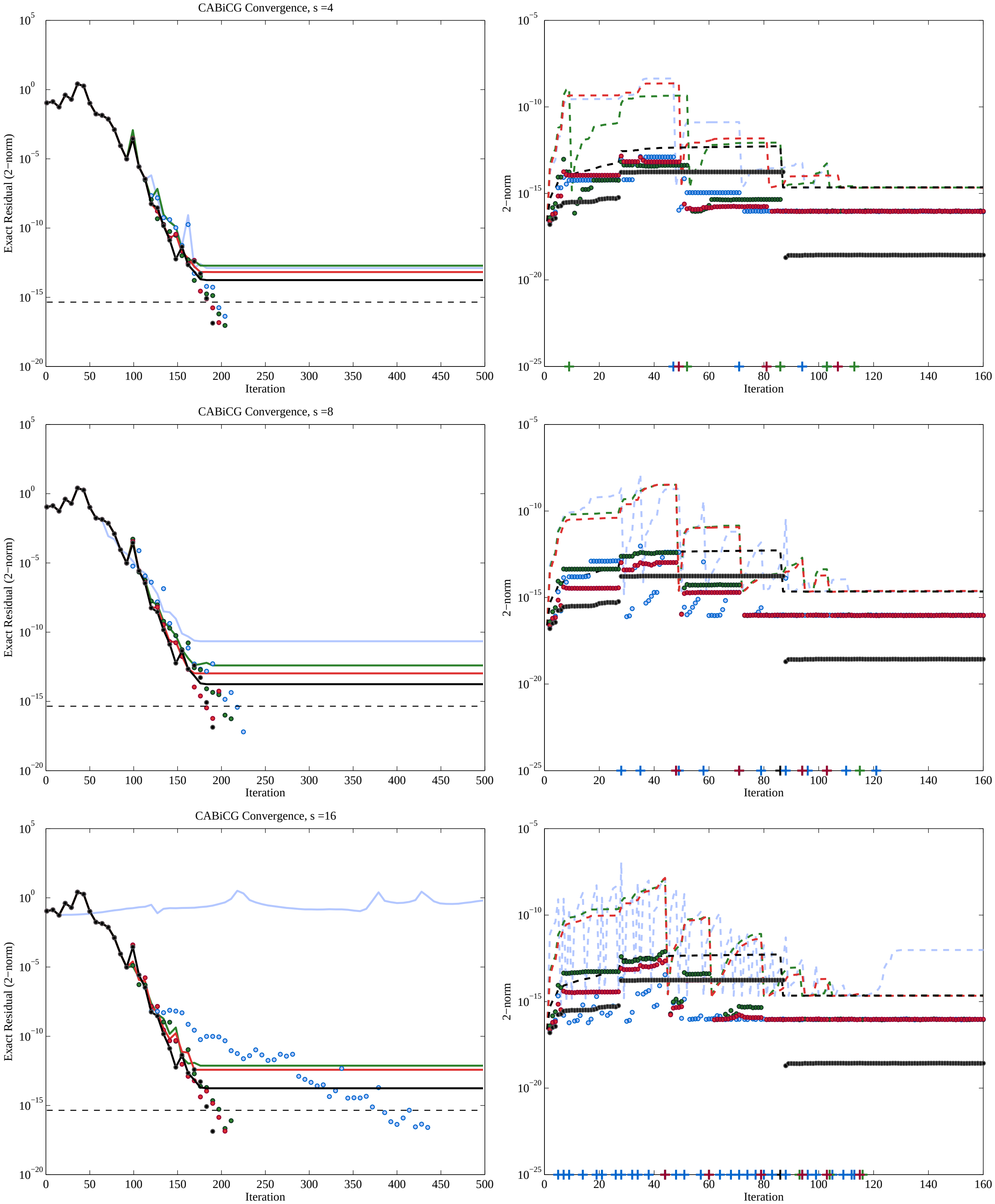

Figure 6: pde900. Model Problem, $N_x = N_y = 30$. Non-normal.

5 Analysis and Conclusions

Our results show that our residual replacement scheme for CA-KSMs is indeed effective, as has been observed for KSMs. For all test cases, the Newton and Chebyshev bases converge to a level considered to be backward stable, with the 2-norm of the true residual equal to $O(\epsilon||A||_2 \cdot ||x||_2)$. From Table 1, we can see that, using the Newton and Chebyshev bases, the number of residual replacements needed to maintain stability grows slowly (linearly) with the basis size, and the total number of replacements is very small compared to the total number of iterations. We expect this behavior because the conditioning of these polynomial bases grows slowly with s for many systems when basis parameters are chosen appropriately [18].

Additionally, our experimental results indicate that residual replacements do not significantly slow the rate of converge for CA-KSMs with the Newton and Chebyshev bases. In fact, in many experiments, the CA-KSM with residual replacement often converges at a faster rate than the CA-KSM without residual replacement. This, combined with the observation that the total number of replacements is small with respect to the total number of iterations, leads us to conclude that, using an appropriate Krylov basis, we can increase the maximum attainable accuracy of CA-KSMs without asymptotically increasing communication or computation costs.

For the monomial basis, however, the basis condition number grows exponentially with s [8], which is reflected by a large number of replacements in our results. We observed that for all test matrices except the last two, nos4 and pde900, the monomial basis became numerically rank deficient at some point during the algorithm for $s = 16$ (denoted by an asterisk in Table 1). Since our generated Krylov subspace is numerically rank deficient here, we expect frequent replacement up to a point (depending on the tolerance parameter $\hat{\epsilon}$). After the residual becomes so small that the condition for replacement is no longer satisfied (designed with the goal that the Lanczos procedure not be disturbed), replacements will stop. At this point, we can't draw meaningful conclusions about the behavior of the monomial basis. In the case of the first two non-normal matrices, cdde1 and jpwh991, the method does not converge in this case. For the two SPD matrices mesh1em1 and mhdb416, however, the method does converge despite the occurrence of a numerically rank-deficient basis. Whether this is due to luck or some underlying properties remains to be determined, and will require closer examination of which iterations were affected by a degenerate subspace. For nos4 and pde900, both numerically full rank for all bases at $s = 16$, the CA-KSM with residual replacement does converge, whereas the CA-KSM without residual replacement does not converge due to round-off error.

Much work remains to be done on the analysis of CA-KSMs in finite precision. In the immediate future, we plan to perform an analysis of the replacement scheme chosen in finite precision to support our experimental results. We also plan to extend this analysis to other CA-KSMs, such as CA-BICGSTAB. This will follow the same general process here, modulo a few extra terms.

We can also extend our error analysis to improve CA-KSM algorithms. One possibility is that we can heuristically determine the maximum s value we can use given ϵ, based on an estimate of basis norm growth. This would allow us to choose an s value that is as large as possible, without requiring a large number of residual replacement steps (which limit our savings in communication). The error analysis performed here could also allow for dynamic selection of s; depending on our error estimate and the frequency of residual replacements, s could be automatically increased or decreased throughout the computation. Many opportunities exist for future research.

References

[1] Z. Bai, D. Hu, and L. Reichel, *A Newton basis GMRES implementation*, IMA Journal of Numerical Analysis **14** (1994), no. 4, 563.

[2] E. Carson, N. Knight, and J. Demmel, *Avoiding communication in two-sided Krylov subspace methods*, Tech. report, Tech. Report, University of California at Berkeley, Berkeley, CA, USA, 2011.

[3] A.T. Chronopoulos and C.W. Gear, *On the efficient implementation of preconditioned s-step conjugate gradient methods on multiprocessors with memory hierarchy*, Parallel computing **11** (1989), no. 1, 37–53.

[4] A.T. Chronopoulos and C.D. Swanson, *Parallel iterative s-step methods for unsymmetric linear systems*, Parallel Computing **22** (1996), no. 5, 623–641.

[5] CW Chronopoulos et al., *S-step iterative methods for symmetric linear systems*, Journal of Computational and Applied Mathematics **25** (1989), no. 2, 153–168.

[6] J. Demmel, M. Hoemmen, M. Mohiyuddin, and K. Yelick, *Avoiding communication in computing Krylov subspaces*, Tech. report, Technical Report UCB/EECS-2007-123, University of California Berkeley EECS, 2007.

[7] D.B. Gannon and J. Van Rosendale, *On the impact of communication complexity on the design of parallel numerical algorithms*, Computers, IEEE Transactions on **100** (1984), no. 12, 1180–1194.

[8] W. Gautschi, *The condition of polynomials in power form*, Math. Comp **33** (1979), no. 145, 343–352.

[9] A. Greenbaum, *Estimating the attainable accuracy of recursively computed residual methods*, SIAM Journal on Matrix Analysis and Applications **18** (1997), 535.

[10] A.C. Hindmarsh and H.F. Walker, *Note on a Householder implementation of the GMRES method*, Tech. report, Lawrence Livermore National Lab., CA (USA); Utah State Univ., Logan (USA). Dept. of Mathematics, 1986.

[11] M. Hoemmen, *Communication-avoiding Krylov subspace methods*, Thesis. UC Berkeley, Department of Computer Science (2010).

[12] W.D. Joubert and G.F. Carey, *Parallelizable restarted iterative methods for nonsymmetric linear systems. Part I: Theory*, International Journal of Computer Mathematics **44** (1992), no. 1, 243–267.

[13] C.E. Leiserson, S. Rao, and S. Toledo, *Efficient out-of-core algorithms for linear relaxation using blocking covers*, Journal of Computer and System Sciences **54** (1997), no. 2, 332–344.

[14] G. Meurant and Z. Strakos, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numerica **15** (2006), 471–542.

[15] G.A. Meurant, *The Lanczos and conjugate gradient algorithms: from theory to finite precision computations*, vol. 19, Society for Industrial Mathematics, 2006.

[16] M. Mohiyuddin, M. Hoemmen, J. Demmel, and K. Yelick, *Minimizing communication in sparse matrix solvers*, Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, ACM, 2009, p. 36.

[17] C.C. Paige, M. Rozioznik, and Z. Strakos, *Modified Gram-Schmidt (MGS), least squares, and backward stability of MGS-GMRES*, SIAM Journal on Matrix Analysis and Applications **28** (2006), no. 1, 264.

[18] B. Philippe and L. Reichel, *On the generation of Krylov subspace bases*, Applied Numerical Mathematics (2011).

[19] J.V. Rosendale, *Minimizing inner product data dependencies in conjugate gradient iteration*, (1983).

[20] G.L.G. Sleijpen and H.A. van der Vorst, *Reliable updated residuals in hybrid Bi-CG methods*, Computing **56** (1996), no. 2, 141–163.

[21] E. Sturler, *A performance model for Krylov subspace methods on mesh-based parallel computers*, Parallel Computing **22** (1996), no. 1, 57–74.

[22] S.A. Toledo, *Quantitative performance modeling of scientific computations and creating locality in numerical algorithms*, Ph.D. thesis, Massachusetts Institute of Technology, 1995.

[23] C.H. Tong and Q. Ye, *Analysis of the finite precision bi-conjugate gradient algorithm for nonsymmetric linear systems*, Math. Comp, Citeseer, 1995.

[24] H.A. Van der Vorst and Q. Ye, *Residual replacement strategies for Krylov subspace iterative methods for the convergence of true residuals*, SIAM Journal on Scientific Computing **22** (1999), no. 3, 835–852.

[25] H.F. Walker, *Implementation of the GMRES method using Householder transformations*, SIAM Journal on Scientific and Statistical Computing **9** (1988), 152.

Appendix A: Round-off in Matrix Powers Computation

We assume the polynomial basis is computed via a 3-term recurrence, as follows:

$$\begin{aligned} \hat{v}_k = fl(v_k) &= \gamma_k (A - \theta_k I)\hat{v}_{k-1} - \sigma_k \hat{v}_{k-2} + \zeta^k \\ &= \gamma_k A \hat{v}_{k-1} - (\gamma_k \theta_k)\hat{v}_{k-1} - \sigma_k \hat{v}_{k-2} + \zeta^k \end{aligned}$$

We wish to find a bound for ζ^k:
$fl(A\hat{v}_{k-1}) = A\hat{v}_{k-1} + \delta_0$
$|\delta_0| \leq m_A \epsilon |A| \cdot |\hat{v}_{k-1}|$

$fl(\gamma_k \cdot fl(A\hat{v}_{k-1})) = \gamma_k A \hat{v}_{k-1} + \delta_1$
$|\delta_1| \leq \epsilon \left(m_A |\gamma_k| \cdot |A| \cdot |\hat{v}_{k-1}| + |\gamma_k A \hat{v}_{k-1}|\right) + O(\epsilon^2)$

$fl(\sigma_k \hat{v}_{k-2}) = \sigma_k \hat{v}_{k-2} + \delta_2$
$|\delta_2| \leq \epsilon |\sigma_k \hat{v}_{k-2}|$

$fl(\gamma_k \theta_k \hat{v}_{k-1}) = \gamma_k \theta_k \hat{v}_{k-1} + \delta_3$
$|\delta_3| \leq \epsilon |(\gamma_k \theta_k)\hat{v}_{k-1}|$

$fl(\, fl((\gamma_k \theta_k)\hat{v}_{k-1}) + fl(\sigma_k \hat{v}_{k-2})\,) = \gamma_k \theta_k \hat{v}_{k-1} + \delta_3 + \sigma_k \hat{v}_{k-2} + \delta_2 + \delta_4 = \gamma_k \theta_k \hat{v}_{k-1} + \sigma_k \hat{v}_{k-2} + \delta_5$
$|\delta_4| \leq \epsilon \left(|\gamma_k \theta_k \hat{v}_{k-1} + \sigma_k \hat{v}_{k-2}|\right) + O(\epsilon^2)$

$$\begin{aligned} |\delta_5| &\leq \epsilon |\sigma_k \hat{v}_{k-2}| + \epsilon |(\gamma_k \theta_k)\hat{v}_{k-1}| + \epsilon \left(|\gamma_k \theta_k \hat{v}_{k-1} + \sigma_k \hat{v}_{k-2}|\right) + O(\epsilon^2) \\ &\leq 2\epsilon \left(|\sigma_k \hat{v}_{k-2}| + |(\gamma_k \theta_k)\hat{v}_{k-1}|\right) + O(\epsilon^2) \end{aligned}$$

$$\begin{aligned} &fl(\, fl(\gamma_k \cdot fl(A\hat{v}_{k-1})) - fl(\, fl((\gamma_k \theta_k)\hat{v}_{k-1}) + fl(\sigma_k \hat{v}_{k-2})\,)\,) = \\ &\qquad \gamma_k A \hat{v}_{k-1} + \delta_1 + \gamma_k \theta_k \hat{v}_{k-1} + \sigma_k \hat{v}_{k-2} + \delta_5 + \delta_6 = \gamma_k A \hat{v}_{k-1} + \gamma_k \theta_k \hat{v}_{k-1} + \sigma_k \hat{v}_{k-2} + \delta_7 \end{aligned}$$

$$|\delta_6| \leq \epsilon \left(|\gamma_k A \hat{v}_{k-1} - \gamma_k \theta_k \hat{v}_{k-1} - \sigma_k \hat{v}_{k-2}|\right) + O(\epsilon^2)$$

$$\begin{aligned} |\delta_7| &\leq \epsilon \left(m_A |\gamma_k| \cdot |A| \cdot |\hat{v}_{k-1}| + |\gamma_k A \hat{v}_{k-1}|\right) + 2\epsilon \left(|\sigma_k \hat{v}_{k-2}| + |(\gamma_k \theta_k)\hat{v}_{k-1}|\right) \\ &\quad + \epsilon \left(|\gamma_k A \hat{v}_{k-1} - \gamma_k \theta_k \hat{v}_{k-1} - \sigma_k \hat{v}_{k-2}|\right) + O(\epsilon^2) \\ &\leq \epsilon \left(|\hat{v}_k| + 2|(\gamma_k \theta_k)\hat{v}_{k-1}| + 2|\sigma_k \hat{v}_{k-2}| + |\gamma_k A \hat{v}_{k-1}| + m_A |\gamma_k| \cdot |A| \cdot |\hat{v}_{k-1}|\right) \end{aligned}$$

$$\implies \boxed{|\zeta^k| \leq \epsilon \left(|\hat{v}_k| + 2|(\gamma_k \theta_k)\hat{v}_{k-1}| + 2|\sigma_k \hat{v}_{k-2}| + |\gamma_k A \hat{v}_{k-1}| + m_A |\gamma_k| \cdot |A| \cdot |\hat{v}_{k-1}|\right)}$$