SIAM J. MATRIX ANAL. APPL.
Vol. 35, No. 1, pp. 22–43

A RESIDUAL REPLACEMENT STRATEGY FOR IMPROVING THE MAXIMUM ATTAINABLE ACCURACY OF s-STEP KRYLOV SUBSPACE METHODS*

ERIN CARSON† AND JAMES DEMMEL‡

Abstract. Krylov subspace methods are a popular class of iterative methods for solving linear systems with large, sparse matrices. On modern computer architectures, both sequential and parallel performance of classical Krylov methods is limited by costly data movement, or *communication*, required to update the approximate solution in each iteration. These motivated communication-avoiding Krylov methods, based on s-step formulations, reduce data movement by a factor of $O(s)$ by reordering the computations in classical Krylov methods to exploit locality. Studies on the finite precision behavior of communication-avoiding Krylov methods in the literature have thus far been empirical in nature; in this work, we provide the first quantitative analysis of the maximum attainable accuracy of communication-avoiding Krylov subspace methods in finite precision. Following the analysis for classical Krylov methods, we derive a bound on the deviation of the true and updated residuals in communication-avoiding conjugate gradient and communication-avoiding biconjugate gradient in finite precision. Furthermore, an estimate for this bound can be iteratively updated within the method without asymptotically increasing communication or computation. Our bound enables an implicit residual replacement strategy for maintaining agreement between residuals to within $O(\epsilon)\|A\|\|x\|$. Numerical experiments on a small set of test matrices verify that, for cases where the updated residual converges, the residual replacement strategy can enable accuracy of $O(\epsilon)\|A\|\|x\|$ with a small number of residual replacement steps, reflecting improvements of up to seven orders of magnitude.

Key words. Krylov subspace methods, maximum attainable accuracy, residual replacement, numerical stability, iterative solvers, minimizing communication, sparse matrix

AMS subject classifications. 65N12, 65N15, 65F10, 65F50, 65Y05, 68W40, 65G50

DOI. 10.1137/120893057

1. Introduction.

Krylov subspace methods (KSMs) are a class of iterative algorithms commonly used for solving the linear system $Ax = b$, where A is large and sparse. In each iteration n, the solution x_n and residual r_n are updated as

$$x_n = x_{n-1} + \alpha_{n-1}p_{n-1}, \quad r_n = r_{n-1} - \alpha_{n-1}Ap_{n-1}, \tag{1.1}$$

or something similar. The above applies specifically to conjugate gradient (CG) and biconjugate gradient (BICG); similar formulas describe steepest descent, conjugate gradient squared (CGS), stabilized biconjugate gradient (BICGSTAB), and other recursively updated residual methods; for a thorough introduction to classical KSMs, see [23]. For simplicity, we restrict our discussion here to linear systems where A is real, square with dimension N, and full rank.

In classical KSM implementations, the updates to x_n and r_n consist of one or more sparse matrix-vector multiplications (SpMVs) and vector operations per iteration. On modern computer architectures, these operations are both *communication bound*; the movement of data, rather than the computation, is the limiting factor in performance.

*Received by the editors September 28, 2012; accepted for publication (in revised form) by J. Liesen October 22, 2013; published electronically January 21, 2014.
http://www.siam.org/journals/simax/35-1/89305.html

†EECS Computer Science Division, University of California, Berkeley, Berkeley, CA 94720-1776 (ecc2z@eecs.berkeley.edu).

‡Mathematics and EECS Computer Science Division, University of California, Berkeley, Berkeley, CA 94720-1776 (demmel@cs.berkeley.edu).

Recent efforts have thus focused on *communication-avoiding* KSMs (CA-KSMs) (see, e.g., [4, 9, 14]), based on s-step KSM formulations (see, e.g., [5, 7, 10, 13, 25, 26, 29]). CA-KSMs reorder the computations in classical KSMs to perform $O(s)$ computation steps of the algorithm for each communication step, allowing an $O(s)$ reduction in total communication cost. In practice, this can translate into significant speedups [19].

The accuracy of classical KSMs in finite precision is studied extensively in the literature (see, e.g., [11, 12, 17, 18, 24, 28]). Such analyses stem from the observation that x_n and r_n in (1.1) have different round-off patterns in finite precision. That is, the expression for x_n does not depend on r_n, nor does the expression for r_n depend on x_n. Therefore, computational errors made in x_n are not self-correcting. These errors can accumulate over many iterations and cause deviation of the *true residual*, $b - Ax_n$, and the *updated residual*, r_n. Writing the true residual as $b - Ax_n = r_n + (b - Ax_n - r_n)$, we can bound its norm by

$$\|b - Ax_n\| \leq \|r_n\| + \|b - Ax_n - r_n\|.$$

When the updated residual r_n is much larger than $b - Ax_n - r_n$, the true residual and the updated residual will be of similar magnitude. However, as the updated residual converges, i.e., $\|r_n\| \to 0$, the size of the true residual depends on $\|b - Ax_n - r_n\|$. This term denotes the size of the deviation between the true and updated residuals. If this deviation grows large, it can limit the *maximum attainable accuracy*, i.e., the accuracy with which we can solve $Ax = b$ on a computer with unit round-off ϵ.

Residual replacement strategies, in which the updated residual is replaced by the true residual in certain iterations, have been shown to maintain $\|b - Ax_n - r_n\| = O(\epsilon)\|A\|\|x\|$ for classical KSMs [28]. The replacement iterations are chosen based on an estimate of $\|b - Ax_n - r_n\|$ such that (1) the size of the deviation of residuals remains small and (2) convergence of the finite precision Lanczos process is undisturbed [24, 28].

In contrast to classical KSMs, the literature lacks an analysis of attainable accuracy in finite precision CA-KSMs. Although many have empirically studied the convergence behavior of s-step formulations (see, e.g., [5, 6, 7, 14, 29, 30]), we are the first, to the best of our knowledge, to provide a quantitative analysis of the maximum attainable accuracy of CA-CG and CA-BICG in finite precision. We derive a computable upper bound on the deviation of the true and updated residual, which allows insight into how the attainable accuracy changes with s and with the choice of polynomial basis.

Furthermore, following the analysis of van der Vorst and Ye [28], this bound enables an efficient residual replacement strategy. We verify that this strategy maintains an agreement between the true and updated residual to within $O(\epsilon)\|A\|\|x\|$ for a small set of test matrices. In all cases, the ratio of residual replacement steps to total number of iterations is negligible (less than 2%), indicating that improved accuracy (up to seven orders of magnitude) can be obtained while still maintaining an asymptotic reduction in communication.

2. Related work. We briefly review work on s-step KSMs and CA-KSMs, as well as work related to analysis of classical KSMs in finite precision.

2.1. s-step Krylov subspace methods. We say "s-step KSM" to refer to variants of Krylov methods where the iteration loop is split into an outer loop and an inner loop, where the inner loop computes s steps of the iteration process for each outer loop. Such formulations have been derived (and rederived) over the past few decades with a number of motivations, namely increasing parallelism [6, 29, 30] and

avoiding data movement between both levels of the memory hierarchy on sequential machines and between processors in parallel environments. A thorough treatment of related work can be found in [14].

Many empirical studies of s-step KSMs found that convergence often deteriorated using $s > 5$ due to the inherent instability of the monomial basis. This motivated research into the use of better-conditioned bases (e.g., Newton or Chebyshev polynomials) for the Krylov subspace, which allowed convergence for higher s values (see, e.g., [3, 13, 15]). Hoemmen has used a novel matrix equilibration and balancing approach to achieve similar effects [14].

We say "*communication-avoiding* KSMs" to refer to s-step KSMs which decrease latency costs by a factor of $O(s)$ over s iterations, including in the s-step basis computation (see [9]). Hoemmen et al. [14, 19] derived communication-avoiding variants of Lanczos, Arnoldi, CG, and GMRES. Details of nonsymmetric Lanczos-based CA-KSMs, such as CA-BICG, CA-CGS, and CA-BICGSTAB can be found in [4].

2.2. Error analysis of Krylov subspace methods. The behavior of classical KSMs in finite precision arithmetic is a well-studied problem. In finite precision, the Lanczos process can result in significant deviation between the updated residual r_n and the true residual, $b - Ax_n$. This limits the maximum attainable accuracy of the solution. Previous authors have devised residual replacement strategies for classical KSMs, which replace the updated residual by the finite precision evaluation of the true residual at carefully chosen iterations. In this way, the agreement between the true and updated residual is maintained within $O(\epsilon)\|A\|\|x\|$.

An upper bound on the maximum attainable accuracy for classical KSMs was provided by Greenbaum [11]. Greenbaum proved that this bound can be determined a priori for CG, but not for methods like BICG, which can have arbitrarily large intermediate iterates. Additionally, Greenbaum has shown that finite precision CG is backward stable in the sense that the Ritz values generated by the finite precision Lanczos process lie in small intervals around the eigenvalues of A. There are many other analyses of the behavior of various KSMs in finite precision (see, e.g., [17, 18, 27], as well as the bibliography in [21]).

Sleijpen and van der Vorst implemented a technique called "flying restarts" to decrease the magnitude of round-off errors in KSMs [24]. In each iteration, an upper bound on the size of the accumulated error is used to determine whether to perform a *group update*, to restart the method, or both. The benefit of a group update strategy is analogous to grouping to reduce round-off error in finite precision recursive summation (see section 4 for more details). Van der Vorst and Ye subsequently devised a residual replacement strategy combined with group updates, which, rather than restarting the method, replaces the updated residual with the computed value of the true residual [28]. The residual replacement iterations are chosen to meet two objectives: first, the accumulated round-off error does not limit the attainable accuracy, and second, the convergence rate of the finite precision Lanczos process is maintained. To determine when these conditions are met, the algorithm iteratively updates an estimate of the accrued error. Our analysis will closely parallel the methodology of [28].

2.3. Communication-avoiding conjugate gradient. We briefly review CA-CG, shown in Algorithm 2. We chose CA-CG for simplicity, although the same general techniques used here can be applied to other CA-KSMs. For reference, CG is given in Algorithm 1. The following brief derivation is meant as a review for the familiar reader; in the interest of space, we refer the unfamiliar reader to numerous other works such as [4, 9, 14, 16, 26].

Algorithm 1. CG.

Given x_0, $r_0 = b - Ax_0$, $p_0 = r_0$
for $n = 1, 2, \ldots,$ until convergence **do**
  $\alpha_{n-1} = r_{n-1}^T r_{n-1} / p_{n-1}^T A p_{n-1}$
  $q_{n-1} = \alpha_{n-1} p_{n-1}$
  $x_n = x_{n-1} + q_{n-1}$
  $r_n = r_{n-1} - A q_{n-1}$
  $\beta_{n-1} = r_n^T r_n / r_{n-1}^T r_{n-1}$
  $p_n = r_n + \beta_{n-1} p_{n-1}$
end for
return x_n, r_n

CA-CG consists of an outer loop, indexed by k, and an inner loop, which iterates from $j = 1 : s$, where we assume $s \ll N$. For clarity, we globally index iterations by $n = sk + j$. Our goal is to determine the information required to compute the CG vectors p_{sk+j}, r_{sk+j}, and x_{sk+j} for $1 \le j \le s$ and $s > 0$, given p_{sk}, r_{sk}, and x_{sk}. It follows from the properties of CG that

$$\begin{aligned} &p_{sk+j}, r_{sk+j} \in \mathcal{K}_{j+1}(A, p_{sk}) + \mathcal{K}_j(A, r_{sk}) \quad \text{and} \\ &x_{sk+j} - x_{sk} \in \mathcal{K}_j(A, p_{sk}) + \mathcal{K}_{j-1}(A, r_{sk}), \end{aligned} \tag{2.1}$$

where $\mathcal{K}_j(A, v) \equiv \text{span}(v, Av, \ldots, A^{j-1}v)$ is the Krylov subspace of dimension j with matrix A and starting vector v. Since $\mathcal{K}_j(A, p_{sk}) \subseteq \mathcal{K}_{j+1}(A, p_{sk})$ (similarly for subspaces with starting vector r_{sk}), we can compute the CG vectors for iterations $sk+1$ through $sk+s$ using the matrices

$$\begin{aligned} P_k &= [\rho_0(A)p_{sk}, \ldots, \rho_s(A)p_{sk}] \quad \text{such that} \quad \text{span}(P_k) = \mathcal{K}_{s+1}(A, p_{sk}) \quad \text{and} \\ R_k &= [\rho_0(A)r_{sk}, \ldots, \rho_{s-1}(A)r_{sk}] \quad \text{such that} \quad \text{span}(R_k) = \mathcal{K}_s(A, r_{sk}), \end{aligned}$$

where $\rho_j(z)$ is a polynomial of degree j which satisfies the 3-term recurrence formula

$$\begin{aligned} &\rho_0(z) = 1, \qquad \rho_1(z) = (z - \theta_0)\rho_0(z)/\gamma_0, \quad \text{and} \\ &\rho_j(z) = \left((z - \theta_{j-1})\rho_{j-1}(z) - \sigma_{j-2}\rho_{j-2}(z)\right)/\gamma_{j-1}. \end{aligned} \tag{2.2}$$

Let $\underline{V}_k = [P_k, R_k]$, and let V_k be the same as $\underline{V}_k$ except with all zeros in columns $s+1$ and $2s+1$. Under certain assumptions on the nonzero structure of A, $\underline{V}_k$ can be computed in each outer loop for the same asymptotic latency cost as a single SpMV (see [9, 19] for details). Since the columns of $\underline{V}_k$ satisfy (2.2), we can write

$$AV_k = \underline{V}_k \underline{B}_k. \tag{2.3}$$

Note that $\underline{B}_k$ is a $(2s+1) \times (2s+1)$ tridiagonal matrix; we use this notation to avoid confusion with the tridiagonal matrix of Lanczos coefficients generated by CG.

Let $p'_{k,j}$, $r'_{k,j}$, and $x'_{k,j}$ represent the coordinates for the CG iterates p_{sk+j}, r_{sk+j}, and $x_{sk+j} - x_{sk}$, respectively, in $\underline{V}_k$. That is,

$$p_{sk+j} = \underline{V}_k p'_{k,j}, \quad r_{sk+j} = \underline{V}_k r'_{k,j}, \quad \text{and} \quad x_{sk+j} - x_{sk} = \underline{V}_k x'_{k,j}. \tag{2.4}$$

The CG iterate updates, for $1 \le j \le s$, can then be written

$$\begin{gathered} \underline{V}_k x'_{k,j} = \underline{V}_k x'_{k,j-1} + \alpha_{sk+j-1} \underline{V}_k p'_{k,j-1}, \quad \underline{V}_k r'_{k,j} = \underline{V}_k r'_{k,j-1} - \alpha_{sk+j-1} A \underline{V}_k p'_{k,j-1}, \\ \text{and} \quad \underline{V}_k p'_{k,j} = \underline{V}_k r'_{k,j} + \beta_{sk+j-1} \underline{V}_k p'_{k,j-1}. \end{gathered}$$

Algorithm 2. CA-CG.

Given x_0, $r_0 = b - Ax_0$, $p_0 = r_0$
for $k = 0, 1, \ldots,$ until convergence **do**
  Calculate $\underline{V}_k = [P_k,\ R_k]$ with starting vectors p_{sk} and r_{sk}, $G_k = \underline{V}_k^T \underline{V}_k$
  $p'_{k,0} = [1,\ 0_{1,2s}]^T$, $r'_{k,0} = [0_{1,s+1},\ 1,\ 0_{1,s-1}]^T$, $x'_{k,0} = [0_{1,2s+1}]^T$
  for $j = 1 : s$ **do**
    $\alpha_{sk+j-1} = \left(r'^T_{k,j-1} G_k r'_{k,j-1}\right) / \left(p'^T_{k,j-1} G_k \underline{B}_k p'_{k,j-1}\right)$
    $q'_{k,j-1} = \alpha_{sk+j-1} p'_{k,j-1}$
    $x'_{k,j} = x'_{k,j-1} + q'_{k,j-1}$
    $r'_{k,j} = r'_{k,j-1} - \underline{B}_k q'_{k,j-1})$
    $\beta_{sk+j-1} = \left(r'^T_{k,j} G_k r'_{k,j}\right) / \left(r'^T_{k,j-1} G_k r'_{k,j-1}\right)$
    $p'_{k,j} = r'_{k,j} + \beta_{sk+j-1} p'_{k,j-1}$
  end for
  $x_{sk+s} = V_k x'_{k,s} + x_{sk}$, $r_{sk+s} = \underline{V}_k r'_{k,s}$, $p_{sk+s} = \underline{V}_k p'_{k,s}$
end for
return $x_{sk+j} = x_{sk} + V_k x'_{k,j}$, $r_{sk+j} = \underline{V}_k r'_{k,j}$

By (2.1), $\underline{V}_k p'_{k,j-1} = V_k p'_{k,j-1}$ for $1 \le j \le s$. We then use (2.3) to obtain

$$\underline{V}_k r'_{k,j} = \underline{V}_k r'_{k,j-1} - \alpha_{sk+j-1} \underline{V}_k \underline{B}_k p'_{k,j-1}.$$

Therefore, in the inner loop of CA-CG, rather than update the CG vectors explicitly, we instead update their coordinates in $\underline{V}_k$, i.e.,

$$x'_{k,j} = x'_{k,j-1} + \alpha_{sk+j-1} p'_{k,j-1}, \quad r'_{k,j} = r'_{k,j-1} - \alpha_{sk+j-1} \underline{B}_k p'_{k,j-1},$$
$$\text{and} \quad p'_{k,j} = r'_{k,j} + \beta_{sk+j-1} p'_{k,j-1}.$$

Thus we can eliminate the communication cost of the SpMV with A in each iteration, and the length-$(2s+1)$ vector updates can be performed locally (parallel)/in cache (sequential) in each inner loop iteration.

We can also avoid communication in computing inner products. Letting $G_k = \underline{V}_k^T \underline{V}_k$, and using (2.4) and (2.3), α_{sk+j-1} and β_{sk+j-1} can be computed by

$$\alpha_{sk+j-1} = \left(r'^T_{k,j-1} G_k r'_{k,j-1}\right) / \left(p'^T_{k,j-1} G_k \underline{B}_k p'_{k,j-1}\right) \quad \text{and}$$
$$\beta_{sk+j-1} = \left(r'^T_{k,j} G_k r'_{k,j}\right) / \left(r'^T_{k,j-1} G_k r'_{k,j-1}\right).$$

The matrix G_k can be computed by one reduction per outer loop, the same asymptotic latency cost as a single inner product (see section 5.4.5 in [14]). As $\underline{B}_k$ and G_k are dimension $(2s+1) \times (2s+1)$, α_{sk+j-1} and β_{sk+j-1} can be computed locally/in cache within the inner loop.

3. Finite precision CA-CG. Throughout our analysis, we use a standard model of floating point arithmetic:

$$\text{fl}(x + y) = x + y + \delta \quad \text{with } |\delta| \le \epsilon(|x + y|),$$
$$\text{fl}(Ax) = Ax + \delta \quad \text{with } |\delta| \le \epsilon N_A |A| |x|,$$

where ϵ is the unit round-off of the machine, $x, y \in \mathbb{R}^N$, and N_A is a constant associated with the matrix-vector multiplication (e.g., the maximal number of nonzero

entries in a row of A). All absolute values and inequalities are componentwise. Using this model, we can also write

$$\mathrm{fl}(y + Ax) = y + Ax + \delta \quad \text{with } |\delta| \leq \epsilon(|y + Ax| + N_A\,|A|\,|x|),$$

where, as in the remainder this analysis, we ignore higher powers of ϵ. We now perform an analysis of round-off error in the CA-CG method. Our goal is to bound the norm of the difference between the true and updated residual in terms of quantities which can be computed inexpensively in each iteration. We note that our bound also holds for CA-BICG, which has identical formulas for computing p_{sk+j}, r_{sk+j}, and $x_{sk+j} - x_{sk}$.

3.1. Basis computation in finite precision. We begin by bounding the error in finite precision computation of $\underline{\hat{V}}_k$, where, here and in the remaining analysis, we use hats to denote quantities computed in finite precision. In exact arithmetic, (2.3) holds, but in finite precision, we have $A\hat{V}_k = \underline{\hat{V}}_k\underline{B}_k - E_k$, where E_k represents the error due to round-off. We seek a componentwise bound on E_k. The analysis below is similar to that for the finite precision Lanczos process (see, e.g., Theorem 3.1 in [1]).

In finite precision, we can write

$$\underline{\hat{V}}_k = \big[\hat{P}_k,\, \hat{R}_k\big] = \big[\hat{\rho}_0(A)\hat{p}_{sk},\, \ldots,\, \hat{\rho}_s(A)\hat{p}_{sk},\, \hat{\rho}_0(A)\hat{r}_{sk},\, \ldots,\, \hat{\rho}_{s-1}(A)\hat{r}_{sk}\big], \tag{3.1}$$

where $\hat{\rho}_i$ is a polynomial of degree i. In terms of starting vector y and parameters γ_i, θ_i, and σ_i, in finite precision (2.2) becomes

$$\hat{\rho}_{i+1}(A)y = \frac{1}{\gamma_i}\Big((A - \theta_i I)\hat{\rho}_i(A)y - \sigma_i\hat{\rho}_{i-1}(A)y\Big) + \nu_{y,i+1}, \tag{3.2}$$

where

$$|\nu_{y,i+1}| \leq \epsilon\Big((2 + N_A)\,|A|\,|\hat{\rho}_i(A)y| + 3\,|\theta_i|\,|\hat{\rho}_i(A)y| + 2\,|\sigma_i|\,|\hat{\rho}_{i-1}(A)y|\Big)/\,|\gamma_i|\,. \tag{3.3}$$

We can rearrange (3.2) to obtain

$$A\hat{\rho}_i(A)y = \gamma_i\hat{\rho}_{i+1}(A)y + \theta_i\hat{\rho}_i(A)y + \sigma_i\hat{\rho}_{i-1}(A)y - \gamma_i\nu_{y,i+1}. \tag{3.4}$$

Premultiplying (3.1) by A and combining (3.4) with (3.3) gives the desired expression, $A\hat{V}_k = \underline{\hat{V}}_k\underline{B}_k - E_k$, with

$$\big|E_k\big| \leq \epsilon\Big((3 + N_A)\big|A\big|\big|\hat{V}_k\big| + 4\big|\underline{\hat{V}}_k\big|\big|\underline{B}_k\big|\Big). \tag{3.5}$$

3.2. Iterate updates in finite precision. We now consider the finite precision error in updating $\hat{x}'_{k,j}$ and $\hat{r}'_{k,j}$ in each inner loop iteration and performing the basis change to obtain $\hat{x}_{sk+j}$ and $\hat{r}_{sk+j}$, which occurs when $j = s$ or when $sk + j$ is the terminal iteration. In the inner loop, the formulas for updating the coordinates are

$$x'_{k,j} = x'_{k,j-1} + q'_{k,j-1} \quad \text{and} \tag{3.6}$$

$$r'_{k,j} = r'_{k,j-1} - \underline{B}_k q'_{k,j-1}, \tag{3.7}$$

where $q'_{k,j-1} = \alpha_{sk+j-1}p'_{k,j-1}$. When (3.6) and (3.7) are implemented in finite precision, they become

$$\hat{x}'_{k,j} = \mathrm{fl}\left(\hat{x}'_{k,j-1} + \hat{q}'_{k,j-1}\right) = \hat{x}'_{k,j-1} + \hat{q}'_{k,j-1} + \xi_{k,j}, \tag{3.8}$$

$$\text{where} \quad \big|\xi_{k,j}\big| \leq \epsilon\big|\hat{x}'_{k,j}\big|, \quad \text{and} \tag{3.9}$$

$$\hat{r}'_{k,j} = \mathrm{fl}\left(\hat{r}'_{k,j-1} - \underline{B}_k\hat{q}'_{k,j-1}\right) = \hat{r}'_{k,j-1} - \underline{B}_k\hat{q}'_{k,j-1} + \eta_{k,j}, \tag{3.10}$$

$$\text{where} \quad \big|\eta_{k,j}\big| \leq \epsilon\Big(\big|\hat{r}'_{k,j}\big| + N_B\big|\underline{B}_k\big|\big|\hat{q}'_{k,j-1}\big|\Big). \tag{3.11}$$

If $j < s$ and $sk+j$ is not the terminal iteration, we denote (but do not compute) the solution and updated residual by

$$x_{sk+j} = \hat{\underline{V}}_k \hat{x}'_{k,j} + \hat{x}_{sk} \quad \text{and} \tag{3.12}$$

$$r_{sk+j} = \hat{\underline{V}}_k \hat{r}'_{k,j}. \tag{3.13}$$

At the end of each outer loop iteration (when $j = s$) and at the terminal iteration, the solution and residual are computed in finite precision, denoted $\hat{x}_{sk+j}$ and $\hat{r}_{sk+j}$, respectively. The error in computing these terms can be bounded as follows:

$$\hat{x}_{sk+j} = \mathrm{fl}\left(\hat{\underline{V}}_k \hat{x}'_{k,j} + \hat{x}_{sk}\right) = \hat{\underline{V}}_k \hat{x}'_{k,j} + \hat{x}_{sk} + \phi_{sk+j} = x_{sk+j} + \phi_{sk+j}, \tag{3.14}$$

$$\text{where} \quad \left|\phi_{sk+j}\right| \le \epsilon\Big(\left|\hat{x}_{sk+j}\right| + N_V \left|\hat{\underline{V}}_k\right|\left|\hat{x}'_{k,j}\right|\Big), \quad \text{and} \tag{3.15}$$

$$\hat{r}_{sk+j} = \mathrm{fl}\left(\hat{\underline{V}}_k \hat{r}'_{k,j}\right) = \hat{\underline{V}}_k \hat{r}'_{k,j} + \psi_{sk+j} = r_{sk+j} + \psi_{sk+j}, \tag{3.16}$$

$$\text{where} \quad \left|\psi_{sk+j}\right| \le \epsilon N_V \left|\hat{\underline{V}}_k\right|\left|\hat{r}'_{k,j}\right|. \tag{3.17}$$

3.3. Deviation of the true and updated residual. Using the results in previous subsections, we obtain an upper bound on the norm of the difference between the true and updated residuals at step $sk+j$. We first prove the following lemma.

LEMMA 3.1. *Consider step $sk+j$ of the finite precision CA-CG algorithm. Let $\delta_{sk+j} = b - Ax_{sk+j} - r_{sk+j}$, where x_{sk+j} and r_{sk+j} are defined in* (3.12) *and* (3.13)*, respectively. Then*

$$\begin{aligned}
\left\|\delta_{sk+j}\right\| \le \left\|b - Ax_0 - r_0\right\| &+ \epsilon \sum_{\ell=0}^{k-1} \left(\|A\| \left\| |\hat{x}_{s\ell+s}| \right\| + N_V \Big(\|A\| \left\| |\hat{\underline{V}}_\ell| |\hat{x}'_{\ell,s}| \right\| + \left\| |\hat{\underline{V}}_\ell| |\hat{r}'_{\ell,s}| \right\| \Big) \right) \\
&+ \epsilon \sum_{\ell=0}^{k-1} \sum_{i=1}^{s} \left(C_1 \|A\| \left\| |\hat{V}_\ell| |\hat{x}'_{\ell,i}| \right\| + C_2 \left\| |\hat{\underline{V}}_\ell| |\underline{B}_\ell| |\hat{x}'_{\ell,i}| \right\| + \left\| |\hat{\underline{V}}_\ell| |\hat{r}'_{\ell,i}| \right\| \right) \\
&+ \epsilon \sum_{i=1}^{j} \left(C_1 \|A\| \left\| |\hat{V}_k| |\hat{x}'_{k,i}| \right\| + C_2 \left\| |\hat{\underline{V}}_k| |\underline{B}_k| |\hat{x}'_{k,i}| \right\| + \left\| |\hat{\underline{V}}_k| |\hat{r}'_{k,i}| \right\| \right),
\end{aligned}$$

where $C_1 = 7 + 2N_A$ and $C_2 = 8 + 2N_B$.

Proof. We can write δ_{sk+j} as

$$\begin{aligned}
\delta_{sk+j} &= b - Ax_{sk+j} - r_{sk+j} \\
&= b - A(\hat{x}_{sk} + \hat{\underline{V}}_k \hat{x}'_{k,j}) - \hat{\underline{V}}_k \hat{r}'_{k,j} \\
&= b - A\hat{x}_{sk} - A\hat{V}_k(\hat{x}'_{k,j-1} + \hat{q}'_{k,j-1} + \xi_{k,j}) - \hat{\underline{V}}_k(\hat{r}'_{k,j-1} - \underline{B}_k \hat{q}'_{k,j-1} + \eta_{k,j}) \\
&= b - A(\hat{x}_{sk} + \hat{V}_k \hat{x}'_{k,j-1}) - \hat{\underline{V}}_k \hat{r}'_{k,j-1} - A\hat{V}_k \xi_{k,j} - \hat{\underline{V}}_k \eta_{k,j} + E_k \hat{q}'_{k,j-1} \\
&= \delta_{sk+j-1} - A\hat{V}_k \xi_{k,j} - \hat{\underline{V}}_k \eta_{k,j} + E_k \hat{q}'_{k,j-1} \\
&= (b - A\hat{x}_{sk} - \hat{r}_{sk}) - \sum_{i=1}^{j} \left(A\hat{V}_k \xi_{k,i} + \hat{\underline{V}}_k \eta_{k,i} - E_k \hat{q}'_{k,i-1} \right) \\
&= \delta_0 - \sum_{\ell=0}^{k-1} \left(A\phi_{s\ell+s} + \psi_{s\ell+s} + \sum_{i=1}^{s} \left(A\hat{V}_\ell \xi_{\ell,i} + \hat{\underline{V}}_\ell \eta_{\ell,i} - E_\ell \hat{q}'_{\ell,i-1} \right) \right) \\
&\quad - \sum_{i=1}^{j} \left(A\hat{V}_k \xi_{k,i} + \hat{\underline{V}}_k \eta_{k,i} - E_k \hat{q}'_{k,i-1} \right).
\end{aligned} \tag{3.18}$$

Using the componentwise error bounds in (3.5), (3.9), (3.11), (3.15), and (3.17),

$$\begin{aligned}|\delta_{sk+j}| \le |b - Ax_0 - r_0| &+ \epsilon \sum_{\ell=0}^{k-1} \left(|A||\hat{x}_{s\ell+s}| + N_V \left(|A||\underline{\hat{V}}_\ell||\hat{x}'_{\ell,s}| + |\underline{\hat{V}}_\ell||\hat{r}'_{\ell,s}| \right) \right) \\ &+ \epsilon \sum_{\ell=0}^{k-1} \sum_{i=1}^{s} \left((7+2N_A)|A||\hat{V}_\ell||\hat{x}'_{\ell,i}| + (8+2N_B)|\underline{\hat{V}}_\ell||\underline{B}_\ell||\hat{x}'_{\ell,i}| + |\underline{\hat{V}}_\ell||\hat{r}'_{\ell,i}| \right) \\ &+ \epsilon \sum_{i=1}^{j} \left((7+2N_A)|A||\hat{V}_k||\hat{x}'_{k,i}| + (8+2N_B)|\underline{\hat{V}}_k||\underline{B}_k||\hat{x}'_{k,i}| + |\underline{\hat{V}}_k||\hat{r}'_{k,i}| \right), \end{aligned} \tag{3.19}$$

where we have used $\sum_{i=1}^{j} |\hat{q}'_{k,i-1}| \le (2+\epsilon) \sum_{i=1}^{j} |\hat{x}'_{k,i}|$. From this follows the desired bound for δ_{sk+j} in terms of norms. ∎

Assume the algorithm terminates at step $sk+j$, so $\hat{x}_{sk+j}$ and $\hat{r}_{sk+j}$ are returned. We bound the deviation of the true residual, $b - A\hat{x}_{sk+j}$, and the updated residual, $\hat{r}_{sk+j}$, in the following theorem.

THEOREM 3.2. *Consider step $sk+j$ of finite precision CA-CG. If the algorithm terminates at step $sk+j$ and returns solution $\hat{x}_{sk+j}$ and updated residual $\hat{r}_{sk+j}$, the norm of the deviation of the true residual and the updated residual, denoted $\hat{\delta}_{sk+j} = b - A\hat{x}_{sk+j} - \hat{r}_{sk+j}$, can be bounded by*

$$\|\hat{\delta}_{sk+j}\| \le \|\delta_{sk+j}\| + \epsilon \Big(\|A\| \|\hat{x}_{sk}\| + (1+N_V)\|A\| \Big\| |\underline{\hat{V}}_k||\hat{x}'_{k,j}| \Big\| + N_V \Big\| |\underline{\hat{V}}_k||\hat{r}'_{k,j}| \Big\| \Big),$$

where $C_1 = 7 + 2N_A$ and $C_2 = 8 + 2N_B$.

Proof. Using (3.14) and (3.16), we obtain

$$\begin{aligned} b - A\hat{x}_{sk+j} - \hat{r}_{sk+j} &= b - A(x_{sk+j} + \phi_{sk+j}) - (r_{sk+j} + \psi_{sk+j}) \\ &= \delta_{sk+j} - A\phi_{sk+j} - \psi_{sk+j}, \end{aligned}$$

and by (3.15), we can write $\|A\phi_{sk+j}\| \le \epsilon \|A\| \left(\|\hat{x}_{sk}\| + (1+N_V) \| |\underline{\hat{V}}_k||\hat{x}'_{k,j}| \| \right)$. Combining this bound with (3.17) and Lemma 3.1 gives the desired result. ∎

It is beneficial to use the quantities $\| |\underline{\hat{V}}_\ell||\underline{B}_\ell||\hat{x}'_{\ell,i}| \|$, $\| |\underline{\hat{V}}_\ell||\hat{x}'_{\ell,i}| \|$, and $\| |\underline{\hat{V}}_\ell||\hat{r}'_{\ell,i}| \|$ above, as $\|\underline{\hat{V}}_\ell\| \|\underline{B}_\ell\| \|\hat{x}'_{\ell,i}\|$, $\|\underline{\hat{V}}_\ell\| \|\hat{x}'_{\ell,i}\|$, and $\|\underline{\hat{V}}_\ell\| \|\hat{r}'_{\ell,i}\|$, respectively, can be overestimates due to sparsity in $\hat{x}'_{\ell,i}$ and $\hat{r}'_{\ell,i}$, respectively, for $i < s$.

Comparing the bound in Theorem 3.2 with the bound in [28], we expect the size of the deviation of the true and updated residual at iteration $sk+j$ in CA-CG to be on the same order as that in CG when $\| |\underline{\hat{V}}_\ell||\hat{x}'_{\ell,i}| \| \approx \|\underline{\hat{V}}_\ell \hat{x}'_{\ell,i}\|$, $\| |\underline{\hat{V}}_\ell||\hat{r}'_{\ell,i}| \| \approx \|\underline{\hat{V}}_\ell \hat{r}'_{\ell,i}\|$, and $\| |\underline{\hat{V}}_\ell||\underline{B}_\ell||\hat{x}'_{\ell,i}| \| \approx \| |A||\underline{\hat{V}}_\ell||\hat{x}'_{\ell,i}| \|$. This indicates that the choice of basis $\underline{\hat{V}}_k$ plays a significant role in determining the accuracy of CA-CG relative to CG.

4. Finite precision CA-CG with residual replacement and group update. We first discuss the residual replacement strategy of van der Vorst and Ye [28], on which our residual replacement strategy is based. In [28], at *replacement steps* $m = m_1, m_2, \ldots, m_y$, the updated residual r_m is replaced with the true residual and a *group update* is performed.

A group update strategy ensures that the deviation of residuals remains bounded by $O(\epsilon)\|A\|\|x\|$ from the last replacement step m_y to the final iteration by reducing error in the local recurrence. Similar techniques are used to reduce error accumulation in evaluating a sum $S = \sum_i w_i$ of small numbers by direct recursive additions [28].

This error can be corrected by grouping operations as $S_1 + S_2 + \cdots = (w_1 + w_2 + \cdots + w_{m_1}) + (w_{m_1+1} + \cdots + w_{m_2}) + \cdots$, where terms close in magnitude are grouped in S_i. This makes rounding error associated with additions within group S_i of magnitude ϵS_i, which can be much smaller than ϵS. This strategy has previously been suggested for use in KSMs by Neumaier and was also used by Sleijpen and van der Vorst (see [20, 24, 28]), where the solution is instead updated by the recurrence

$$x_n = x_0 + \sum_{i=1}^{n} q_{i-1} = x_0 + (q_0 + \cdots + q_{m_1-1}) + \cdots .$$

We use the same strategy as [28], i.e., when a residual replacement occurs, a group update is also performed. In iteration $sk + j + m$, where m was the last residual replacement step, we denote (but do not compute) the group solution z_{sk+j+m} as the sum of the group solution computed in step m, z_m, and the current *partial solution*, $x_{sk+j} = x_{sk} + V_k x'_{k,j}$, i.e., $z_{sk+j+m} = z_m + x_{sk} + V_k x'_{k,j}$ (see Algorithm 3).

If $sk + j + m$ is a residual replacement step, we compute $\hat{z}_{sk+j+m}$ and set the true residual to $\mathrm{fl}(b - A\hat{z}_{sk+j+m})$. As subsequent iterates cannot be represented in $\underline{\hat{V}}_k$, a replacement step requires starting a new outer loop iteration, potentially before completing s iterations of the inner loop. Thus when replacement occurs we reset $k = j = 0$ and $x_{sk+j} = x_0 = 0$. Although additional outer loop iterations require additional communication, our results show that the number of replacements required is small with respect to the total number of iterations, and thus extra cost is negligible.

As described in [28], we seek to determine iterations where replacing the updated residual with the true residual does not alter the rate of convergence, based on a bound on potential perturbation to the finite precision Lanczos recurrence. We briefly review the discussion in [28], which motivates the condition for residual replacement.

4.1. Selecting residual replacement steps. Consider finite precision classical CG, where, in iteration n, the updated residuals $\hat{r}_n$ and search directions $\hat{p}_n$ satisfy

$$\hat{r}_n = \hat{r}_{n-1} - \alpha_{n-1} A\hat{p}_{n-1} + \eta_n \qquad \text{and} \qquad \hat{p}_n = \hat{r}_n + \beta_{n-1}\hat{p}_{n-1} + \tau_n. \tag{4.1}$$

Let e_n denote the nth identity column of appropriate size. We can then write the above equations in matrix form as

$$AZ_n = Z_n T_n - \frac{1}{\alpha'_n}\frac{\hat{r}_{n+1}}{\|\hat{r}_0\|}e_{n+1}^T + F_n, \quad \text{with} \quad Z_n = \left[\frac{\hat{r}_0}{\|\hat{r}_0\|}, \ldots, \frac{\hat{r}_n}{\|\hat{r}_n\|}\right],$$

where T_n is invertible and tridiagonal, $\alpha'_n = e_n^T T_n^{-1} e_1$, and $F_n = [f_0, \ldots, f_n]$, with

$$f_n = \frac{A\tau_n}{\|\hat{r}_n\|} + \frac{1}{\alpha_n}\frac{\eta_{n+1}}{\|\hat{r}_n\|} - \frac{\beta_{n-1}}{\alpha_{n-1}}\frac{\eta_n}{\|\hat{r}_n\|}. \tag{4.2}$$

It has been shown (see, e.g., [17]) that if $\hat{r}_n$ satisfies (4.1) and Z_{n+1} is full rank,

$$\|\hat{r}_{n+1}\| \leq (1 + K_n) \min_{\rho \in \mathcal{P}_n, \rho(0)=1} \|\rho(A + \Delta A_n)\hat{r}_0\|, \tag{4.3}$$

where $\mathcal{P}_n$ is the set of polynomials of degree n, $K_n = \left\|(AZ_n - F_n)T_n^{-1}\right\| \left\|Z_{n+1}^+\right\|$ and $\Delta A_n = -F_n Z_n^+$. Therefore, regardless of how $\hat{r}_n$ is generated, if it satisfies (4.1), we can bound its norm by (4.3). Then by (4.2), we know we can replace the updated residual with the true residual without affecting the convergence rate when η_n is not too large relative to $\|\hat{r}_n\|$ and $\|\hat{r}_{n-1}\|$.

We will use $\eta^{\text{RR}}_{sk+j+m}$ to denote the perturbation term in CA-CG with residual replacement and group update strategies, analogous to the η_n term in (4.1). We seek a bound on the norm of $\eta^{\text{RR}}_{sk+j+m}$. Comparing this bound with the norms $\|r_{sk+j+m}\|$ and $\|r_{sk+j+m-1}\|$ of the updated residuals in CA-CG with residual replacement will then allow us to select safe residual replacement steps using the following criteria:

$$\|\eta^{\text{RR}}_{sk+j+m-1}\| \leq \hat{\epsilon}\|r_{sk+j+m-1}\| \quad \text{and} \quad \|\eta^{\text{RR}}_{sk+j+m}\| > \hat{\epsilon}\|r_{sk+j+m}\|. \tag{4.4}$$

Above, $\hat{\epsilon}$ is a tolerance parameter. Because $\hat{\epsilon}$ controls perturbations to the Lanczos recurrence, it should be chosen as small as possible. However, if it is too small, residual replacement will terminate early in the iteration and the accumulated error after the last replacement can become significant [28]. The value $\hat{\epsilon} = \sqrt{\epsilon}$ has been found to balance these two constraints for standard KSMs [28].

4.2. Bounding the error term. We will now describe and analyze the error in CA-CG with residual replacement and group update strategies (see Algorithm 3). Note that in this analysis, we index iterations as $sk + j + m$, where m is the last residual replacement iteration and $sk + j$ is defined as before, where $k, j = 0$ are reset immediately after a replacement step. The goal of this section is a bound on the perturbation term $\eta^{\text{RR}}_{sk+j+m}$ in the recurrence for the updated residual in CA-CG with residual replacement and group update,

$$\hat{r}_{sk+j+m} = r_{sk+j+m-1} - Aq_{sk+j+m-1} + \eta^{\text{RR}}_{sk+j+m}.$$

This will enable us to determine when we can replace the updated residual with the true residual without affecting convergence. We denote the current group solution

$$z_{sk+j+m} = \begin{cases} \hat{z}_{sk+j+m} = \text{fl}\left(\hat{z}_m + \hat{x}_{sk+j}\right) = \hat{z}_m + \hat{x}_{sk+j} + \zeta_{sk+j+m}, & \text{RR step}, \\ \hat{z}_m + \hat{x}_{sk+j} = \hat{z}_m + x_{sk+j} + \phi_{sk+j}, & j = s, \\ \hat{z}_m + x_{sk+j} = \hat{z}_m + \hat{x}_{sk} + \underline{\hat{V}}_k \hat{x}'_{k,j}, & 1 \leq j < s, \end{cases} \tag{4.5}$$

where $\hat{z}_m$ is the approximate solution computed in the last group update in step m, $x_{sk+j} = \hat{x}_{sk} + \underline{\hat{V}}_k \hat{x}'_{k,j}$, $\hat{x}_{sk+j} = x_{sk+j} + \phi_{sk+j}$, and

$$|\zeta_{sk+j+m}| \leq \epsilon|\hat{z}_{sk+j+m}| \leq \epsilon\Big(|\hat{z}_m| + |\hat{x}_{sk}| + |\underline{\hat{V}}_k \hat{x}'_{k,j}|\Big). \tag{4.6}$$

We use r_{sk+j+m} to denote the residual in CA-CG with residual replacement and group update, defined as

$$r_{sk+j+m} = \begin{cases} \hat{r}_{sk+j+m} = \text{fl}\left(b - A\hat{z}_{sk+j+m}\right) = b - A\hat{z}_{sk+j+m} + \mu_{sk+j+m}, & \text{RR step}, \\ \hat{r}_{sk+j+m} = \text{fl}(\underline{\hat{V}}_k \hat{r}'_{k,j}) = \underline{\hat{V}}_k \hat{r}'_{k,j} + \psi_{sk+j}, & j = s, \\ r_{sk+j+m} = \underline{\hat{V}}_k \hat{r}'_{k,j}, & 1 \leq j < s, \end{cases} \tag{4.7}$$

where

$$|\mu_{sk+j+m}| \leq \epsilon\Big(|\hat{r}_{sk+j+m}| + N_A|A||\hat{z}_{sk+j+m}|\Big). \tag{4.8}$$

After the last replacement step m, but before the next replacement step, we denote $\delta_{sk+j+m} = b - Az_{sk+j+m} - r_{sk+j+m}$. In step m we compute $\hat{r}_m = \text{fl}\left(b - A\hat{z}_m\right) = b - A\hat{z}_m + \mu_m$. Since $r_m = \underline{\hat{V}}_0 \hat{r}'_{0,0} = \hat{r}_m$ and $z_m = \hat{z}_m + \hat{x}_0 + \underline{\hat{V}}_0 \hat{x}'_{0,0} = \hat{z}_m$,

$\delta_m = b - Az_m - r_m = b - A\hat{z}_m - \hat{r}_m = -\mu_m$. We now state the main result of this section.

THEOREM 4.1. *Consider finite precision CA-CG with residual replacement and group update, where iteration $sk+j+m$ is a replacement step and m was the last replacement step. The recurrence for the updated residual satisfies*

$$\hat{r}_{sk+j+m} = r_{sk+j+m-1} - Aq_{sk+j+m-1} + \eta^{RR}_{sk+j+m},$$

where

$$\eta^{RR}_{sk+j+m} = -E_k\hat{q}'_{k,j} + \underline{\hat{V}}_k\eta_{k,j} + \delta_{sk+j+m} - A\phi_{sk+j} - A\zeta_{sk+j+m} + \mu_{sk+j+m}$$

and

(4.9)

$$\begin{aligned}\|\eta^{RR}_{sk+j+m}\| \le \epsilon&\Big(\|\hat{r}_m\| + (1+2N')\|A\|\|\hat{z}_m\|\Big)\\ &+ \epsilon\sum_{\ell=0}^{k-1}\left(\|A\|\|\hat{x}_{s\ell+s}\| + C_1\|A\|\left\||\underline{\hat{V}}_\ell||\hat{x}'_{\ell,s}|\right\| + N'\left\||\underline{\hat{V}}_\ell||\hat{r}'_{\ell,s}|\right\|\right)\\ &+ \epsilon\sum_{\ell=0}^{k-1}\sum_{i=1}^{s}\left(C_2\|A\|\left\||\hat{V}_\ell||\hat{x}'_{\ell,i}|\right\| + C_3\left\||\underline{\hat{V}}_\ell||\underline{B}_\ell||\hat{x}'_{\ell,i}|\right\| + \left\||\underline{\hat{V}}_\ell||\hat{r}'_{\ell,i}|\right\|\right)\\ &+ \epsilon\sum_{i=1}^{j}\left(C_2\|A\|\left\||\hat{V}_k||\hat{x}'_{k,i}|\right\| + C_3\left\||\underline{\hat{V}}_k||\underline{B}_k||\hat{x}'_{k,i}|\right\| + \left\||\underline{\hat{V}}_k||\hat{r}'_{k,i}|\right\|\right),\end{aligned}$$

with $C_1 = 2+2N'$, $C_2 = 7+2N'$, and $C_3 = 8+2N'$, where $N' = \max(N_A, N_B, N_V)$.

Proof. If iteration $sk+j+m$ is a replacement step, we compute

$$\begin{aligned}\hat{z}_{sk+j+m} &= \mathrm{fl}(\hat{z}_m + \hat{x}_{sk+j}) = z_{sk+j+m} + \phi_{sk+j} + \zeta_{sk+j+m},\\ \hat{r}_{sk+j+m} &= \mathrm{fl}(b - A\hat{z}_{sk+j+m}) = b - A\hat{z}_{sk+j+m} + \mu_{sk+j+m}\\ &= b - Az_{sk+j+m} - A\phi_{sk+j} - A\zeta_{sk+j+m} + \mu_{sk+j+m}\\ &= r_{sk+j+m} + \delta_{sk+j+m} - A\phi_{sk+j} - A\zeta_{sk+j+m} + \mu_{sk+j+m}\\ &= \underline{\hat{V}}_k\hat{r}'_{k,j-1} - \underline{\hat{V}}_k\underline{B}_k\hat{q}'_{k,j-1} + \underline{\hat{V}}_k\eta_{k,j} + \delta_{sk+j+m} - A\phi_{sk+j} - A\zeta_{sk+j+m} + \mu_{sk+j+m}\\ &= r_{sk+j+m-1} - Aq_{sk+j+m-1} + \eta^{\mathrm{RR}}_{sk+j+m}\end{aligned}$$

with

$$\eta^{\mathrm{RR}}_{sk+j+m} = -E_k\hat{q}'_{k,j} + \underline{\hat{V}}_k\eta_{k,j} + \delta_{sk+j+m} - A\phi_{sk+j} - A\zeta_{sk+j+m} + \mu_{sk+j+m}. \tag{4.10}$$

Following the derivation of (3.18), we can write δ_{sk+j+m} as

$$\begin{aligned}\delta_{sk+j+m} = -\mu_m &- \sum_{\ell=0}^{k-1}\left(A\phi_{s\ell+s} + \psi_{s\ell+s} + \sum_{i=1}^{s}\left(A\hat{V}_\ell\xi_{\ell,i} + \underline{\hat{V}}_\ell\eta_{\ell,i} - E_\ell\hat{q}'_{\ell,i-1}\right)\right)\\ &- \sum_{i=1}^{j}\left(A\hat{V}_k\xi_{k,i} + \underline{\hat{V}}_k\eta_{k,i} - E_k\hat{q}'_{k,i-1}\right),\end{aligned} \tag{4.11}$$

and then

$$\begin{aligned}\delta_{sk+j+m} &- E_k\hat{q}'_{k,j-1} + \underline{\hat{V}}_k\eta_{k,j}\\ &= -\mu_m - \sum_{\ell=0}^{k-1}\left(A\phi_{s\ell+s} + \psi_{s\ell+s} + \sum_{i=1}^{s}\left(A\hat{V}_\ell\xi_{\ell,i} + \underline{\hat{V}}_\ell\eta_{\ell,i} - E_\ell\hat{q}'_{\ell,i-1}\right)\right)\\ &\quad - \sum_{i=1}^{j}A\hat{V}_k\xi_{k,i} - \sum_{i=1}^{j-1}\left(\underline{\hat{V}}_k\eta_{k,i} - E_k\hat{q}'_{k,i-1}\right).\end{aligned} \tag{4.12}$$

Substituting (4.12) into (4.10), and using bounds in (3.5), (3.9), (3.11), (3.15), (3.17), and (4.8), as well as $|\hat{x}_{sk}| \le \sum_{\ell=0}^{k-1}|\underline{\hat{V}}_\ell||\hat{x}'_{\ell,s}| + O(\epsilon)$, we obtain the following componentwise bound:

$$\begin{aligned}\left|\eta^{\mathrm{RR}}_{sk+j+m}\right| \le{}& \epsilon\left(|\hat{r}_m| + (1+2N')\,|A|\,|\hat{z}_m|\right)\\ &+ \epsilon\sum_{\ell=0}^{k-1}\left(|A||\hat{x}_{s\ell+s}| + (2+2N')|A||\underline{\hat{V}}_\ell||\hat{x}'_{\ell,s}| + N'|\underline{\hat{V}}_\ell||\hat{r}'_{\ell,s}|\right)\\ &+ \epsilon\sum_{\ell=0}^{k-1}\sum_{i=1}^{s}\left((7+2N')|A||\hat{V}_\ell||\hat{x}'_{\ell,i}| + |\underline{\hat{V}}_\ell|\left((8+2N')|\underline{B}_\ell||\hat{x}'_{\ell,i}| + |\hat{r}'_{\ell,i}|\right)\right)\\ &+ \epsilon\sum_{i=1}^{j}\left((7+2N')|A||\hat{V}_k||\hat{x}'_{k,i}| + |\underline{\hat{V}}_k|\left((8+2N')|\underline{B}_k||\hat{x}'_{k,i}| + |\hat{r}'_{k,i}|\right)\right)\end{aligned}$$

with $N' = \max(N_A, N_B, N_V)$. The desired bound on the norm of $\eta^{\mathrm{RR}}_{sk+j+m}$ follows. ∎

We can compute (4.9) in each iteration, and thus this bound can be used in the residual replacement condition (4.4). The following section discusses computation and communication costs associated with computing an estimate for (4.9).

4.3. Residual replacement algorithm for CA-CG. In each iteration, we will update an estimate for $\|\eta^{\mathrm{RR}}_{sk+j+m}\|$, which we denote d_{sk+j+m}. Based on (4.9), we will iteratively update the estimate d_{sk+j+m} by

$$\begin{aligned}d_{sk+j+m} &= d_{sk+j+m-1}\\ &+ \epsilon\left((4+N')\left(\|A\|\left\||\hat{V}_k||\hat{x}'_{k,j}|\right\| + \left\||\underline{\hat{V}}_k||\underline{B}_k||\hat{x}'_{k,j}|\right\|\right) + \left\||\underline{\hat{V}}_k||\hat{r}'_{k,j}|\right\|\right) \qquad (4.13)\\ &+ \epsilon\begin{cases}\|A\|\left(\|\hat{x}_{sk+s}\| + (2+2N')\left\||\underline{\hat{V}}_k||\hat{x}'_{k,s}|\right\|\right) + N'\left\||\underline{\hat{V}}_k||\hat{r}'_{k,s}|\right\|, & j = s,\\ 0 & \text{otherwise,}\end{cases} \qquad (4.14)\end{aligned}$$

where we omit the factors of two due to the bound $\sum_{i=1}^{j}|\hat{q}'_{k,i-1}| \le (2+\epsilon)\sum_{i=1}^{j}|\hat{x}'_{k,i}|$, which is pessimistic. At each replacement step m, the value of d_m is reset to $d_m = \epsilon\left(\|\hat{r}_m\| + (1+2N')\,\|A\|\,\|\hat{z}_m\|\right)$.

Based on (4.4), and using (4.7), we perform a residual replacement and group update step when

$$d_{sk+j+m-1} \le \hat{\epsilon}\left\|\underline{\hat{V}}_k\hat{r}'_{k,j-1}\right\|_2, \quad d_{sk+j+m} > \hat{\epsilon}\left\|\underline{\hat{V}}_k\hat{r}'_{k,j}\right\|_2, \text{ and } d_{sk+j+m} > 1.1d_{\mathrm{init}}, \tag{4.15}$$

where the third condition is recommended in [28] to ensure that the error has nontrivially increased since the last replacement step in an effort to avoid unnecessary

Algorithm 3. CA-CG with residual replacement (CA-CG-RR).

Given: $\|A\|_2$, x_0
$z_0 = x_0$, $x_0 = 0$, $r_0 = p_0 = b - Az_0$, $d_0 = d_{\text{init}} = \epsilon(\|r_0\|_2 + N' \|A\|_2 \|z_0\|_2)$, RR $= 0$
for $k = 0, 1, \ldots,$ until convergence **do**
  Calculate $\underline{V}_k = [P_k, R_k]$ with starting vectors p_{sk+m} and r_{sk+m}, $G_k = \underline{V}_k^T \underline{V}_k$
  $p'_{k,0} = [1, 0_{1,2s}]^T$, $r'_{k,0} = [0_{1,s+1}, 1, 0_{1,s-1}]^T$, $x'_{k,0} = [0_{1,2s+1}]^T$
  for $j = 1 : s$ **do**
    $\alpha_{sk+j-1} = \left(r'^T_{k,j-1} G_k r'_{k,j-1}\right) / \left(p'^T_{k,j-1} G_k \underline{B}_k p'_{k,j-1}\right)$
    $q'_{k,j-1} = \alpha_{sk+j-1} p'_{k,j-1}$
    $x'_{k,j} = x'_{k,j-1} + q'_{k,j-1}$
    $r'_{k,j} = r'_{k,j-1} - \underline{B}_k q'_{k,j-1}$
    $\beta_{sk+j-1} = \left(r'^T_{k,j} G_k r'_{k,j}\right) / \left(r'^T_{k,j-1} G_k r'_{k,j-1}\right)$
    $p'_{k,j} = r'_{k,j} + \beta_{sk+j-1} p'_{k,j-1}$
    Update d_{sk+j+m} by (4.13)
    if condition (4.15) holds **then**
      $z_{sk+j+m} = z_m + x_{sk} + \underline{V}_k x'_{k,j}$, $r_{sk+j+m} = b - Az_{sk+j+m}$, $x_0 = 0$
      $d_{\text{init}} = d_{sk+j+m} = \epsilon\left(\|r_{sk+j+m}\|_2 + (1 + 2N') \|A\|_2 \|z_{sk+j+m}\|_2\right)$
      RR $= 1$, $m = m + sk + j$, $k = j = 0$, **break**
    end if
  end for
  if RR $= 0$ **then**
    $x_{sk+s} = \underline{V}_k x'_{k,s} + x_{sk}$, $r_{sk+s+m} = \underline{V}_k r'_{k,s}$
    Update d_{sk+s+m} by (4.14)
  end if
  $p_{sk+j+m} = \underline{V}_k p'_{k,s}$, RR $= 0$
end for
return $z_{sk+j+m} = z_m + x_{sk} + \underline{V}_k x'_{k,j}$, $r_{sk+j+m} = \underline{V}_k r'_{k,j}$

replacements. If this statement is true, we set $z_{sk+j+m} = \hat{z}_{sk+j+m} = \text{fl}(\hat{z}_m + \hat{x}_{sk+j})$, and we set $r_{sk+j+m} = \hat{r}_{sk+j+m} = \text{fl}(b - A\hat{z}_{sk+j+m})$, as in (4.5) and (4.7).

Note that we use $\left\|\hat{\underline{V}}_k \hat{r}'_{k,j}\right\|_2 = (\hat{r}'^T_{k,j} G_k \hat{r}'_{k,j})^{1/2} + O(\epsilon)$ to estimate the other side of the inequalities in (4.15). Since these terms are multiplied by $\hat{\epsilon} = \epsilon^{1/2}$ in (4.15), the $O(\epsilon)$ error due to use of G_k is ignored.

A residual replacement step does incur additional costs in CA-CG. When a residual replacement occurs, we must break from the inner loop (perhaps before completing s steps) and compute $\hat{z}_{sk+j+m}$ and $\hat{r}_{sk+j+m} = \text{fl}(b - A\hat{z}_{sk+j+m})$ (the communication cost depends on the data layout/structure of A). The algorithm must then begin a new outer loop, generating new s-step bases with respect to the replaced residual.

If the number of replacements is large (i.e., we compute the true residual every iteration), this can result in a significant slowdown due to increased communication costs. Fortunately, our experimental results in the next section demonstrate that the number of replacements performed can be low compared to the total number of iterations. The CA-CG algorithm with residual replacement is shown in Algorithm 3.

4.4. Avoiding communication in residual replacement. To iteratively update (4.13) and (4.14) we must be able to inexpensively estimate $\left\||\hat{\underline{V}}_k||\underline{B}_k||\hat{x}'_{k,j}|\right\|$, $\left\||\hat{\underline{V}}_k||\hat{x}'_{k,j}|\right\|$, $\left\||\hat{\underline{V}}_k||\hat{r}'_{k,j}|\right\|$, $\|A\|$, $\|\hat{r}_m\|$, $\|\hat{z}_m\|$, and $\|\hat{x}_{sk+s}\|$ in each inner iteration of

CA-CG. We assume we have an estimate for $\|A\|$, which need only be computed once. We discuss how to obtain the remaining quantities without increasing the asymptotic computation or communication cost of CA-CG.

Let $\tilde{G}_k$ be the $(2s+1) \times (2s+1)$ matrix $\left|\underline{\hat{V}}_k\right|^T \left|\underline{\hat{V}}_k\right|$. Given $\tilde{G}_k$, we can compute

$$\left\| |\underline{\hat{V}}_k| |\hat{x}'_{k,j}| \right\|_2 = \sqrt{|\hat{x}'_{k,j}|^T \tilde{G}_k |\hat{x}'_{k,j}|} + O(\epsilon), \quad \left\| |\underline{\hat{V}}_k| |\hat{r}'_{k,j}| \right\|_2 = \sqrt{|\hat{r}'_{k,j}|^T \tilde{G}_k |\hat{r}'_{k,j}|} + O(\epsilon),$$

$$\text{and} \quad \left\| |\underline{\hat{V}}_k| |\underline{B}_k| |\hat{x}'_{k,j}| \right\|_2 = \sqrt{|\hat{x}'_{k,j}|^T |\underline{B}_k|^T \tilde{G}_k |\underline{B}_k| |\hat{x}'_{k,j}|} + O(\epsilon). \tag{4.16}$$

In each outer loop, the computation of $\tilde{G}_k$ requires a single synchronization (summing a list of $O(s) \times O(s)$ matrices). As we perform an equivalent reduction to compute G_k in each outer loop of CA-CG, computing $\tilde{G}_k$ increases the communication and computation costs by no worse than a factor of two. If the reduction to obtain $\tilde{G}_k$ can be done simultaneously with that to obtain G_k, no additional latency is incurred.

The computation of quantities in (4.16) require $O(s^3)$ operations per s steps and no communication. Again, we perform equivalent operations to compute α_{sk+j-1} and β_{sk+j-1} in each inner loop of CA-CG, so the computation cost is increased by a constant factor. Note that, until now, we have not assumed a specific norm. The 2-norm allows us to compute inner products locally using $\tilde{G}_k$, and thus maintains the communication-avoiding properties of CA-CG.

We must also compute $\|\hat{z}_m\|_2$ and $\|\hat{r}_m\|_2$ at every residual replacement step m. The quantity $\|\hat{r}_m\|_2$ can be computed in the next outer loop after replacement in the same way as the computation of α_{sk+j-1} in CA-CG, i.e., $\|\hat{r}_m\|_2 = (\hat{r}'^T_{0,0} G_0 \hat{r}'_{0,0})^{1/2} + O(\epsilon)$. The computation of $\|\hat{z}_m\|_2$ can be accomplished similarly by fusing the reduction with that of G_0 in the outer loop immediately following a residual replacement step. Again, this increases the bandwidth by a constant factor and does not increase the latency cost. Since

$$\|\hat{x}_{sk+s}\|_2 \leq \sum_{\ell=0}^{k} \left\| \underline{\hat{V}}_\ell \hat{x}'_{\ell,s} \right\|_2 + O(\epsilon) = \sum_{\ell=0}^{k} \sqrt{\hat{x}'^T_{\ell,s} G_\ell \hat{x}'_{\ell,s}} + O(\epsilon),$$

an estimate for this quantity can be iteratively updated in each outer loop iteration using only local quantities.

5. Numerical experiments. We evaluated our residual replacement strategy using many matrices from the University of Florida Sparse Matrix Collection [8] and the NEP Collection [2], including the 28 matrices tested by van der Vorst and Ye [28]. We present a small subset of our results, from various problem domains, which are representative of the general behavior observed. We chose to show results for matrices with large N as this is where we expect CA-KSMs to be beneficial in terms of performance. The selected matrices along with relevant properties are listed in Table 1. We tested both symmetric positive definite (SPD) and nonsymmetric matrices, using CA-CG and CA-BICG, respectively. All matrices except cdde have real eigenvalues.

In our experiments, we compare classical (BI)CG with CA-(BI)CG, both with and without residual replacement. For CA-(BI)CG, tests were run for $s = [4, 8, 12]$, with the monomial, Newton, and Chebyshev bases (see, e.g., [22]). Coefficients for the Newton and Chebyshev bases were computed using Leja-ordered points obtained from $O(s)$ Ritz value estimates, as described in [4, 14, 22]. We used row and column scaling to equilibrate the input matrix A, preserving symmetry for the SPD case, as described in [14]. For each matrix, we selected a right-hand side b such that

TABLE 1

Matrices used in test cases. Norm and condition numbers reflect the equilibrated system. Note that cdde and xenon1 are nonsymmetric.

Matrix	Domain	N	nnz	cond	2-norm	SPD?
cdde [2, 4]	comp. fluid dynamics	$2.6 \cdot 10^5$	$1.3 \cdot 10^6$	5.5	6.0	N
consph [8]	FEM/Spheres	$8.3 \cdot 10^4$	$6.0 \cdot 10^6$	$9.7 \cdot 10^3$	9.7	Y
thermal1 [8]	thermal	$8.3 \cdot 10^4$	$5.7 \cdot 10^5$	$3.0 \cdot 10^5$	1.9	Y
xenon1 [8]	materials	$4.9 \cdot 10^4$	$1.2 \cdot 10^6$	$3.3 \cdot 10^4$	3.2	N
G2circuit [8]	circuit simulation	$1.5 \cdot 10^5$	$7.3 \cdot 10^5$	$2.3 \cdot 10^5$	2.0	Y

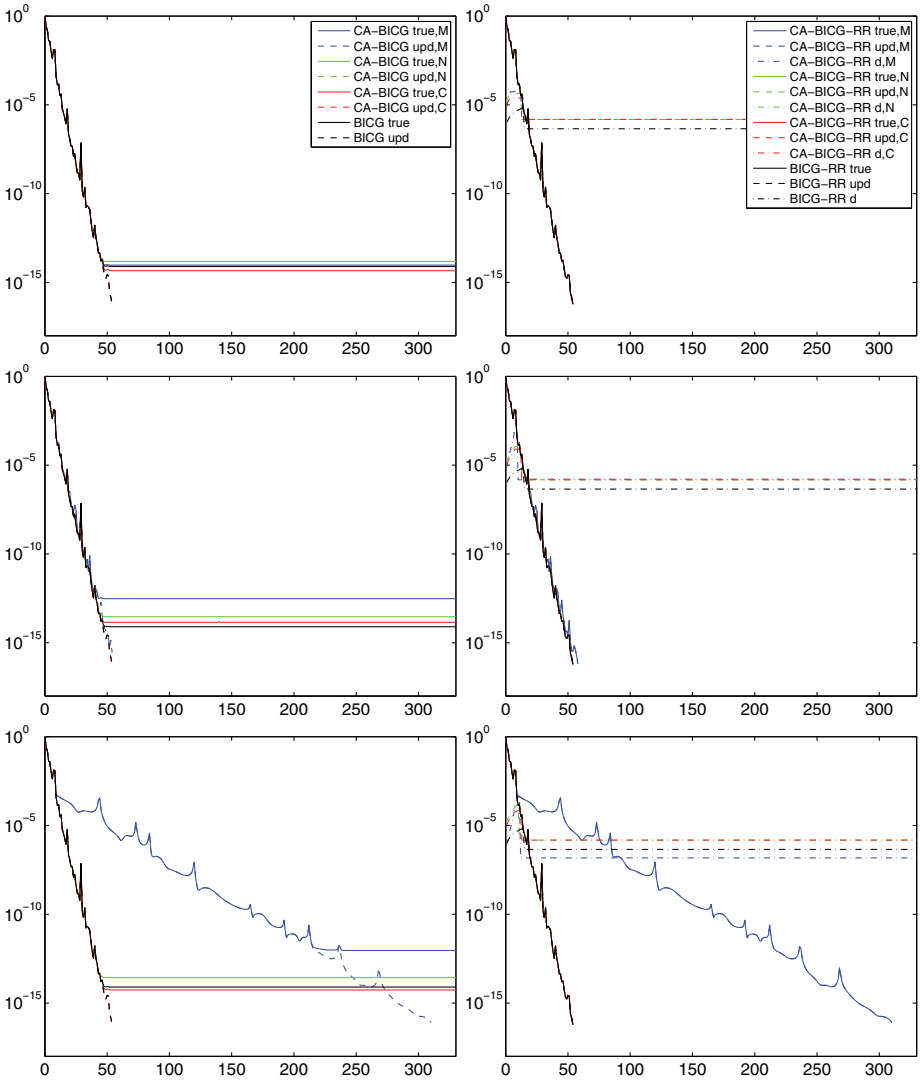

FIG. 1. *cdde convergence.*

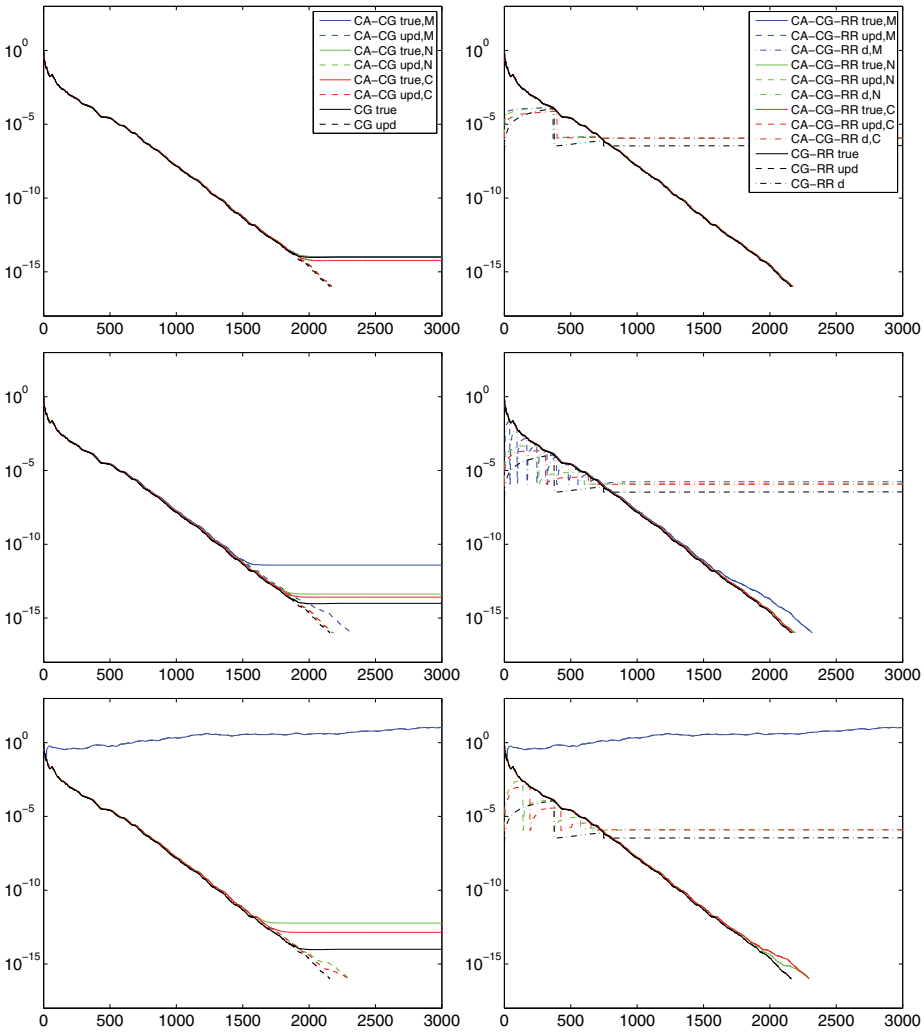

FIG. 2. *consph convergence.*

$\|x\|_2 = 1$, $x_i = 1/\sqrt{N}$, and used the zero vector as the initial guess. All experiments were performed in double precision, i.e., $\epsilon \approx 10^{-16}$. In each test, the convergence criterion used was $\|r_{sk+j+m}\| / \|r_0\| \leq 10^{-16}$. Since $r_0 = b$ and $\|b\| \leq \|A\| \|x\|$, $\|r_{sk+j+m}\| / \|r_0\| \leq 10^{-16}$ implies $\|r_{sk+j+m}\| = O(\epsilon) \|A\| \|x\|$.

Van der Vorst and Ye [28] use $\hat{\epsilon} = 10^{-8} \approx \sqrt{\epsilon}$ and $N_A = 1$ in their experiments (with the assumption that A is very sparse). We use $\hat{\epsilon} = 10^{-8}$ for most experiments; for tests using the monomial basis with $s = 12$, we observed that using $\hat{\epsilon} = 10^{-8}$ caused restarts to occur earlier and more frequently than was desired, as the bound $\eta^{\mathrm{RR}}_{sk+j+m}$ can be quite pessimistic due to the growth of $\left|\underline{\hat{V}}_k\right|$ for large s values. To account for this potential overestimate, we used the value $\hat{\epsilon} = O(10^{-7})$ for all tests with the monomial basis and $s = 12$. Similarly to [28], we set $N' = N_A = N_B = N_V = 1$. While this choice of parameters produces desirable results here, these heuristics deserve a rigorous theoretical analysis; we leave this as future work.

Although our bounds hold regardless of the bases used in constructing $\underline{\hat{V}}_k$, the

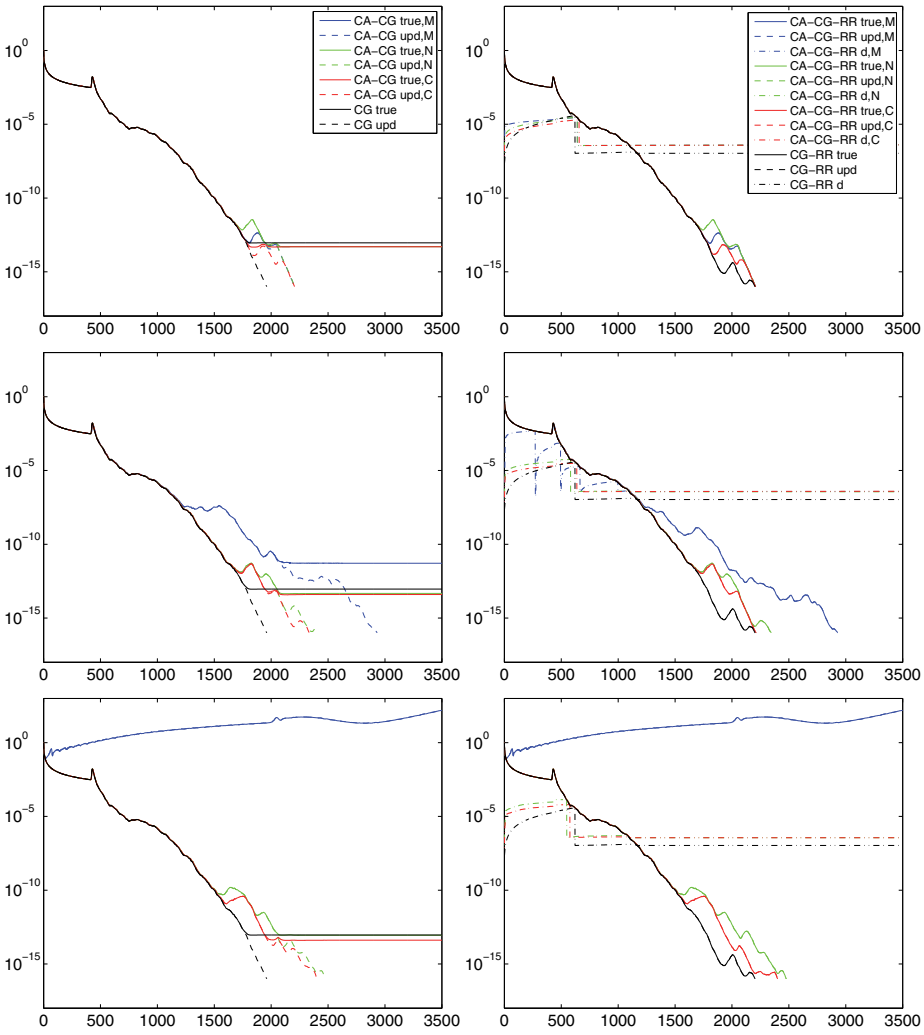

FIG. 3. *thermal*1 *convergence.*

quality of the computed basis does affect the convergence rate of the CA-KSM. In the extreme case of a degenerate basis, the Lanczos process can break down, causing divergence of the updated residual. In such cases, any effort to maintain agreement between the true and updated residual is futile. We, therefore, only test the residual replacement strategy on cases where the updated residual converges.

Figures 1, 2, 3, 4, and 5 show convergence of the true and updated residuals for CA-(BI)CG with and without residual replacement, for each polynomial basis. Plots in the left column show CA-(BI)CG, and plots on the right show CA-(BI)CG with the residual replacement scheme. Plots in the top, middle, and bottom rows show tests where CA-(BI)CG was run with $s = 4, 8,$ and 12, respectively. For comparison, we plot classical (BI)CG and classical (BI)CG with the replacement scheme in [28].

For all plots, the x-axis is iteration number and the y-axis is the 2-norm of the quantities listed in the legend, where "true" is the true residual, "upd" is the updated

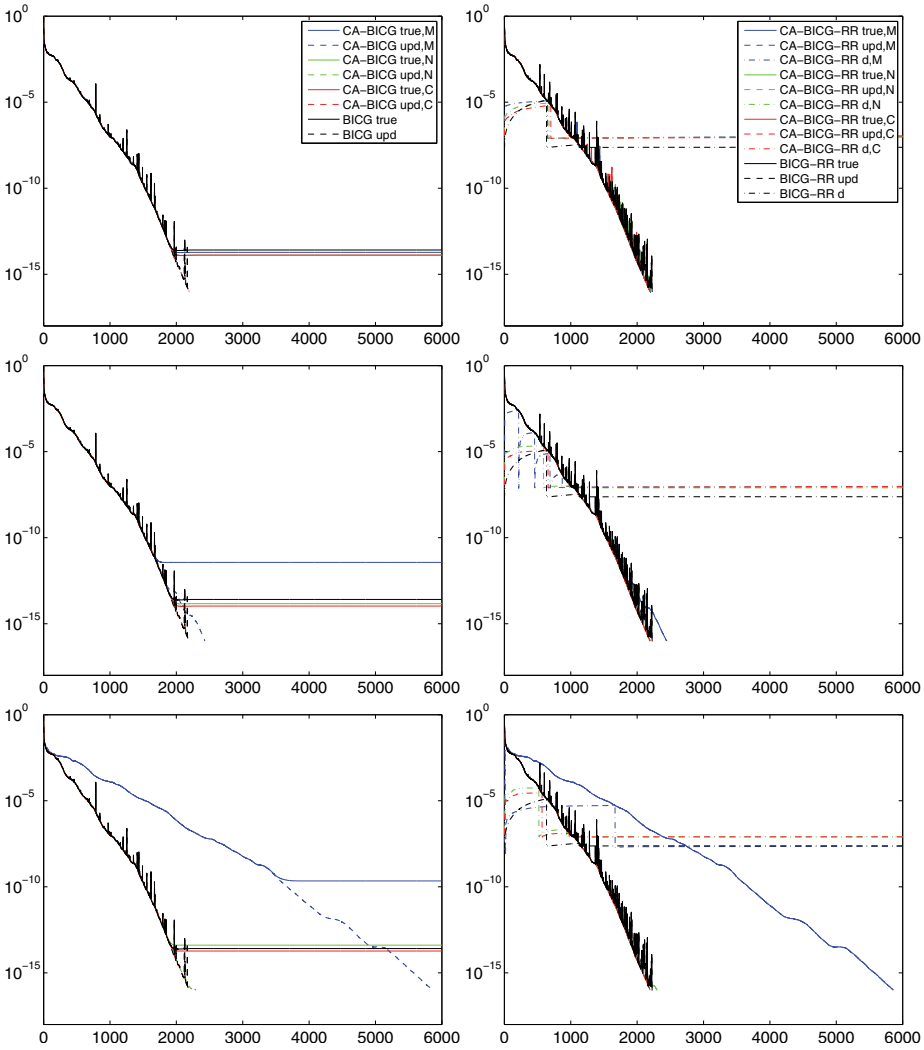

FIG. 4. *xenon1 convergence.*

residual, and "d" is $d_{sk+j+m}/\hat{\epsilon}$, "M", "N", and "C" denote tests with monomial, Newton, and Chebyshev bases, respectively, and "CA-(BI)CG-RR" denotes CA-(BI)CG with residual replacement.

In tests with residual replacement (right columns), we plot the computed value of $d_{sk+j+m}/\hat{\epsilon}$, denoted with dashed-dotted lines, where d_{sk+j+m} is given by (4.13) and (4.14). We can see that residual replacements occur immediately after this quantity grows larger than the residual norm, as per the replacement criterion in (4.15).

Plots without residual replacement (left column) show that, as our bounds indicate, the attainable accuracy in CA-(BI)CG generally grows worse with increasing s, especially using bases where $\left|\underline{\hat{V}}_k\right|$ grows quickly with s. It is worth mentioning that without residual replacement, a well-conditioned polynomial basis can allow the CA-KSM to achieve slightly better accuracy than the classical method (see, e.g., the Newton and Chebyshev bases in Figures 3 and 5).

The improvement in maximum attainable accuracy due to the replacement scheme

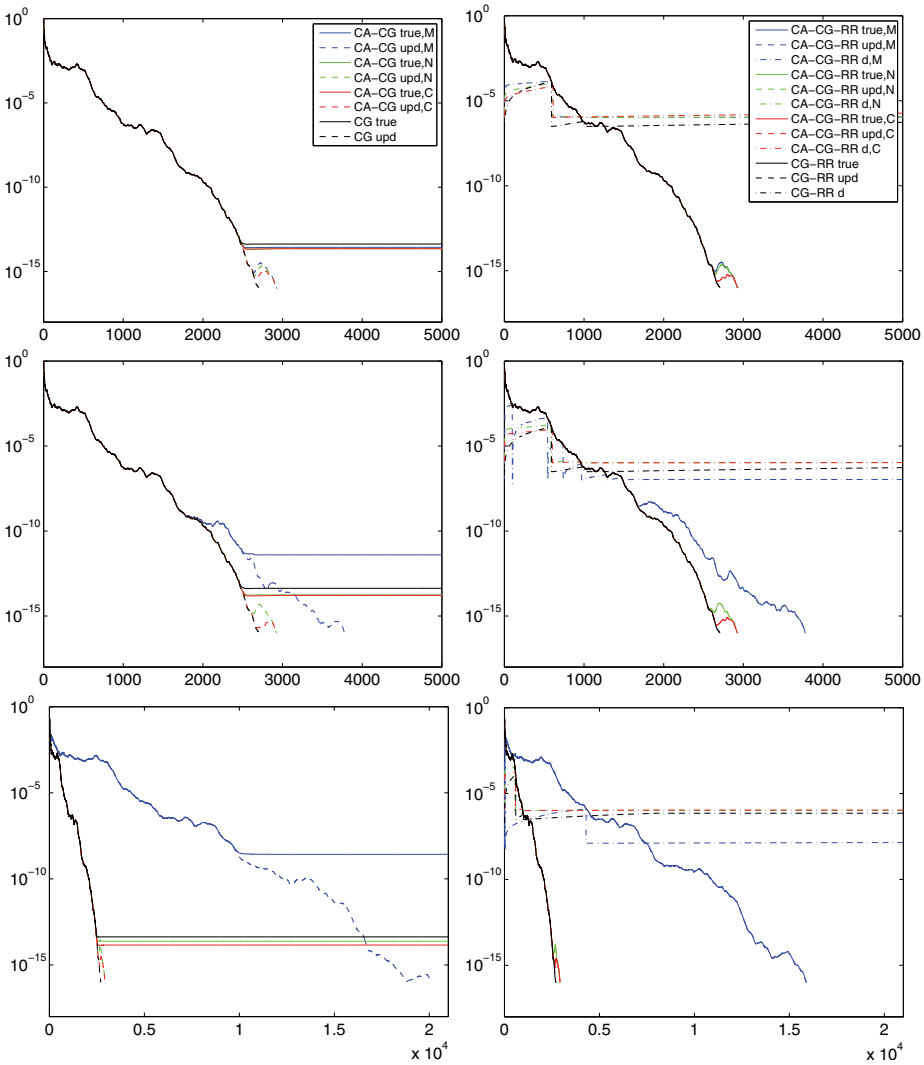

FIG. 5. *G2circuit convergence. Note that the x-axis scale differs for $s = 12$ (bottom row).*

for each test is summarized in Table 2, which gives the 2-norm of the true residual (normalized by $\|r_0\|$ as described above) for CA-(BI)CG without and with residual replacement. Our results show that, for cases where the updated residual converges to $O(\epsilon)\,\|A\|\,\|x\|$, the residual replacement scheme for CA-(BI)CG improves convergence of the true residual to within $O(\epsilon)\,\|A\|\,\|x\|$. As shown in Table 2, this improvement can be as large as seven orders of magnitude.

In all cases, the updated residuals in the algorithms with replacement converge at approximately the same rate in the algorithms without replacement. We note that for tests with the monomial basis and $s = 12$, this behavior is sensitive to the chosen $\hat{\epsilon}$ value. Choosing $\hat{\epsilon}$ too low can result in replacing too frequently, as described above, which can cause slower convergence or divergence. In Figure 4, we can see that, although the overall rate of convergence is unaffected, residual replacement does result in more irregular convergence for both CA-BICG and classical BICG.

TABLE 2

Improvement in true residual 2-norm. For each test, the left column gives the 2-norm of the true residual without replacement (no RR), and the right gives the 2-norm of the true residual with replacement (RR). Dashes (-) indicate tests where the updated residual did not converge.

		cdde		consph		thermal1		xenon1		G2circuit	
		no RR	RR	no RR	RR	no RR	RR	no RR	RR	no RR	RR
Classical		8e-15	6e-17	1e-14	1e-16	9e-14	1e-16	3e-14	9e-17	4e-14	1e-16
Monomial	4	1e-14	6e-17	1e-14	1e-16	5e-14	1e-16	2e-14	8e-17	3e-14	1e-16
	8	3e-13	7e-17	4e-12	1e-16	5e-12	1e-16	4e-12	1e-16	4e-12	1e-16
	12	9e-13	8e-17	–	–	–	–	2e-10	1e-16	3e-9	1e-16
Newton	4	2e-14	6e-17	1e-14	1e-16	5e-14	1e-16	1e-14	1e-16	2e-14	1e-16
	8	3e-14	6e-17	4e-14	1e-16	5e-14	1e-16	1e-14	1e-16	2e-14	1e-16
	12	3e-14	6e-17	6e-13	1e-16	9e-14	1e-16	4e-14	1e-16	2e-14	1e-16
Chebyshev	4	5e-15	1e-16	6e-15	1e-16	5e-14	1e-16	1e-14	1e-16	2e-14	1e-16
	8	1e-14	6e-17	3e-14	1e-16	4e-14	1e-16	1e-14	1e-16	2e-14	1e-16
	12	6e-15	6e-17	1e-13	1e-16	4e-14	1e-16	2e-14	1e-16	1e-14	1e-16

TABLE 3

Percentage residual replacement steps. For each test, the first number is the percentage of total iterations that were replacement steps, calculated by the fraction in parentheses, in which the numerator gives the number of replacement steps and the denominator gives total iterations. Dashes (-) indicate tests where the updated residual did not converge.

		cdde	consph	thermal1	xenon1	G2circuit
Classical		1.9 (1/54)	.09 (2/2164)	.09 (2/2111)	.09 (2/2226)	.07 (2/2707)
Monomial	4	1.9 (1/54)	.09 (2/2173)	.05 (1/2206)	.05 (1/2202)	.09 (2/2930)
	8	1.7 (1/58)	.52 (12/2319)	.17 (5/2929)	.20 (5/2446)	.12 (6/3779)
	12	.32 (1/310)	–	–	.03 (2/5859)	.02 (3/15901)
Newton	4	1.9 (1/54)	.09 (2/2176)	.04 (1/2226)	.05 (1/2194)	.07 (2/2930)
	8	1.9 (1/54)	.09 (2/2192)	.09 (2/2345)	.09 (2/2196)	.07 (2/2928)
	12	1.9 (1/54)	.17 (4/2296)	.08 (2/2477)	.09 (2/2299)	.07 (2/2930)
Chebyshev	4	1.9 (1/54)	.09 (2/2169)	.05 (1/2206)	.05 (1/2194)	.03 (1/2937)
	8	1.9 (1/54)	.09 (2/2179)	.05 (1/2215)	.05 (1/2194)	.07 (2/2928)
	12	1.9 (1/54)	.13 (3/2294)	.08 (2/2401)	.09 (2/2195)	.07 (2/2930)

The percentage of total iterations that were residual replacement iterations for each test, shown in Table 3 did not exceed 2% in any of our experiments. The highest number of replacements in any test was 12 (for monomial with $s = 8$ in Figure 2). Since the number of residual replacement steps is very small relative to the total number of iterations, the communication cost of performing residual replacements (adding an additional outer loop iteration) is negligible in the context of CA-(BI)CG.

6. Future work and conclusions. In this work, we provide the first analysis of maximum attainable accuracy in finite precision s-step KSMs, namely CA-CG and CA-BICG, by deriving a bound on the norm of the deviation of the true and updated residuals. An estimate for this bound can be updated iteratively within CA-CG and CA-BICG for minimal additional cost. Following related work in [28] for classical CG, we use this bound in devising an implicit residual replacement and group update strategy for improving agreement between the true and updated residuals in CA-CG and CA-BICG. Numerical experiments on real-world problems demonstrate that this strategy can improve solution accuracy to within a factor of $O(\epsilon)\|A\|\|x\|$ with a small number of residual replacement steps.

Much work remains to be done on the analysis of finite precision s-step KSMs. We plan to extend the analysis and scheme developed here to other s-step KSMs, such as CA-BICGSTAB [4], as well as test the performance impact of our residual replace-

ment scheme in high-performance parallel implementations. The development of a convergence theory for s-step KSMs in finite precision, analogous to those developed for classical KSMs, remains an open problem.

Acknowledgments. The authors would like to thank the anonymous reviewers and the associate editor, whose suggestions and constructive feedback led to significant improvements in both the analysis and presentation of this paper.

REFERENCES

[1] Z. Bai, *Error analysis of the Lanczos algorithm for the nonsymmetric eigenvalue problem*, Math. Comp., 62 (1994), pp. 209–226.
[2] Z. Bai, D. Day, J. Demmel, and J. Dongarra, *A test matrix collection for non-hermitian eigenvalue problems*, Technical Report CS-97-355, Department of CS, University of Tennessee, Knoxville, TN, 1997.
[3] Z. Bai, D. Hu, and L. Reichel, *A Newton basis GMRES implementation*, IMA J. Numer. Anal., 14 (1994), pp. 563–581.
[4] E. Carson, N. Knight, and J. Demmel, *Avoiding communication in nonsymmetric Lanczos-based Krylov subspace methods*, SIAM J. Sci. Comput., 35 (2013), pp. S542–S561.
[5] A. T. Chronopoulos and C. W. Gear, *On the efficient implementation of preconditioned s-step conjugate gradient methods on multiprocessors with memory hierarchy*, Parallel Comput., 11 (1989), pp. 37–53.
[6] A. T. Chronopoulos and C. W. Gear, *s-step iterative methods for symmetric linear systems*, J. Comput. Appl. Math, 25 (1989), pp. 153–168.
[7] A. T. Chronopoulos and C. D. Swanson, *Parallel iterative s-step methods for unsymmetric linear systems*, Parallel Comput., 22 (1996), pp. 623–641.
[8] T. Davis and Y. Hu, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), 1.
[9] J. Demmel, M. Hoemmen, M. Mohiyuddin, and K. Yelick, *Avoiding Communication in Computing Krylov Subspaces*, Technical Report UCB/EECS-2007-123, EECS Department, U.C. Berkeley, Berkeley, CA, 2007.
[10] D. Gannon and J. Van Rosendale, *On the impact of communication complexity on the design of parallel numerical algorithms*, Trans. Comput., 100 (1984), pp. 1180–1194.
[11] A. Greenbaum, *Estimating the attainable accuracy of recursively computed residual methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 535–551.
[12] M. H. Gutknecht and Z. Strakoš, *Accuracy of two three-term and three two-term recurrences for Krylov space solvers*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 213–229.
[13] A. Hindmarsh and H. Walker, *Note on a Householder Implementation of the GMRES Method*, Technical Report UCID-20899, Lawrence Livermore National Laboratory, Livermore, CA, 1986.
[14] M. Hoemmen, *Communication-avoiding Krylov Subspace Methods*, Ph.D. thesis, EECS Department, U.C. Berkeley, Berkeley, CA, 2010.
[15] W. Joubert and G. Carey, *Parallelizable restarted iterative methods for nonsymmetric linear systems. Part I: Theory*, Int. J. Comput. Math., 44 (1992), pp. 243–267.
[16] C. E. Leiserson, S. Rao, and S. Toledo, *Efficient out-of-core algorithms for linear relaxation using blocking covers*, J. Comput. System Sci., 54 (1997), pp. 332–344.
[17] G. Meurant, *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*, SIAM, Philadelphia, 2006.
[18] G. Meurant and Z. Strakoš, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numer., 15 (2006), pp. 471–542.
[19] M. Mohiyuddin, M. Hoemmen, J. Demmel, and K. Yelick, *Minimizing communication in sparse matrix solvers*, in Proc. ACM/IEEE Conference on Supercomputing, 2009.
[20] A. Neumaier, *Iterative regularization for large-scale ill-conditioned linear systems*, Oral presentation, Numerical Linear Algebra Meeting, Oberwolfach, 1994.
[21] C. Paige, M. Rozložnik, and Z. Strakoš, *Modified Gram-Schmidt (MGS), least squares, and backward stability of MGS-GMRES*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 264–284.
[22] B. Philippe and L. Reichel, *On the generation of Krylov subspace bases*, Appl. Numer. Math., 62 (2012), pp. 1171–1186.
[23] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.
[24] G. Sleijpen and H. van der Vorst, *Reliable updated residuals in hybrid Bi-CG methods*, Computing, 56 (1996), pp. 141–163.

[25] E. Sturler, *A performance model for Krylov subspace methods on mesh-based parallel computers*, Parallel Comput., 22 (1996), pp. 57–74.
[26] S. Toledo, *Quantitative Performance Modeling of Scientific Computations and Creating Locality in Numerical Algorithms*, Ph.D. thesis, MIT, Cambridge, MA, 1995.
[27] C. H. Tong and Q. Ye, *Analysis of the finite precision bi-conjugate gradient algorithm for nonsymmetric linear systems*, Math. Comp., 69 (2000), pp. 1559–1576.
[28] H. Van der Vorst and Q. Ye, *Residual replacement strategies for Krylov subspace iterative methods for the convergence of true residuals*, SIAM J. Sci. Comput., 22 (2000), pp. 835–852.
[29] J. Van Rosendale, *Minimizing Inner Product Data Dependencies in Conjugate Gradient Iteration*, Technical Report 172178, ICASE-NASA, Hampton, VA, 1983.
[30] H. F. Walker, *Implementation of the GMRES method using Householder transformations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 152–163.