Residual Replacement Strategies for Krylov Subspace Iterative Methods for the Convergence of True Residuals

Henk A. van der Vorst *   Qiang Ye †

March 1, 1999

Abstract

In this paper, a strategy is proposed for alternative computations of the residual vectors in Krylov subspace methods, which improves the agreement of the computed residuals and the true residuals to the level of $O(\mathbf{u})||A||||x||$. Building on earlier ideas on residual replacement and on insights in the finite precision behaviour of the Krylov subspace methods, computable error bounds are derived for iterations that involve occasionally replacing the computed residuals by the true residuals, and they are used to monitor the deviation of the two residuals and hence to select residual replacement steps, so that the recurrence relations for the computed residuals, which control the convergence of the method, are perturbed within safe bounds. Numerical examples are presented to demonstrate the effectiveness of this new residual replacement scheme.

1 Introduction

Krylov subspace iterative methods for solving a large linear system $Ax = b$ typically consist of iterations that recursively update approximate solutions x_n and the corresponding residual vectors r_n $(= b - Ax_n)$. They can be written in a general form as follows.

Algorithm 1. Template for Krylov subspace Method:
Input: an initial approximation x_0; $r_0 = b - Ax_0$;
For $n = 1, 2, \cdots$ until convergence
    Generate a correction vector q_n by the method;
    $x_n = x_{n-1} + q_n$
    (the vector x_n does not occur in other statements)
    $r_n = r_{n-1} - Aq_n$
End for

*Department of Mathematics, Utrecht University, P.O. Box 80010, NL-3508 TA Utrecht, The Netherlands E-mail: vorst@math.uu.nl

†Department of Mathematics, University of Manitoba, Winnipeg, Manitoba, Canada R3T 2N2. E-mail: ye@gauss.amath.umanitoba.ca Research supported by grants from University of Manitoba Research Development Fund and from Natural Sciences and Engineering Research Council of Canada

Most Krylov subspace iterative methods, including the conjugate gradient method (CG) [12], the bi-conjugate gradient method (Bi-CG) [4, 13], CGS [19], and BiCGSTAB [22], fit in this framework (see [2, 11, 16] for other methods).

In exact arithmetic, the recursively defined r_n in Algorithm 1 is exactly the residual for the approximate solution x_n, because $b - Ax_n - r_n = b - Ax_{n-1} - r_{n-1} = b - Ax_0 - r_0 = 0$. In a floating point arithmetic, however, the round-off patterns for x_n and r_n will be different. It is important to note that any error made in the computation of x_n is not reflected by a corresponding error in r_n, or in other words, computational errors to x_n do not force the method to correct, since x_n has no influence on the iteration process. This leads to the well known situation that $b - Ax_n$ and r_n may differ significantly. This phenomenon has been extensively discussed in the literature, see [10, 11, 18] and the references cited there. Indeed, if we denote the computed results of x_n, r_n by $\hat{x}_n, \hat{r}_n$, respectively (but we still use q_n to denote the computed update vector of the algorithm), then we have

$$\hat{x}_n = fl(\hat{x}_{n-1} + q_n) = \hat{x}_{n-1} + q_n + \psi_n, \quad |\psi_n| \leq \mathbf{u}|\hat{x}_n| + O(\mathbf{u}^2) \tag{1}$$

$$\hat{r}_n = fl(\hat{r}_{n-1} - Aq_n) = \hat{r}_{n-1} - Aq_n + \eta_n, \quad |\eta_n| \leq \mathbf{u}(|\hat{r}_n| + N|A||q_n|) + O(\mathbf{u}^2), \tag{2}$$

where $fl(z)$ denotes the computed result of z in finite arithmetic, the absolute value and inequalities on vectors are componentwise, and $\mathbf{u}$ is the machine roundoff unit. The vectors ψ_n and η_n represent rounding error terms, and they can be bounded by a straightforward error analysis (see Section 3 for details). In particular, the relations (1) and (2) show that ψ_n and η_n depend only on the iteration vectors $\hat{x}_n$, $\hat{r}_n$, and q_n.

We will call $b - A\hat{x}_n$ the *true residual* for the approximation $\hat{x}_n$ and call $\hat{r}_n$, as obtained by recurrence formula (2), the *computed residual* (or the *updated residual*). Then the difference between the two satisfies (using the finite precision recurrences (1) and (2))

$$\begin{aligned} b - A\hat{x}_n - \hat{r}_n &= b - A\hat{x}_{n-1} - \hat{r}_{n-1} - A\psi_n - \eta_n \\ &= -\Sigma_{i=1}^n (A\psi_i + \eta_i), \end{aligned}$$

where we assume for now that $b - Ax_0 - r_0 = 0$. Hence, the difference between the true and the updated residuals is a result of accumulated rounding errors. In particular, a significant deviation of $b - A\hat{x}_n$ from $\hat{r}_n$ may be expected, if there is a $\hat{x}_i$ or $\hat{r}_i$ with large norm during the iteration (a not uncommon situation for Bi-CG and CGS). On the other hand, even when all ψ_i or η_i are small (as is common for CG), but if it takes a relatively large number of iterations for convergence, the sheer accumulation of ψ_i and η_i could also lead to a nontrivial deviation.

What makes all this so important is that, in a finite precision implementation, the sequence $\hat{r}_n$ satisfies, almost to machine precision $\mathbf{u}$, its defining recurrence relation, and as was observed for many Krylov subspace methods, this is the driving force behind convergence of $\hat{r}_n$ [10, 15, 18, 20]. Indeed, residual bounds have been obtained in [20] for CG and Bi-CG, which show that even a significantly perturbed recurrence relation (with perturbations much larger than the machine precision) usually still leads to eventual convergence of the computed residuals. This theoretical insight has been our motivation and justification for the residual replacement scheme to be presented in Section 2.1. On the other hand, the true residual $b - A\hat{x}_n$ itself has no self-correcting mechanism for convergence, mainly because any perturbation made to $\hat{x}_n$ does not have an effect on the iteration parameters, whereas errors in $\hat{r}_n$ immediately lead to other iteration parameters.

Thus, in a typical convergent iteration process, $\hat{r}_n$ converges to a level much smaller than $\mathbf{u}$ eventually, but the true residual $b - A\hat{x}_n$ can only converge to the level dictated by $\Sigma_{i=1}^{n}(A\psi_i + \eta_i)$, since

$$b - A\hat{x}_n = \hat{r}_n - \Sigma_{i=1}^{n}(A\psi_i + \eta_i).$$

Usually, when $\hat{r}_n$ is still bigger than the accumulated error $\Sigma_{i=1}^{n}(A\psi_i + \eta_i)$, $b - A\hat{x}_n$ agrees well with $\hat{r}_n$ in magnitude, but when $\hat{r}_n$ has converged to a level that is smaller than the accumulated error, then $b - A\hat{x}_n \sim \Sigma_{i=1}^{n}(A\psi_i + \eta_i)$ is just the accumulated error and has no agreement at all with $\hat{r}_n$. In summary, a straightforward implementation would reduce the true residuals at best to $\Sigma_{i=1}^{n}(A\psi_i + \eta_i)$. A bound for this has been obtained in [10] and it is called the *attainable accuracy*. We note that this term could be significant even if only one of ψ_i or η_i is large, or if n is large.

The above problems become most serious in methods such as CGS and Bi-CG where intermediate $\hat{x}_n$ and $\hat{r}_n$ can have very large norm, and this may result in a large ψ_n or η_n. Several popular methods, such as BiCGSTAB [22], BiCGSTAB(ℓ) [17], QMR [7], TFQMR [5], and composite step BiCG [1], have been developed to reduce the norm of $\hat{r}_n$ (see [6] for details). We note that controlling the size of $\|\hat{r}_n\|$ only does not solve the deviation problem in all situations, as, for instance, the accumulation of tiny errors over a long iteration may still result in a nontrivial deviation.

A simple approach for solving the deviation problem is to replace the computed residuals by the true residuals at some iteration step to restore the agreement. Then the deviation at subsequent steps will be the error accumulation after that iteration only. This includes a complete replacement strategy that simply computes r_n by $b - Ax_n$ at every iteration, and a periodic replacement strategy that updates r_n by $b - Ax_n$ only at intervals of the iteration count. While such a strategy maintains agreement of the two kinds of residuals, it turns out that the convergence of the r_n may deteriorate (as we will see, it may result in unacceptably large perturbations to the lanczos recurrence relation for the residual vectors that steers the convergence, see Section 2.3). Recently, Sleijpen and van der Vorst [18], motivated by suggestions made by Neumaier (see [11, 18]), introduced a very sophisticated replacement scheme that includes a so-called *flying-restart* procedure. It was demonstrated that this new residual replacement strategy can be very effective in the sense that it can improve the convergence of the true residuals by several orders of magnitude. For practical implementations, such a strategy is very useful because it leads to meaningful residuals and this is important for stopping the iteration process at the right point. Of course, one could, after termination of the classical process, simply test the true residual, but the risk is that the true residual stagnated already long before termination, so that much work has been done in vain.

The present paper will follow the very same idea of replacing the computed residual by the true residual at selected steps, in order to maintain close agreement between the two residuals, but we propose a simpler strategy so that the replacement is done only when it is necessary and at phases in the iteration where it is harmless, that is that convergence mechanism for $\hat{r}_n$ is not destroyed. Specifically, we shall present a rigorous error analysis for iterations with residual replacement and we will propose computable bounds for the deviation between the computed and true residuals. This will be used to select the replacement phases in the iteration in such a way that the Lanczos recurrence among $\hat{r}_n$ is sufficiently well maintained. For the resulting strategy, it will be shown that, provided that the computed residuals converge, the true residual will converge to the level $O(\mathbf{u})\|A\|\|x\|$, the smallest level that one can expect for an approximation.

The paper has been organized as follows. In Section 2, we develop a refined residual replacement strategy and we discuss some strategies that have been reported by others. We give an error analysis in Section 3, and we derive some bounds for the deviation to be used in the replacement condition.

We present a complete implementation in Section 4. It turns out that the residual replacement strategy can easily be incorporated in existing codes. Some numerical examples are reported in Section 5, and we finish with remarks in Section 6.

The vector norm used in this paper is one of the 1, 2, or ∞-norm.

2 Residual Replacement Strategy

In this section, we develop a replacement strategy that maintains the convergence of the true residuals. A formal analysis is postponed to the next section. The specific iterative method can be any of those that fit in the general framework of Algorithm 1. Throughout this paper, we shall consider only iteration processes for which the computed residual $\hat{r}_n$ converges to a sufficiently small level.

As mentioned in Section 1, we follow the basic idea to replace the computed residual $\hat{r}_m$ by the true residual $fl(b - A\hat{x}_m)$ at some selected steps $m = m_1, m_2, \cdots, m_k$. We will refer to such an iteration step as one where *residual replacement* occurs. Hence, the residual generated at an arbitrary step n could be either the usual updated residual $\hat{r}_n = fl(\hat{r}_{n-1} - Aq_{n-1})$ or the true residual $fl(b - A\hat{x}_n)$, depending on whether replacement has taken place or not at step n. In order to distinguish the two possible formulations, we denote by r_n the residual obtained at step n of the process with the replacement strategy, that is

$$r_n = \begin{cases} fl(b - A\hat{x}_n), & \text{if } n = m_1, m_2, \cdots, m_k \\ \hat{r}_n = fl(r_{n-1} - Aq_{n-1}), & \text{otherwise} \end{cases}$$

With the residual replacement at step m ($m = m_1, m_2, \cdots, m_k$), the residual deviation is immediately reduced to

$$\delta_m \equiv b - A\hat{x}_m - r_m = b - A\hat{x}_m - fl(b - A\hat{x}_m) = -\xi_m,$$

and it can be shown (see Lemma 1 of Section 2.2) that $|\xi_m| \leq \mathbf{u}(|r_m| + N|A||\hat{x}_m|) + O(\mathbf{u}^2)$. For the subsequent iterations $n > m$, but before the next replacement step, we clearly have that

$$\begin{aligned} \delta_n &= b - A\hat{x}_n - r_n = b - A\hat{x}_m - r_m - \Sigma_{i=m+1}^{n}(A\psi_i + \eta_i) \\ &= -\xi_m - \Sigma_{i=m+1}^{n}(A\psi_i + \eta_i). \end{aligned} \tag{3}$$

Therefore, the accumulated deviation before step m has no effect to the deviation after updating ($n > m$). However, in order for such a strategy to succeed, two conditions must be met, namely,

- the computed residual r_n should preserve the convergence mechanism of the original process that has been steered by the $\hat{r}_n$ vectors;
- from the last updating step m to the termination step K, the accumulated error $\Sigma_{i=m+1}^{K}(A\psi_i + \eta_i)$ should be small relative to $\mathbf{u}(|r_m| + N|A||\hat{x}_m|)$, which is the upperbound for $|\xi_m|$.

We discuss in the next two subsections how to satisfy these two objectives.

2.1 Maintaining convergence of computed residuals

In order that r_n maintains the convergence mechanism of the original updated residuals, it should preserve the property that gives rise to the convergence of the original $\hat{r}_n$. We therefore need to identify the properties that lead to convergence of the iterative method *in finite precision arithmetic*. While this may be different for each individual method, it has been observed for several Krylov subspace methods (including CG [10, 20], Bi-CG [20], CGS, BiCGSTAB, and BiCGSTAB(ℓ) [18]), that the recurrence $r_n = r_{n-1} - Aq_n$ and a similar one for q_n is satisfied almost to machine precision and this small local error is one of the properties behind the convergence of the computed residuals. Furthermore, the analysis of [20] suggests that convergence is well maintained even when the recurrence equations are perturbed with perturbations that are significantly greater than the machine precision. This latter property is the basis for our residual replacement strategy. Therefore, we briefly discuss this perturbation phenomenon for Bi-CG (or CG), as presented in [20].

Consider the Bi-CG iteration which contains $r_n = r_{n-1} - \alpha_{n-1}Ap_{n-1}$ and $p_n = r_n + \beta_n p_{n-1}$. In finite arithmetic, $\hat{r}_n$ and $\hat{p}_n$, which denote the computed results of r_n and p_n, respectively, satisfy the perturbed recurrence

$$\hat{r}_n = \hat{r}_{n-1} - \alpha_{n-1}A\hat{p}_{n-1} + \eta_n \quad \text{and} \quad \hat{p}_n = \hat{r}_n + \beta_n\hat{p}_{n-1} + \tau_n,$$

where η_n and τ_n are rounding error terms that can be bounded in terms of $\mathbf{u}$. Combining these two equations, we obtain the following perturbed matrix equation in a normalized form

$$AZ_n = Z_nT_n - \frac{1}{\alpha'_n}\frac{\hat{r}_{n+1}}{\|\hat{r}_1\|}e_n^T + F_n \quad \text{with} \quad Z_n = [\frac{\hat{r}_1}{\|\hat{r}_1\|}, \cdots, \frac{\hat{r}_n}{\|\hat{r}_n\|}], \tag{4}$$

where T_n is an invertible tridiagonal matrix[1], $\alpha'_n = \|\hat{r}_n\|\alpha_n/\|r_1\| = e_n^T T_n^{-1}e_1$ and $F_n = [f_1, \cdots, f_n]$ with

$$f_n = \frac{A\tau_n}{\|\hat{r}_n\|} + \frac{1}{\alpha_n}\frac{\eta_{n+1}}{\|\hat{r}_n\|} - \frac{\beta_n}{\alpha_{n-1}}\frac{\eta_n}{\|\hat{r}_n\|}. \tag{5}$$

We note that (4) is just an equation satisfied by an exact Bi-CG iteration under a perturbation F_n. In particular, detailed bounds on τ_n and η_n will, under some mild assumptions, lead to $F_n \sim O(\mathbf{u})$.

The main result of [20] states that if a sequence $\hat{r}_n$ satisfies (4) and Z_{n+1} has full rank, then we have

$$\|\hat{r}_{n+1}\| \leq (1 + K_n) \min_{p\in\mathcal{P}_n, p(0)=1} \|p(A + \Delta A_n)\hat{r}_1\|, \tag{6}$$

where $K_n = \|(AZ_n - F_n)T_n^{-1}\|\,\|Z_{n+1}^+\|$ and $\Delta A_n = -F_nZ_n^+$. The case $F_n = 0$ reduces to the known theoretical bound for the exact BiCG residuals [1]. Therefore, even when $\hat{r}_n$ and its exact counterpart are completely different, their norms are bounded by similar quantities and are usually comparable. Of course, in both cases, the bounds depend on the quality of the constructed basis. More importantly, a closer examination of the bound reveals that even if the perturbation F_n is in magnitude much larger than $\mathbf{u}$, the quantities in the bound, and thus $\|\hat{r}_{n+1}\|$, may not be significantly affected. Indeed, in [20] numerical experiments were presented, where relatively large artificial random perturbations had been injected to the recurrence for r_n; yet it did not significantly affect the convergence mechanism.

An implication of this analysis is that, regardless of how $\hat{r}_n$ is generated but as long as it satisfies (4), its norm can be bounded by (6). Hence, we can replace $\hat{r}_n$ by $r_n = fl(b - Ax_n)$

[1] We assume that no breakdowns of the iteration process have occurred

when $\eta_n = r_n - (r_{n-1} - Aq_n)$ are not too large relative to $\|r_n\|$ and $\|r_{n-1}\|$ (see (5)), and we may still expect it to converge in a similar fashion. Indeed, this criterion explains why the residual replacement strategies like $r_n = fl(b - Ax_n)$ work sometimes, but do not work always (see Section 2.3). Here, it will be used to determine when it is safe to replace $\hat{r}_n$ by $r_n = fl(b - Ax_n)$. We note that the above discussion is for Bi-CG, but the phenomenon it reveals seems to be valid for many other methods, especially for those methods that are based on Bi-CG (CGS, BiCGSTAB, and others).

Now we consider the case that residual replacement is carried out at step m, that is $r_m = fl(b - A\hat{x}_m) = b - A\hat{x}_m + \xi_m$. It follows from the definition of δ_m and $\hat{r}_m$ that $b - A\hat{x}_m = \hat{r}_m + \delta_m = r_{m-1} - Aq_{m-1} + \eta_m + \delta_m$. So, the updated residual r_m satisfies

$$r_m = r_{m-1} - Aq_{m-1} + \eta'_m \quad \text{with} \quad \eta'_m = \eta_m + \delta_m + \xi_m. \tag{7}$$

Thus depending on the magnitude of $\|\eta'_m\|$ relative to $\|r_m\|$ and $\|r_{m-1}\|$, the use of $r_m = fl(b - A\hat{x}_m)$ may result in large perturbations to the recurrence relation. Therefore, a residual replacement strategy should ensure that the replacement is only done when $\|\eta'_m\| / \min\{\|r_m\|, \|r_{m-1}\|\}$ is not too large.

In a typical iteration, as the iteration proceeds, $\|\delta_n\|$, and hence $\|\eta'_n\|$, increases while $\|\hat{r}_n\|$ decreases. Replacement will reduce δ_n but, in order to maintain the recurrence relation, it should be carried out before $\|\eta'_n\|$ becomes too large relative to $\|\hat{r}_n\|$. For this reason, we propose to set a threshold ϵ and carry out a replacement when $\|\eta'_n\| / \|\hat{r}_n\|$ reaches the threshold. To be precise, we replace the residual at step n by $r_n = b - Ax_n$, if

$$\|\eta'_{n-1}\| \leq \epsilon \|\hat{r}_{n-1}\| \quad \text{and} \quad \|\eta'_n\| > \epsilon \|\hat{r}_n\|. \tag{8}$$

We note that, in principle, residual replacement can be carried out for all steps up to where $\|\eta'_n\|$ reaches certain point. However, from the stability point of view, it is preferred to generate the residual by the recurrence as much as possible, since $\|\eta'_n\|$ is generally bigger than the recurrence rounding error $\|\eta_n\|$ (of order $\mathbf{u}$).

2.2 Groupwise solution updating to reduce error accumulations

From the discussions of Section 2.1, we learn that residual replacement should only be carried out up to certain point. In this subsection, we will discuss how to maintain, after the last replacement, the deviation at the order of $\mathbf{u}|A||x_n|$, in which case x_n is a backward stable solution. Note that, for any x_n, $\mathbf{u}\|A\|\|x_n\|$ is the lowest value one can expect for its residual. This is simply because even with the exact solution x, both $b - A(fl(x))$ and $fl(b - Ax) \sim \mathbf{u}|A||x|$.

If $m = m_k$ is the last updating step, which menas that we are in the final phase of the iteration process, then, because of (3), the deviation at step $n > m$ is

$$\delta_n = -\xi_m - \Sigma_{i=m+1}^n \eta_i - \Sigma_{i=m+1}^n A\psi_i.$$

From our updating condition, we have that $\|r_n\| \leq \|\eta'_n\|/\epsilon \sim O(\frac{\mathbf{u}}{\epsilon})$. So, if ϵ is chosen not too close to $\mathbf{u}$, $\|r_n\|$ is small and $\hat{x}_n \sim x$ for $n \geq m$. We now discuss the three different parts of δ_n. The discussion here is only to motivate the groupwise updating strategy; a more rigorous analysis will be given in the next section.

- $|\xi_m| \leq \mathbf{u}(|r_m| + N|A||\hat{x}_m|) \sim O(\mathbf{u})|A||x|$.

- Since $|\hat{r}_i| << |b| \le |A||x|$ and $|\eta_i| \sim \mathbf{u}|\hat{r}_i|$, we have that $\Sigma_{i=m+1}^{n}|\eta_i| << O(\mathbf{u})|A||x|$.
- For the ψ_i part, $|\psi_i| \sim \mathbf{u}|\hat{x}_i| \sim \mathbf{u}|x|$. Hence, $\Sigma_{i=m+1}^{n}|A\psi_i| \sim \Sigma_{i=m+1}^{n}\mathbf{u}\|A\|\|x\| = (n-m)\mathbf{u}\|A\|\|x\|$. If $n-m$ is large, the accumulation of errors over $n-m$ steps can be significant. We note that this is the same type of error accumulation in evaluating a sum $S = \Sigma_{i=1}^{\infty}c_i$ of small numbers by direct recursive additions, which can fortunately be corrected through appropriately grouping the arithmetic operations as $S_1 + S_2 + \cdots = (c_1 + \cdots + c_{m_1}) + (c_{m_1+1} + \cdots + c_{m_2}) + \cdots\cdots$ with terms of similar order of magnitude in the same group S_i and $S_1 >> S_2 >> \cdots$. In this way, the rounding errors associated with a large number of additions inside a group S_i is of the magnitude of $\mathbf{u}S_i$, which can be much smaller than $\mathbf{u}S$. The same technique can be adopted for computing x_n as

$$x_n = x_0 + \Sigma_{i=1}^{n}q_i = x_0 + (q_1 + \cdots + q_{m_1}) + (q_{m_1+1} + \cdots + q_{m_2}) + \cdots\cdots.$$

  Specifically, the recurrence for x_n can be carried out in the following equivalent form

  Groupwise Solution Update: $z = x_0$; $\hat{x}_0 = 0$;
  For $n = 1, 2, \cdots$ until convergence
  $\quad\hat{x}_n = fl(\hat{x}_{n-1} + q_n) = \hat{x}_{n-1} + q_n + \psi_n$
  $\quad$if $n = m_i$ (i.e. group update)
  $\quad\quad z = fl(z + \hat{x}_n) = z + \hat{x}_n + \zeta_n$
  $\quad\quad\hat{x}_n = 0$
  $\quad$end if
  End for

  Such a groupwise update scheme has been suggested by Neumaier, and it has been worked out by Sleijpen and van der Vorst (see [18] for both references). By doing so, the error in the local recurrence is reduced. Indeed, for $i \ge m$, $|\hat{x}_i| = |z + \hat{x}_i - z| \sim |x - z| << |x|$. Then $|\psi_i| \sim \mathbf{u}|\hat{x}_i|$ (instead of $\mathbf{u}|x_i|$). Hence, $\Sigma_{i=m+1}^{n}|A\psi_i| << (n-m)\mathbf{u}|A||x|$.

In summary, with groupwise updating of the approximated solution, all three parts of δ_n can be maintained at the level of $\mathbf{u}|A||x|$. We mention that groupwise updating can also be used to obtain better performance of a code for modern architectures, because it allows for level-3 BLAS operations. This has been suggested in [21, page 52, note 5].

2.3 Some other residual replacement strategies

We briefly comment on some other residual replacement strategies.

For the naive strategy of "replacing always" (the residuals are computed always as $b - Ax_n$) or for "periodic replacement" (update periodically at every ℓ steps), replacement is carried out throughout the iteration, even when $\|r_n\|$ is very small. This, as we know, may result in large perturbations to the recurrence equations relative to $\|r_n\|$, since $|\eta_n'|$ is at least $|\xi_n| \sim \mathbf{u}|A||x_n|$, see (7). In that case, as $\|r_n\|$ decreases, the recurrence relation may be perturbed too much and hence the convergence property deteriorates. This is the typical behaviour observed in such implementations.

We note that if ξ_n can be made to decrease as $\|r_n\|$ does, then replacement can be carried out at later stages of the iterations. This leads to the strategy of "flying-restart" of Sleijpen and van der Vorst [18], which significantly reduces ξ_n, and hence η_n', at a replacement step. In the flying-restart strategy b is replaced by $fl(b - Ax_m)$ at some but not all of the residual replacement steps (say $m = n_1, n_2, \cdots n_l$), in addition to the residual replacement $r_m = fl(b - Ax_m)$. The advantage of this is that, at the flying-restart step n_{i+1}, the residual is updated by $r_{n_{i+1}} = fl(r_{n_i} - A\hat{x}_{n_{i+1}}) = r_{n_i} - A\hat{x}_{n_{i+1}} + \xi_{n_{i+1}}$ (noting that $b \leftarrow r_{n_i}$) and $|\xi_{n_{i+1}}| \sim \mathbf{u}(|r_{n_i}| + |A||\hat{x}_{n_{i+1}}|)$. Then

$$|r_{n_i} - A\hat{x}_{n_{i+1}} - r_{n_{i+1}}| = |\zeta_{n_{i+1}}| \sim \mathbf{u}(|r_{n_i}| + |A||\hat{x}_{n_{i+1}}|)$$

which decreases as r_{n_i} and $\hat{x}_{n_{i+1}}$ decrease. This is the term that determines the perturbation to the recurrence and can be kept small relative to r_n. However, the deviation satisfies

$$b - Ax_{n_{i+1}} - r_{n_{i+1}} = b - Ax_{n_i} - r_{n_i} - \xi_{n_{i+1}},$$

(assuming $x_{n_{i+1}} = x_{n_i} + \hat{x}_{n_{i+1}}$). Namely, the deviation at each flying-restart step carries forward to the later steps. Therefore flying-restart should only be used at carefully selected steps where $\xi_{n_i} \sim \mathbf{u}(\|b\| + N\|A\|\|x\|)$. However, it is not easy to identify a condition to monitor this. It is also necessary to have two different conditions for the residual replacement and flying-restart. Fortunately, our discussion in the last two subsections shows that carrying out replacement carefully at some selected steps, in combination with groupwise update, is usually sufficient. We shall not pursue the flying-restart idea further in this paper.

3 Error Analysis of the Residual Replacement Scheme

In this section, we formally analyze the residual replacement strategy as developed in Section 2.1 (and presented in Algorithm 2 below). In particular, we develop a computable bound for $\|\delta_n\|$ and $\|\eta'_n\|$, that can be used for the implementation of the residual replacement condition.

We first summarize residual replacement strategy in the following algorithm, written in a form that identifies relevant rounding errors for later theoretical analysis.

Algorithm 2: Iterative Method with Residual Replacement:
Given an initial approximation $z = x_0$ (a floating point vector);
set $\hat{x}_0 = 0$; $r_0 = fl(b - Ax_0) = b - Ax_0 + \xi_0$;
For $n = 1, 2, \cdots$, until convergence
    Generate a correction vector q_n by the method;
    $\hat{x}_n = fl(\hat{x}_{n-1} + q_n) = \hat{x}_{n-1} + q_n + \psi_n$
    $\hat{r}_n = fl(r_{n-1} - Aq_n) = r_{n-1} - Aq_n + \eta_n$
    if residual replacement condition (8) holds
        $z = fl(z + \hat{x}_n) = z + \hat{x}_n + \zeta_n$
        $\hat{x}_n = 0$
        $r_n = fl(b - Az) = b - Az + \xi_n$
    else
        $r_n = \hat{r}_n$
    end if
    (denote but not compute $x_n = z + \hat{x}_n$ and $\delta_n = b - Ax_n - r_n$)
End for
$z = fl(z + \hat{x}_n) = z + \hat{x}_n + \zeta_n$

Note that x_n and δ_n are theoretical quantities as defined by the formulas and are not to be computed. The vectors $\psi_n, \eta_n, \zeta_n, \xi_n$ represent rounding error terms, due to finite precision arithmetic.

At step n of the iterative method, q_n is computed in finite precision arithmetic by the algorithm. However, the rounding errors involved in the computation of q_n are irrelevant for the deviation of the two residuals, which solely depends on the different treatment of q_n in the recurrences for r_n and x_n.

Throughout this paper, we assume that A is a floating point matrix. Our error analysis is based on the following standard model for roundoff errors in basic matrix computations [8, p.66] (all inequalities are componentwise).

$$fl(x+y) = x+y+e \quad \text{with} \quad |e| \leq \mathbf{u}(|x+y|) \tag{9}$$

$$fl(Ax) = Ax+e \quad \text{with} \quad |e| \leq \mathbf{u}N|A||x| + O(\mathbf{u}^2). \tag{10}$$

where $x, y \in R^N$ are floating point vectors, N is a constant associated with the matrix-vector multiplication (for instance, the maximal number of nonzero entries per row of A).

It is easy to show that

$$fl(y+Ax) = y+Ax+e \quad \text{with} \quad |e| \leq \mathbf{u}(|y+Ax| + N|A||x|) + O(\mathbf{u}^2).$$

Using this, the following lemma, which includes (1) and (2), is obtained.

Lemma 1 *The error terms in the computed recurrence of Algorithm 2 are bounded as follows:*

$$|\psi_n| \leq \mathbf{u}|\hat{x}_n| + O(\mathbf{u}^2) \tag{11}$$

$$|\eta_n| \leq \mathbf{u}(|\hat{r}_n| + N|A||q_n|) + O(\mathbf{u}^2). \tag{12}$$

For a step at which a residual replacement is carried out:

$$|\zeta_n| \leq \mathbf{u}|x_n| + O(\mathbf{u}^2), \tag{13}$$

$$|\xi_n| \leq \mathbf{u}(|r_n| + N|A||x_n|) + O(\mathbf{u}^2). \tag{14}$$

Proof From (9), we have that $|\psi_n| \leq \mathbf{u}|\hat{x}_{n-1} + q_n| \leq \mathbf{u}(|\hat{x}_n| + |\psi_n|)$. This leads to the bound for $|\psi_n|$. For a residual replacement step, the updated z is x_n by definition, that is $x_n = z + \hat{x}_n + \zeta_n$. Therefore, $|\zeta_n| \leq \mathbf{u}|x_n| + O(\mathbf{u}^2)$. The bounds for η_n and ξ_n follow similarly. □

Lemma 2 *Let m be the number of step at which a residual replacement is carried out and let $n > m$ denote a later step, but still before the next replacement step. Then, we have that*

$$\Sigma_{i=m+1}^{n}|\psi_i| \leq \mathbf{u}\Sigma_{i=m+1}^{n}|\hat{x}_i| + O(\mathbf{u}^2),$$

$$\Sigma_{i=m+1}^{n}|q_i| \leq (2+\mathbf{u})\Sigma_{i=m+1}^{n}|\hat{x}_i|$$

and

$$\Sigma_{i=m+1}^{n}|\eta_i| \leq \mathbf{u}\Sigma_{i=m+1}^{n}|\hat{r}_i| + 2\mathbf{u}N|A|\Sigma_{i=m+1}^{n}|\hat{x}_i| + O(\mathbf{u}^2).$$

Proof The first bound follows directly from Lemma 1. For $i \geq m+1$ we have that $q_i = \hat{x}_i - \hat{x}_{i-1} - \psi_i$. Noting that $\hat{x}_m = 0$, it follows

$$\Sigma_{i=m+1}^{n}|q_i| \leq \Sigma_{i=m+1}^{n}(|\hat{x}_i| + |\hat{x}_{i-1}| + |\psi_i|) \leq (2+\mathbf{u})\Sigma_{i=m+1}^{n}|\hat{x}_i|.$$

Similarly,

$$\begin{aligned}\Sigma_{i=m+1}^{n}|\eta_i| &\leq \mathbf{u}\Sigma_{i=m+1}^{n}(|\hat{r}_i| + N|A||q_i|) + O(\mathbf{u}^2)\\ &\leq \mathbf{u}\Sigma_{i=m+1}^{n}|\hat{r}_i| + 2\mathbf{u}N|A|\Sigma_{i=m+1}^{n}|\hat{x}_i| + O(\mathbf{u}^2).\end{aligned}$$

□

We now consider the deviation of the two residuals.

Lemma 3 *Let m be the number of an iteration step at which residual replacement is carried out and let $n > m$ denote a later iteration step, still before the next replacement step. Then, we have that $\delta_m = -\xi_m$ and*

$$\delta_n = \delta_{n-1} - (A\psi_n + \eta_n) = -\xi_m - \Sigma_{i=m+1}^{n}(A\psi_i + \eta_i).$$

Proof At step m, by the definition of x_m in Algorithm 2, $x_m = z + \hat{x}_m = z$ with z being the updated z-vector and $\hat{x}_m = 0$. Furthermore, $r_m = fl(b - Az) = fl(b - Ax_m) = b - Ax_m + \xi_m$. Therefore $\delta_m = -\xi_m$. Hence, for the range of $n > m$, and before the next residual replacement step:

$$\begin{aligned}\delta_n &= b - Ax_n - r_n = b - A(z + \hat{x}_n) - \hat{r}_n\\ &= b - A(z + \hat{x}_{n-1} + q_n + \psi_n) - (\hat{r}_{n-1} - Aq_n + \eta_n)\\ &= \delta_{n-1} - A\psi_n - \eta_n\\ &= \delta_m - \Sigma_{i=m+1}^{n}(A\psi_i + \eta_i).\end{aligned}$$

□

With Lemma 2, we obtain the following computable bound on δ_n.

Lemma 4 *Let m be the number of an iteration step at which residual replacement is carried out and let $n > m$ denote a later iteration step, still before the next replacement step. Then, we have $\|\delta_m\| \leq \mathbf{u}(\|r_m\| + N\|A\|\|x_m\|) + O(\mathbf{u}^2)$ and*

$$\|\delta_n\| \leq \mathbf{u}N\|A\|\|x_m\| + \mathbf{u}(1+2N)\|A\|\Sigma_{i=m+1}^{n}\|\hat{x}_i\| + \mathbf{u}\Sigma_{i=m}^{n}\|r_i\| + O(\mathbf{u}^2).$$

Proof The bound for $\|\delta_m\|$ follows from that for ξ_m, see (14). From Lemma 2 and Lemma 3, it follows that

$$\begin{aligned}|\delta_n| &\leq |\xi_m| + \Sigma_{i=m+1}^{n}(|A||\psi_i| + |\eta_i|)\\ &\leq \mathbf{u}(|r_m| + N|A||x_m|) + |A|\mathbf{u}\Sigma_{i=m+1}^{n}|\hat{x}_i|\\ &\quad +\mathbf{u}\Sigma_{i=m+1}^{n}|\hat{r}_i| + 2\mathbf{u}N|A|\Sigma_{i=m+1}^{n}|\hat{x}_i| + O(\mathbf{u}^2)\\ &= \mathbf{u}N|A||x_m| + \mathbf{u}(1+2N)|A|\Sigma_{i=m+1}^{n}|\hat{x}_i| + \mathbf{u}\Sigma_{i=m}^{n}|r_i| + O(\mathbf{u}^2),\end{aligned}$$

which leads to the bound for δ_n in terms of norms. □

We note that it is possible to obtain a sharper bound by accumulating the vectors in the bound for $|\delta_n|$. Our experiments do not show any significant advantage of such an approach. We next consider the perturbation to the recurrence.

Theorem 1 *Consider step n of the iteration and let $m < n$ be the last step before n, at which a residual replacement is carried out. If replacement is also done at step n, then let $x_n' = fl(x_m + \hat{x}_n) = x_m + \hat{x}_n + \zeta_n$ be the computed approximate solution and $r_n' = fl(b - Ax_n') = b - Ax_n' + \xi_n$ be the residual. Then the residual r_n' satisfies the following approximate recurrence*

$$r_n' = r_{n-1} - Aq_n + \eta_n', \tag{15}$$

where $\eta_n' = \eta_n + \delta_n - A\zeta_n + \xi_n$ and

$$\|\eta_n'\| \leq \mathbf{u}\|A\|(1+2N)(\|x_m\| + \Sigma_{i=m+1}^{n}\|\hat{x}_i\|) + \mathbf{u}\Sigma_{i=m}^{n}\|r_i\| + O(\mathbf{u}^2). \tag{16}$$

Proof First, in the notation of Alg. 2, $x_n' = x_m + \hat{x}_n + \zeta_n = x_n + \zeta_n$. Then,

$$\begin{aligned} r_n' &= b - Ax_n - A\zeta_n + \xi_n \\ &= r_n + \delta_n - A\zeta_n + \xi_n \\ &= r_{n-1} - Aq_n + \eta_n + \delta_n - A\zeta_n + \xi_n \\ &= r_{n-1} - Aq_n + \eta_n', \end{aligned}$$

where we have used that $b - Ax_n = r_n + \delta_n$ and $r_n = \hat{r}_n = r_{n-1} - Aq_n + \eta_n$. Furthermore, by Lemma 3,

$$\eta_n + \delta_n = \xi_m - \Sigma_{i=m+1}^{n} A\psi_i - \Sigma_{i=m+1}^{n-1}\eta_i.$$

Also, $\|A\zeta_n\| \leq \mathbf{u}\|A\|(\|x_m\| + \|\hat{x}_n\|)$, and $\|\xi_n\| \leq \mathbf{u}(\|r_n'\| + N\|A\|\|x_n'\|) + O(\mathbf{u}^2) \leq \mathbf{u}(\|r_n'\| + N\|A\|\|x_m\| + N\|A\|\|\hat{x}_n\|) + O(\mathbf{u}^2)$. Combining these three, and using that $r_n' = r_n + O(\mathbf{u})$, the bound on $\|\eta_n'\|$ is obtained as in Lemma 4. □

Note that bound (16) is computable at each iteration step. Therefore, we can implement the residual replacement criterion (8) with this bound instead of $\|\eta_n'\|$. We note that the factor 2 in the bound comes from the bound for q_i in Lemma 2, which is pessimistic since $q_i \sim \hat{x}_i$. Therefore, we can use the following d_n as an estimate for $\|\eta_n'\|$:

$$d_n \equiv \mathbf{u}N\|A\|(\|x_m\| + \Sigma_{i=m+1}^{n}\|\hat{x}_i\|) + \mathbf{u}\Sigma_{i=m}^{n}\|r_i\|. \tag{17}$$

Hence, we shall use the following residual replacement criterion, that is residual replacement is done if

$$d_{n-1} \leq \epsilon\|\hat{r}_{n-1}\|, \text{ and } d_n > \epsilon\|\hat{r}_n\|. \tag{18}$$

With this strategy, the replaced residual vector r_n satisfies the recurrence equation (15) with $\|\eta_n'\| \sim O(\epsilon)\|\hat{r}_n\|$. With this property, we consider situations where r_n converges. We now discuss convergence of the true residual.

Theorem 2 *Consider Algorithm 2 with the residual replacement criterion (18), and assume that the algorithm terminates at step $n = K$ with $\|r_K\| < \mathbf{u}\|A\|\,\|x_K\|$. Let m be the number of the last residual replacement iteration step before termination. If*

$$L = (K - m + 1)(1 + 2N)\|A\|\|A^{-1}\|(1 + 3/\epsilon) < 1/\mathbf{u}, \tag{19}$$

then

$$\begin{aligned} \|b - Ax_K\| &\leq \|r_K\| + \mathbf{u}N\|A\|\|x_K\|/(1-\mathbf{u}L) + O(\mathbf{u}^2) \\ &\sim \mathbf{u}N\|A\|\|x_K\|. \end{aligned}$$

Proof From (17), we have $d_K > \mathbf{u}N\|A\|(\|x_m\| + \|\hat{x}_K\|) \geq \mathbf{u}\|A\|\,\|x_K\|$. Furthermore, at the termination step, we have $\|r_K\| < \mathbf{u}\|A\|\,\|x_K\|$ and hence $d_K > \|r_K\| > \epsilon\|r_K\|$. Since m is the last updating step, we have for $n \geq m$, $d_n > \epsilon\|r_n\|$ as otherwise there would be another residual replacement after m. That implies $\|r_n\| < d_n/\epsilon \leq d_K/\epsilon$. Define

$$\tilde{d}_n \equiv \mathbf{u}N\|A\|\|x_m\| + \mathbf{u}\|A\|(1+2N)\Sigma_{i=m+1}^{n}\|\hat{x}_i\| + \mathbf{u}\Sigma_{i=m}^{n}\|\hat{r}_i\|,$$

which is an upper bound for $\|\delta_n\|$ (Lemma 4) and $\tilde{d}_n \geq d_n$. Then

$$\begin{aligned} \|\hat{x}_n\| &= \|x_n - x_m\| = \|A^{-1}((b - Ax_m) - (b - Ax_n))\| \\ &= \|A^{-1}(r_m + \delta_m - r_n - \delta_n)\| \\ &\leq \|A^{-1}\|(\|r_m\| + \|r_n\| + \|\delta_n - \delta_m\|) \\ &\leq \|A^{-1}\|(1 + 2/\epsilon)\tilde{d}_K + O(\mathbf{u}^2), \end{aligned}$$

where $\|\delta_n - \delta_m\| \leq \tilde{d}_n + O(\mathbf{u}^2) \leq \tilde{d}_K + O(\mathbf{u}^2)$. Thus,

$$\begin{aligned} \tilde{d}_K &= \mathbf{u}N\|A\|\|x_K - \hat{x}_K\| + \mathbf{u}(1+2N)\|A\|\Sigma_{i=m+1}^{K}\|\hat{x}_i\| + \mathbf{u}\Sigma_{i=m}^{K}\|\hat{r}_i\| \\ &\leq \mathbf{u}N\|A\|\|x_K\| + \mathbf{u}N\|A\|\|\hat{x}_K\| + \mathbf{u}(1+2N)\|A\|\Sigma_{i=m+1}^{K}\|\hat{x}_i\| + \mathbf{u}\Sigma_{i=m}^{K}\|\hat{r}_i\| \\ &\leq \mathbf{u}N\|A\|\|x_K\| + \mathbf{u}(K-m+1)(1+2N)\|A\|\|A^{-1}\|(1+2/\epsilon)\tilde{d}_K \\ &\quad + \mathbf{u}(K-m+1)\tilde{d}_K/\epsilon + O(\mathbf{u}^2) \\ &\leq \mathbf{u}N\|A\|\|x_K\| + \mathbf{u}(K-m+1)(1+2N)\|A\|\|A^{-1}\|(1+3/\epsilon)\tilde{d}_K + O(\mathbf{u}^2) \\ &\leq \mathbf{u}N\|A\|\|x_K\| + \mathbf{u}L\tilde{d}_K + O(\mathbf{u}^2), \end{aligned}$$

which implies

$$\tilde{d}_K \leq \mathbf{u}N\|A\|\|x_K\|/(1-\mathbf{u}L) + O(\mathbf{u}^2).$$

Thus the bound follows from

$$\|b - Ax_K\| \leq \|r_K\| + \|\delta_K\| \leq \|r_K\| + \tilde{d}_K + O(\mathbf{u}^2).$$

□

We add two remarks with respect to this theorem.

Remark 1: If the main condition (19) is satisfied, then the deviation, and hence the true residual, will remain at the level of $\mathbf{u}N\|A\|\|x_K\|$ at termination. Such an approximate solution is backward stable and it is best one can expect. The condition suggests that ϵ should not be chosen too small. Otherwise, the replacement strategy will be terminated too early so that the accumulation after the last replacement might become significant. As can be expected, however, the theoretical condition is more restrictive than practically necessary and our numerical experience suggests that ϵ can be much smaller than what (19) dictates, without destroying the conclusion of the theorem.

Remark 2: On the other hand, in Section 2.1 we have seen that ϵ controls perturbations to the recurrence of r_n, and for this reason it is desirable to choose it as small as possible. In our experience, there is a large range of ϵ around $\sqrt{\mathbf{u}}$ that balances the two needs.

4 Reliable Implementation of Iterative Methods

In this section, we summarize the main results of the previous sections into a complete implementation. We also address some implementation issues.

It is easy to see from the definition of d_n (see (17)) that it increases except at the residual replacement steps when it is reset to $\mathbf{u}(N\|A\|\|x_m\| + \|r_m\|)$. Our residual replacement strategy is to reduce d_n whenever necessary (as determined by the replacement criterion) so as to keep it at the level of $\mathbf{u}N\|A\|\|x_K\|$ at termination. With the use of criterion (18), however, there are situations where the residual replacement is carried out in consecutive steps while d_n remains virtually unchanged, namely when $\|r_n\|$ stays around $d_n/\epsilon \sim \mathbf{u}N\|A\|\|x_n\|/\epsilon$. ¿From the stability point of view, it is preferred not to replace the residuals in such situations. To avoid unnecessary replacement in such cases, we impose as an additional condition that residual replacement is carried out only when d_n has a nontrivial increase from the d_m of the previous replacement step m.

Therefore, we propose $d_n > 1.1 d_m$ as a condition in addition to (18) for the residual replacement. The following scheme sketches a complete implementation.

Algorithm 3: Reliable Implementation of Algorithm 1.
Input an initial approximation $z = x_0$; a residual replacement threshold ϵ; an estimate of $N\|A\|$;
Set $r_0 = b - Ax_0$; $\hat{x}_0 = 0$; $d_{init} = d_0 = \mathbf{u}(\|r_0\| + N\|A\|\|x_0\|)$,
For $n = 1, 2, \cdots$, until convergence
  Generate a correction vector q_n by the Iterative Method;
  $\hat{x}_n = \hat{x}_{n-1} + q_n$
  $r_n = r_{n-1} - Aq_n$
  $d_n = d_{n-1} + \mathbf{u}N\|A\|\,\|\hat{x}_n\| + \mathbf{u}\|r_n\|$
  if $d_{n-1} \le \epsilon\|r_{n-1}\|$, $d_n > \epsilon\|r_n\|$ and $d_n > 1.1 d_{init}$
    $z = z + \hat{x}_n$
    $\hat{x}_n = 0$
    $r_n = b - Az$
    $d_{init} = d_n = \mathbf{u}(\|r_n\| + N\|A\|\|z\|)$
  end if
End for
$z = z + \hat{x}_n$

Remark: In this reliable implementation, we need estimates for N (the maximal number of nonzero entries per row of A) and $\|A\|$. In our experience with sparse matrices, the simple choice $N = 1$, still leads to a practical estimate d_n for $\|\delta_n\|$. In any case, we note that precise estimates are not essential, because the replacement threshold ϵ can be adjusted. We also need to choose this ϵ. Our extensive numerical testing (see section 5) suggests that $\epsilon \sim \sqrt{\mathbf{u}}$ is a practical criterion. However, there are examples where this choice leads to stagnating residuals at some unacceptable level. In such cases, choosing a smaller ϵ will regain the convergence to $O(\mathbf{u})$.

The presented implementation requires one extra matrix-vector multiplication when an replacement is carried out. Since only a few steps with replacement are required, this extra cost is marginal relative to the other costs. However, some savings can be made by selecting a slightly smaller ϵ and carrying out residual replacement at the step next to the one for which the residual replacement criterion is satisfied (cf [18]). It also requires one extra vector storage for the groupwise solution update (for z) and computation of a vector norm $\|\hat{x}_n\|$ for the update of d_n ($\|r_n\|$ is usually computed in the algorithm for stopping criteria).

5 Numerical Examples

In this section, we present some numerical examples to show how Algorithm 3 works and to demonstrate its effectiveness. We present our testing results for CG, Bi-CG and CGS. All tests are carried out in MATLAB on a SUN Sparc-20 workstation, with $\mathbf{u} \approx 10^{-16}$.

In all examples, unless otherwise specified, the replacement threshold ϵ is chosen to be 10^{-8}. $\|A\|_\infty$ is explicitly computed and N is set to 1. In Examples 1 and 2, we also compare d_n and the deviation $\|\delta_n\|$.

Example 1: The matrix is a finite-difference discretization on a 64×64 grid for

$$-\nabla(a(x,y)\nabla u) = f(x,y) \text{ on } R = (0,1) \times (0,1),$$

with a homogeneous Dirichlet boundary condition. $a(x,y) = exp(y^2)$ and $f(x,y) = x^2 y$. We apply CG and Reliable CG (i.e. Alg. 3) to solve this linear system and the convergence results are given in Figure 1.

In Figure 1 (and similarly in Figures 2 and 3 for the next example), we give in (a) the convergence history of the (normalized) computed residual for CG (solid line), the (normalized) true residuals for CG (dashed line) and for reliable CG (dotted line). In (b), we also give the (normalized) deviations of the two residuals $\|\delta_n\| = \|b - Ax_n - r_n\|$ for CG (dash-dotted line) and for reliable CG (dotted line) and the bound d_n for reliable CG (in x-mark).

Example 2: The matrix is a finite-difference discretization on a 64×64 grid for the following convection diffusion equation

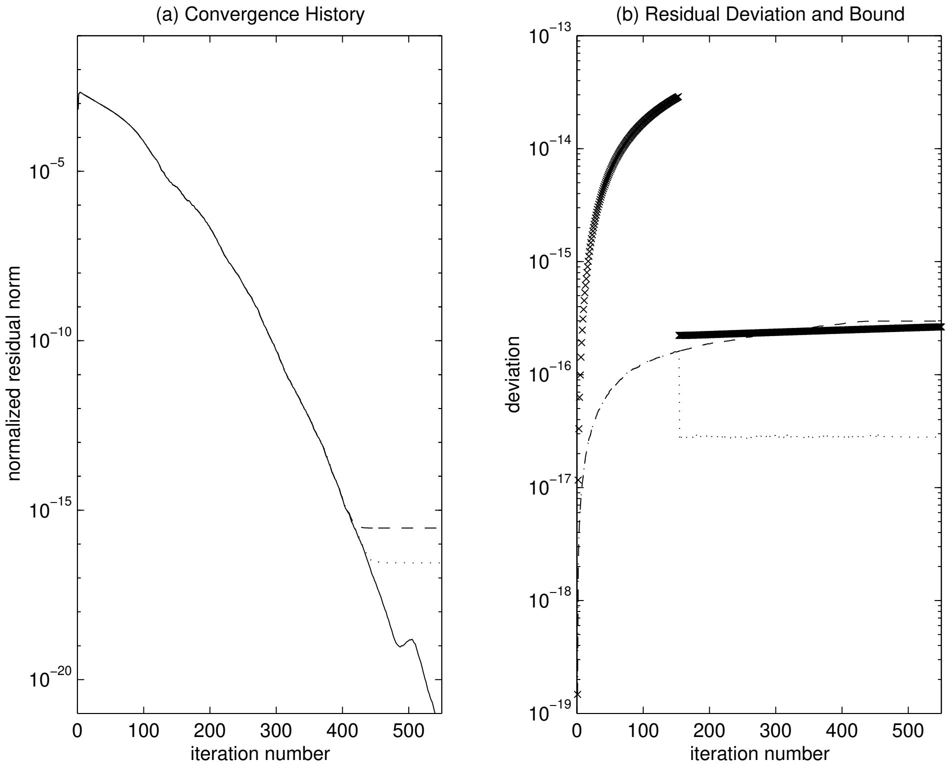
$$-\triangle u + \gamma(xu_x + yu_y) + \beta u = f(x,y) \quad \text{on} \quad (0,1)^2;$$

with a homogeneous Dirichlet boundary condition. The function f is a constant. We consider Bi-CG and CGS for solving the linear systems with $\gamma = -50, \beta = 0$, and $\gamma = -10, \beta = 1$, respectively. The results are shown in Figure 2 for Bi-CG, and in Figure 3 for CGS.

In the above examples, we have observed the following typical convergence behaviour. For the original implementations, the deviation increases and finally stagnates at some level, which is exactly where the true residual stagnates, while the computed residual continues to converge. With the reliable implementations, when the deviation increases to a certain level relative to r_n, a residual replacement is carried out and this reduces the error level. Eventually, the deviation and hence the true residual arrive at the level of $\mathbf{u}\|A\|\|x\|$. We also note that the bound d_n captures the behaviour of $\|\delta_n\|$ very closely, although it may be an overestimate for δ_n by a few orders of magnitude. In all three cases, the final residual norms for the reliable implementation are smaller than the ones as obtained by the MATLAB function $A\backslash b$.

Example 3: In this case, we have tested the algorithm for Bi-CG (or CG if symmetric definite) and CGS, on the Harwell-Boeing collection of sparse matrices [3]. We compare the original implementations, the reliable implementations and the implementations of Sleijpen and van der Vorst [18] (based on their replacement criteria (16) and (18)). In Table 1, we give the results for those matrices for which the computed residuals converge to a level smaller than $\mathbf{u}\|A\|\|x\|$ so that there is a deviation of the two residuals. For those cases where b is not given, we choose it such that a

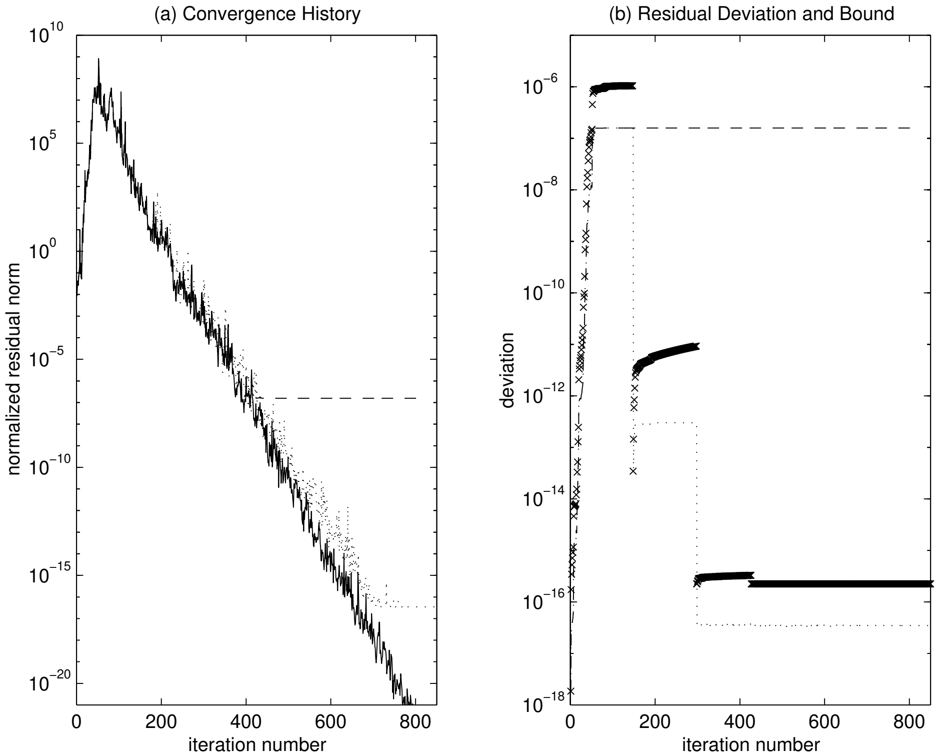
Figure 1: **Example 1 (CG)** (a): solid - computed residual of CG; dashed - true residual of CG; dotted - true residual of reliable CG; (b): dash-dotted - $||b - Ax_n - r_n||$ of CG, dotted - $||b - Ax_n - r_n||$ of reliable CG; x - d_n of reliable CG



given random vector is the solution. We note that for some matrices, it may take $10n$ iterations to achieve that, which is not practical. However, we have included these results in order to show that even with excessive numbers of iterations, we still arrive at small true residuals eventually. We list the normalized residuals $res = ||b - Ax_n||/(||A||||x_n||)$ attained by the three implementations and by Gaussian elimination with partial pivoting (MATLAB $A\backslash b$). We also list the number of residual replacements (n_r) for our reliable implementations and the number of flying-restart (n_f) and the number of residual replacements (n_r) for the implementations of Sleijpen and van der Vorst (SvdV). There are two cases for which the computed residuals do not converge to $O(\mathbf{u})||b||$ with the choice of $\epsilon = 1e - 8$. For those cases, a slightly smaller ϵ will recover the stability and the results are listed in the last row of the table.

We see that in all cases, the reliable implementation reduces the normalized residual to $O(\mathbf{u})$ and $res2$ is the smallest among the three implementations, even smaller than MATLAB $A\backslash b$. The improvement on the true residual is more apparent in CGS than in Bi-CG (or CG). Except in a few cases, both the reliable implementation presented here and the implementation of Sleijpen and van der Vorst work well and are comparable. So the main advantage of the new approach is its simplicity and an occasional improvement in accuracy.

Figure 2: **Example 2 (Bi-CG)** (a): solid - computed residual of Bi-CG; dashed - true residual of Bi-CG; dotted - true residual of reliable Bi-CG; (b): dashed - $||b - Ax_n - r_n||$ of Bi-CG, dotted - $||b - Ax_n - r_n||$ of reliable Bi-CG; x - d_n of reliable Bi-CG



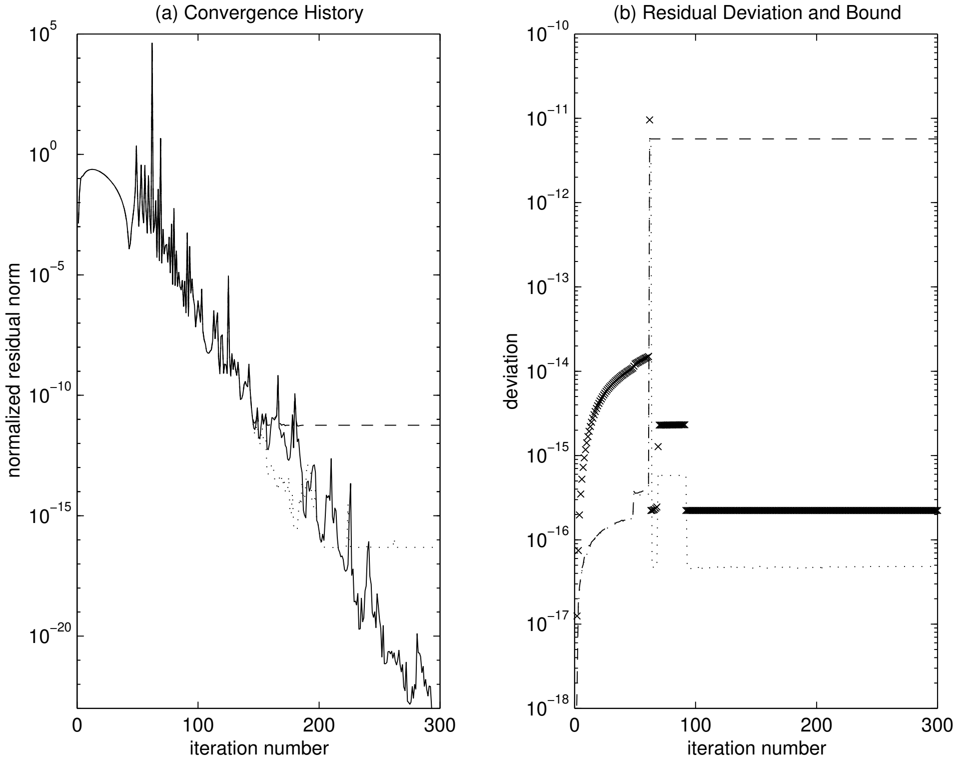
6 Concluding Remarks

We have presented a new residual replacement scheme for improving the convergence of the true residuals in finite precision implementations of Krylog subspace iterative methods. By carefully monitoring the deviation of the computed residual and the true residual and incorporating the earlier ideas on residual replacement, we obtain a reliable implementation that preserves the convergence mechanism of the computed residuals, as well as sufficiently small deviations. An error analysis shows that this approach works under certain conditions, and numerical tests demonstrate its effectiveness. Comparison with an earlier approach shows that the new scheme is simpler and easier to implement as an add-on to existing implementations for iterative methods.

We point out that the basis for the present work is the understanding that the convergence behaviour (of computed residuals) in finite precision arithmetic is preserved under small perturbations to the recurrence relations. Such a supporting analysis is available for Bi-CG (and CG) [20], but it is still an empirical observation for most other Krylov subspace methods. It would be interesting to derive a theoretical analysis confirming this phenomenon for those methods as well.

Acknowledgements: We would like to thank Ms. Lorrita McKnight for assistance in carrying out the tests on Harwell-Boeing matrices.

Figure 3: **Example 2 (CGS)** (a): solid - computed residual of CGS; dashed - true residual of CGS; dotted - true residual of reliable CGS; (b): dashed - $\|b - Ax_n - r_n\|$ of CGS, dotted - $\|b - Ax_n - r_n\|$ of reliable CGS; x - d_n of reliable CGS



References

[1] R. E. Bank and T. F. Chan, *An Analysis of the Composite Step Biconjugate Gradient Algorithm for Solving nonsymmetric Systems,* Numer. Math., 66:295-319 (1993).

[2] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, V. Pozo, Romime C., and H. van der Vorst, *Templates for the solution of linear systems: Building blocks for iterative methods*, SIAM, Philadelphia, PA, 1994.

[3] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Sparse Matrix Test Problems,* ACM Trans. Math. Softw., 15 (1989), pp.1-14.

[4] R. Fletcher, *Conjugate Gradient Methods for Indefinite Systems,* in Proc. Dundee Conference on Numerical Analysis, 1975, Lecture Notes in Mathematics 506, G. A. Watson, ed., Springer-Verlag, Berlin, 1976, pp. 73-89.

[5] R. Freund, *A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems* SIAM J. Sci. Stat. Comput. 14, pp. 470-482 (1993).

[6] R. Freund, G. Golub and N. Nachtigal, *Iterative solutions of linear systems* Acta Numerica 1(1992):57-100.

[7] R. W. Freund and N. M. Nachtigal, *QMR : a Quasi-minimal Residual Method for non-Hermitian Linear Systems,* Numer. Math., 60:315-339 (1991).

[8] G.H. Golub, C.F. Van Loan, *Matrix Computations,* The Johns Hopkins University Press, Baltimore, 1983.

[9] A. Greenbaum, *Behavior of Slightly Perturbed Lanczos and Conjugate-Gradient Recurrences,* Lin. Alg. and its Appl., 113:7-63 (1989).

[10] A. Greenbaum, *Estimating the attainable accuracy of recursively computed residual methods,* SIAM J. Matrix Anal. Appl. 18(1997):535-551.

[11] M. Gutknecht, *Lanczos-type solvers for nonsymmetric linear systems of equations,* Acta Numerica 6, pp. 271-397 (1997).

[12] M. Hestenes and E. Stiefel, *Methods of Conjugate Gradients for solving linear systems*, J. Res. NBS. 49:409-436 (1952).

[13] C. Lanczos, *Solution of Systems of Linear Equations by Minimized Iterations,* J. Res. Natl. Bur. Stand. 49, pp. 33-53 (1952).

[14] Y. Notay, *On the convergence rate of the conjugate gradients in presence of rounding errors*, Numer. Math. 65:301-317 (1993).

[15] C. Paige, *Accuracy and effectiveness of the Lanczos algorithm for the Symmetric eigenproblem,* Linear Alg. Appl. 34(1980):235-258.

[16] Y. Saad, *Iterative Methods for Sparse Linear Systems* PWS Publishing, Boston, MA, 1996.

[17] G. Sleijpen and D. Fokkema, *BICGSTAB(L) for linear equations involving unsymmetric matrices with complex spectrum* Electronic Trans. Numer. Anal. 1, pp. 11-32 (1993).

[18] G. Sleijpen and H. van der Vorst, *Reliable updated residuals in hybrid Bi-CG methods* Computing 56:144-163 (1996).

[19] P. Sonnefeld, *CGS, A Fast Lanczos-type Solver for nonsymmetric Linear Systems* SIAM J. Sci. Stat. Comput. 10, p. 36-52 (1989).

[20] C. H. Tong and Q. Ye, *Analysis of the Finite Precision Bi-Conjugate Gradient algorithm for Nonsymmetric Linear Systems,* Stanford SCCM Report 95-11, Stanford University, Stanford, CA, October 1995. Math. Comp. (to appear).

[21] H.A. van der Vorst, *The performance of FORTRAN implementations for preconditioned conjugate gradients on vector computers*, Parallel Computing, 3:49-58 (1986).

[22] H. van der Vorst, *BiCGSTAB, A fast and smoothly converging variant of BiCG for the solution of nonsymmetric linear systems* SIAM J. Sci. Stat. Comput. 13, 631-644 (1992).

Table 1: **Example 3: Comparison of Normalized Residuals** res0 - $A\backslash b$; res1 - original implementation; res2 - reliable implementation; res3 - implementation of SvdV.

	$A\backslash b$	Bi-CG (or CG)					CGS				
$Matrix$	$res0$	$res1$	$res2,\ n_r$		$res3,\ n_f\ (n_r)$		$res1$	$res2,\ n_r$		$res3,\ n_f\ (n_r)$	
bcspwr06	-	7e-15	1e-20	19	1e-19	5(9)	6e-13	2e-17	14	3e-17	32(48)
bcspwr07	-	1e-15	2e-17	46	9e-17	2(6)	2e-12	2e-17	20	1e-7	220(404)
bcspwr08	-	2e-15	3e-17	9	1e-16	5(7)	7e-14	2e-17	14	4e-16	79(103)
bcspwr09	-	3e-15	2e-20	42	6e-20	5(6)	3e-13	2e-17	13	4e-16	40(70)
jpwh991	1e-16	9e-17	3e-17	1	3e-17	1(1)	7e-17	3e-17	1	3e-17	1(1)
fs6801	1e-17	7e-17	1e-17	2	8e-18	1(1)	2e-16	9e-18	2	1e-17	3(5)
fs6802	8e-18	1e-16	8e-18	3	2e-17	1(1)	4e-16	8e-18	6	2e-17	4(4)
fs6803	6e-18	3e-16	1e-13[2]	11	8e-16	4(5)	4e-14	6e-17	33	1e-17	3(5)
fs7601	7e-18	7e-17	9e-18	1	7e-18	1(1)	5e-15	5e-18	2	6e-18	1(2)
jagmesh1	3e-16	4e-15	5e-17	2	1e-17	3(5)	1e-12	5e-17	5	5e-15	20(26)
nos3	1e-16	3e-16	6e-17	2	7e-17	1(1)	2e-16	6e-17	2	7e-17	1(1)
nos4	8e-17	2e-16	5e-17	1	6e-17	1(1)	2e-16	5e-17	1	8e-17	1(1)
nos5	1e-16	3e-16	5e-17	2	6e-17	1(1)	3e-16	6e-17	2	7e-17	1(1)
nos6	6e-17	4e-16	3e-17	9	8e-17	1(1)	4e-16	2e-17	14	1e-16	1(1)
1138bus	9e-18	2e-16	1e-17	8	9e-17	1(1)	7e-10	4e-12[3]	21	2e-10	17(29)
orsirr1	4e-17	1e-15	1e-17	2	2e-17	5(9)	9e-14	1e-17	6	2e-17	11(18)
orsirr2	7e-17	2e-16	1e-17	2	1e-17	3(4)	5e-14	1e-17	5	2e-16	4(7)
orsreg1	2e-16	8e-16	7e-17	1	4e-16	1(1)	7e-15	8e-17	2	6e-16	3(5)
pores1	3e-17	2e-16	3e-17	2	4e-17	2(3)	5e-15	3e-17	5	9e-17	2(4)
pores3	3e-17	8e-16	2e-17	3	3e-16	4(5)	2e-12	2e-17	11	5e-17	16(28)
saylr3	-	3e-16	3e-17	2	6e-17	1(1)	2e-16	3e-17	2	7e-17	1(1)
saylr4	3e-16	1e-15	8e-17	4	5e-16	1(1)	2e-15	4e-19	27	7e-19	8(15)
sherman1	5e-17	3e-16	3e-17	2	5e-17	1(2)	2e-10	3e-17	3	4e-17	4(41)
sherman3	6e-19	2e-17	4e-19	9	1e-18	23(114)	5e-10	6e-19	30	1e-16	62(407)
sherman4	6e-17	2e-16	3e-17	1	3e-17	1(4)	2e-12	3e-17	2	3e-17	1(11)
sherman5	7e-18	2e-14	3e-18	2	3e-18	2(52)	4e-8	3e-18	17	7e-18	31(215)
watt1	1e-22	2e-16	4e-23	15	3e-22	1(1)	5e-17	1e-22	2	3e-22	1(1)
watt2	5e-18	2e-16	4e-18	26	5e-17	2(2)	3e-15	3e-19	125	5e-14	83(125)

2: $res2 = 1e-17$, if $\epsilon = 1e-12$; 3: $res2 = 1e-17$, if $\epsilon = 1e-12$;