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 $O(u||A|||x||)$. 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 $n = 1, 2, \ldots$ 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

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

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

where $f(z)$ denotes the computed result of $z$ in finite arithmetic, the absolute value and inequalities on vectors are componentwise, and $u$ is the machine roundoff unit. The vectors $\psi_n$ and $\eta_n$ 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 $\psi_n$ and $\eta_n$ depend only on the iteration vectors $\hat{x}_n, \hat{r}_n$, and $q_n$.

We will call $b - Ax_n$ 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))

$$
b - A\hat{x}_n - \hat{r}_n = b - Ax_{n-1} - \hat{r}_{n-1} - A\psi_n - \eta_n
$$

$$
= -\sum_{i=1}^{n} (A\psi_i + \eta_i),
$$

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 $\psi_i$ or $\eta_i$ are small (as is common for CG), but if it takes a relatively large number of iterations for convergence, the shear accumulation of $\psi_i$ and $\eta_i$ 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 $u$, 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 \( u \) 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 \( b - A\hat{x}_n \approx \Sigma_{i=1}^n (A\psi_i + \eta_i) \) 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 \( \psi_i \) or \( \eta_i \) 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 \( \psi_n \) or \( \eta_n \). Several popular methods, such as BiCGSTAB [22], BiCGSTAB(\( \ell \)) [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 \( O(u\|A\|\|x\|) \), 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 $\infty$-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 $r_n$ 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 $f(l(b - A\hat{x}_m))$ at some selected steps $m = m_1, m_2, \ldots, m_k$. 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 $\hat{r}_n = f(l(b - A\hat{x}_n))$ or the true residual $f(l(b - A\hat{x}_n))$, 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

$$r_n = \begin{cases} f(l(b - A\hat{x}_n)), & \text{if } n = m_1, m_2, \ldots, m_k \\ \hat{r}_n = f(l(\hat{r}_{n-1} - A\hat{q}_{n-1})), & \text{otherwise} \end{cases}$$

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

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

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

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

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 $u(|r_m| + N|A||\hat{x}_m|)$, which is the upper bound 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 \( \tilde{r}_n \). 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(\( \ell \)) [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 finite arithmetic, the perturbed recurrence

\[
\tilde{r}_n = \tilde{r}_{n-1} - \alpha_{n-1} \tilde{p}_{n-1} + \eta_n \quad \text{and} \quad \tilde{p}_n = \tilde{r}_n + \beta_n \tilde{p}_{n-1} + \tau_n,
\]

where \( \eta_n \) and \( \tau_n \) are rounding error terms that can be bounded in terms of \( u \). Combining these two equations, we obtain the following perturbed matrix equation in a normalized form

\[
AZ_n = Z_n T_n - \frac{1}{\alpha_n} \tilde{r}_n^T \tilde{r}_n + F_n \quad \text{with} \quad Z_n = \begin{bmatrix} \tilde{r}_1, \ldots, & \tilde{r}_n \end{bmatrix},
\]

where \( T_n \) is an invertible tridiagonal matrix\(^1\), \( \alpha'_n = \| \tilde{r}_n \| \| \tilde{r}_1 \| = \tilde{r}_n^T T_n^{-1} \tilde{r}_1 \) and \( F_n = [f_1, \ldots, f_n] \) with

\[
f_n = -\frac{A \tilde{r}_n}{\| \tilde{r}_n \|} + \frac{1}{\alpha_n} \frac{\eta_n}{\| \tilde{r}_n \|} - \frac{\beta_n}{\alpha_{n-1}} \frac{\eta_n}{\| \tilde{r}_n \|}.
\]

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 \( \tau_n \) and \( \eta_n \) will, under some mild assumptions, lead to \( F_n \sim O(u) \).

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

\[
\| \tilde{r}_{n+1} \| \leq (1 + K_n) \min_{p \in F_{n+1}(0) \neq 0} \| p(A + \Delta A) \tilde{r}_n \|,
\]

where \( K_n = \| (AZ_n - F_n) T_n^{-1} \| \| Z_{n+1}^+ \| \) and \( \Delta A_n = -F_n Z_n^+ \). The case \( F_n = 0 \) reduces to the known theoretical bound for the exact BiCG residuals [1]. Therefore, even when \( \tilde{r}_n \) 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 \( u \), the quantities in the bound, and thus \( \| \tilde{r}_{n+1} \| \), 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 \( \tilde{r}_n \) is generated but as long as it satisfies (4), its norm can be bounded by (6). Hence, we can replace \( \tilde{r}_n \) by \( r_n = fl(b - Ax_n) \)

\[^1\text{We assume that no breakdowns of the iteration process have occurred.}\]
when \( \eta_n = r_n - (r_{n-1} - Aq_m) \) 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 \( r_n = f(\hat{b} - Ax_n) \) 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 \( r_n = f(\hat{b} - Ax_n) \). 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 \( r_m = f(\hat{b} - Ax_m) = \frac{b - A\hat{x}_m + \xi_m}{r_m - Aq_{m-1} + \eta_m + \delta_m} \). It follows from the definition of \( \delta_m \) 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

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

Thus depending on the magnitude of \( \|\eta'_m\| \) relative to \( \|r_m\| \) and \( \|r_{m-1}\| \), the use of \( r_m = f(\hat{b} - Ax_m) \) 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, \( \|\delta_m\| \), and hence \( \|\eta'_m\| \), increases while \( \|\hat{r}_n\| \) decreases. Replacement will reduce \( \delta_m \) but, in order to maintain the recurrence relation, it should be carried out before \( \|\eta'_m\| \) becomes too large relative to \( \|\hat{r}_n\| \). For this reason, we propose to set a threshold \( \epsilon \) and carry out a replacement when \( \|\eta'_m\|/\|\hat{r}_n\| \) reaches the threshold. To be precise, we replace the residual at step \( n \) by \( r_n = \hat{b} - Ax_n \), if

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

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 \( u \)).

### 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 \( u|A||x_n| \), in which case \( x_n \) is a backward stable solution. Note that, for any \( x_n \), \( u|A||x_n| \) is the lowest value one can expect for its residual. This is simply because even with the exact solution \( x \), both \( b - A(f(\hat{x})) \) and \( f(\hat{b} - Ax) \sim u|A||x| \).

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

\[
    \delta_n = -\xi_m - \sum_{i=m+1}^n \eta_i - \sum_{i=m+1}^n A\psi_i.
\]

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

- \( |\xi_m| \leq u|\delta_m| + N|A||\hat{x}_m| \sim O(u)|A||x| \).
• Since $|\hat{r}_n^j| < |b| \leq ||A||x|$ and $|\eta_n| \sim u|\hat{r}_n|$ we have that $\sum_{i=m+1}^{n} |\eta_n| << O(u||A||x)$.

• For the $\psi_n$ part, $|\psi_n| \sim u|\hat{x}_n| \sim u|x|$. Hence, $\sum_{i=m+1}^{n} |\psi_n| \sim \sum_{i=m+1}^{n} u||A|||x|| = (n-m)u||A|||x||$. If $n-m$ is large, the accumulation of errors over $n-m$ steps can be significant. We note this is the same type of error accumulation in evaluating a sum $S = \sum_{i=1}^{n} c_i$ of small numbers by direct recursive additions, which can be corrected by appropriately grouping the arithmetic operations as $S_1 + S_2 + \cdots = (c_1 + \cdots + c_m) + (c_{m+1} + \cdots + c_{2m}) + \cdots$ 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 $uS_i$, which can be much smaller than $uS$.

The same technique can be adopted for computing $x_n$ as

$$x_n = x_0 + \sum_{i=1}^{n} \hat{g}_i = x_0 + (q_1 + \cdots + q_{m1}) + (q_{m1+1} + \cdots + q_{m2}) + \cdots$$

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

**Groupwise Solution Update**: $z = x_0^g; \hat{x}_0 = 0$;

For $n = 1, 2, \cdots$ until convergence

\[
\hat{x}_n = f(\hat{x}_{n-1} + q_n) = \hat{x}_{n-1} + q_n + \psi_n
\]

if $n = m$ (i.e. **group update**)

\[
z = fI(z + \hat{x}_n) = z + \hat{x}_n + \zeta_n
\]

\[
\hat{x}_n = 0
\]

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 $i \geq m$, $|\hat{x}_i| = |z + \hat{x}_i - z| \sim |x - z| << |x|$. Then $|\psi_n| \sim u|\hat{x}_i|$ (instead of $u|x_i|$). Hence, $\sum_{i=m+1}^{n} |\psi_n| << (n-m)u||A|||x||$.

In summary, with groupwise updating of the approximated solution, all three parts of $\delta_n$ can be maintained at the level of $u||A|||x||$. 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 after every $\ell$ 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 $|\xi_n| \sim u||A||x_n||$, 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 $\xi_n$ 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 $\xi_n$, and hence $\eta_n$, at a replacement step. In the flying-restart
strategy \( b \) is replaced by \( f \ell(b - Ax_m) \) at some but not all of the residual replacement steps (say \( m = n_1, n_2, \ldots, n_t \)), in addition to the residual replacement \( r_m = f \ell(b - Ax_m) \). The advantage of this is that, at the flying-restart step \( n_{i+1} \), the residual is updated by \( r_{n_{i+1}} = f \ell(r_n - A\hat{x}_{n_{i+1}}) = r_n - A\hat{x}_{n_{i+1}} + \xi_{n_{i+1}} \) (noting that \( b \leftarrow r_n \)) and \( |\xi_{n_{i+1}}| \sim u(|r_n| + |A||\hat{x}_{n_{i+1}}|) \). Then

\[
|r_n - A\hat{x}_{n_{i+1}} - r_{n_{i+1}}| = |\xi_{n_{i+1}}| \sim u(|r_n| + |A||\hat{x}_{n_{i+1}}|)
\]

which decreases as \( r_n \) 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

\[
b - Ax_{n_{i+1}} - r_{n_{i+1}} = b - A\hat{x}_{n_{i+1}} - r_n - \xi_{n_{i+1}},
\]

(assuming \( x_{n_{i+1}} = x_n + \hat{x}_{n_{i+1}} \)). 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 \( \xi_n \sim u(||b|| + N||A||||x||) \). 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 \( \hat{x}_0 = 0 \); \( r_0 = f \ell(b - Ax_0) = b - Ax_0 + \xi_0 \);
For \( n = 1, 2, \ldots \), until convergence

- Generate a correction vector \( q_n \) by the method;
- \( \hat{x}_n = f \ell(\hat{x}_{n-1} + q_n) = \hat{x}_{n-1} + q_n + \psi_n \)
- \( \hat{r}_n = f \ell(r_{n-1} - Aq_n) = r_{n-1} - Aq_n + \eta_n \)

if residual replacement condition (8) holds

- \( z = f \ell(z + \hat{x}_n) = z + \hat{x}_n + \zeta_n \)
- \( \hat{x}_n = 0 \)
- \( r_n = f \ell(b - Az) = b - Az + \xi_n \)

else

- \( r_n = \hat{r}_n \)
end if

(\( \text{denote but not compute} \ x_n = z + \hat{x}_n \) and \( \delta_n = b - A\hat{x}_n - r_n \))

End for

\[
z = f \ell(z + \hat{x}_n) = z + \hat{x}_n + \zeta_n
\]
Note that \( x_n \) and \( \delta_n \) 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 \([8, \text{p.66}]\) (all inequalities are componentwise).

\[
\begin{align*}
  f(x + y) &= x + y + \epsilon \quad \text{with} \quad |\epsilon| \leq u(|x + y|) \tag{9} \\
  f(Ax) &= Ax + \epsilon \quad \text{with} \quad |\epsilon| \leq uN|A||x| + O(u^2). \tag{10}
\end{align*}
\]

where \( x, y \in \mathbb{R}^N \) 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
\[
  f(l(y + Ax)) = y + Ax + \epsilon \quad \text{with} \quad |\epsilon| \leq u(|y + Ax| + N|A||x|) + O(u^2).
\]

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:

\[
\begin{align*}
  |\psi_n| &\leq u|x_n| + O(u^2) \tag{11} \\
  |\eta_n| &\leq u|x_n| + N|A||q_n| + O(u^2). \tag{12}
\end{align*}
\]

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

\[
\begin{align*}
  |\zeta_n| &\leq u|x_n| + O(u^2), \tag{13} \\
  |\xi_n| &\leq u|y_n| + N|A||x_n| + O(u^2). \tag{14}
\end{align*}
\]

**Proof** From (9), we have that \( |\psi_n| \leq u|\hat{x}_{n-1} + q_n| \leq u(|\hat{x}_n| + |\psi_n|). \) 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, \( |\zeta_n| \leq u|x_n| + O(u^2) \). The bounds for \( \eta_n \) and \( \xi_n \) follow similarly. \( \square \)

**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

\[
\begin{align*}
  \sum_{i=m+1}^n |\psi_i| &\leq u\sum_{i=m+1}^n |\hat{x}_i| + O(u^2), \\
  \sum_{i=m+1}^n |\eta_i| &\leq (2 + u)\sum_{i=m+1}^n |\hat{x}_i| \\
  \sum_{i=m+1}^n |\eta_i| &\leq u\sum_{i=m+1}^n |\hat{x}_i| + 2uN|A|\sum_{i=m+1}^n |\hat{x}_i| + O(u^2).
\end{align*}
\]
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
\[
\sum_{i=m+1}^n |q_i| \leq \sum_{i=m+1}^n (|\hat{x}_i| + |\hat{x}_{i-1}| + |\psi_i|) \leq (2 + u)\sum_{i=m+1}^n |\hat{x}_i|.
\]
Similarly,
\[
\sum_{i=m+1}^n |q_i| \leq u\sum_{i=m+1}^n (|\hat{r}_i| + N|A||q_i|) + O(u^2)
\leq u\sum_{i=m+1}^n |\hat{r}_i| + 2uN|A|\sum_{i=m+1}^n |\hat{x}_i| + O(u^2).
\]

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

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, $r_m = f_l(b - Ax_m)$, and $r_m = b - Ax_m + \xi_m$. Therefore $\delta_m = -\xi_m$. Hence, for the range of $n > m$, and before the next residual replacement step:
\[
\begin{align*}
\delta_n &= b - Ax_n - r_n = b - A(z + \hat{x}_n) - \hat{r}_n \\
&= b - A(z + \hat{x}_{n-1} + q_n + \psi_n) - (\hat{r}_{n-1} - Aq_n + \eta_n) \\
&= \delta_{n-1} - A\psi_n - \eta_n \\
&= \delta_m - \sum_{i=m+1}^n (A\psi_i + \eta_i).
\end{align*}
\]

With Lemma 2, we obtain the following computable bound on $\delta_n$.

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 $\|\delta_m\| \leq u(\|r_m\| + N\|A\|\|x_m\|) + O(u^2)$ and
\[
\|\delta_n\| \leq uN\|A\|\|x_m\| + u(1 + 2N)\|A\|\sum_{i=m+1}^n \|\hat{x}_i\| + u\sum_{i=m+1}^n \|r_i\| + O(u^2).
\]

Proof The bound for $\|\delta_n\|$ follows from that for $\xi_m$, see (14). From Lemma 2 and Lemma 3, it follows that
\[
\begin{align*}
|\delta_n| &\leq |\xi_m| + \sum_{i=m+1}^n (|A||\psi_i| + |\eta_i|) \\
&\leq u(|r_m| + N\|A\|\|x_m\|) + |A|u\sum_{i=m+1}^n \|\hat{x}_i\| \\
& \quad + u\sum_{i=m+1}^n |\hat{r}_i| + 2uN\|A\|\sum_{i=m+1}^n |\hat{x}_i| + O(u^2) \\
&= uN\|A\|\|x_m\| + u(1 + 2N)\|A\|\sum_{i=m+1}^n \|\hat{x}_i\| + u\sum_{i=m+1}^n \|r_i\| + O(u^2),
\end{align*}
\]
which leads to the bound for $\delta_n$ in terms of norms.

\[\square\]
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 $x_n' = f(x_m + \hat{x}_n) = x_m + \hat{x}_n + \zeta_n$ be the computed approximate solution and $r_n' = f(b - Ax_n') = b - Ax_n' + \xi_n$ be the residual. Then the residual $r_n'$ satisfies the following approximate recurrence

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

where $\eta_n' = \eta_n + \delta_n - AC_n + \xi_n$ and

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

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

$$r_n' = b - Ax_n - AC_n + \xi_n = r_n + \delta_n - AC_n + \xi_n = r_{n-1} - Aq_n + \eta_n + \delta_n - AC_n + \xi_n = r_{n-1} - Aq_n + \eta_n',
$$

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,

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

Also, $\|AC_n\| \leq u\|A\|\|x_m\| + \|\hat{x}_n\|$, and $\|\xi_n\| \leq u\|\|r_n\| + N\|A\|\|x_n\|\|r_n\| + O(u^2) \leq u\|\|r_n\| + N\|A\|\|x_m\| + N\|A\|\|\hat{x}_n\|\| + O(u^2)$. Combining these three, and using that $r_n' = r_n + O(u)$, 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 $q_i \sim \hat{x}_i$ 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 $\|\eta_n'\|$:

$$d_n \equiv uN\|A\|\|x_m\| + \sum_{i=m+1}^{n}\|\hat{x}_i\| + u\sum_{i=m}^{n}\|r_i\|.
$$

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

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

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 $\|r_K\| < u\|A\|\|x_K\|$. Let $m$ be the number of the last residual replacement iteration step before termination. If

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

then
\[
\|b - Ax_K\| \leq \|r_K\| + uN\|A\|\|x_K\|/(1 - uL) + O(u^2)
\]

\[
\sim uN\|A\|\|x_K\|.
\]

**Proof** From (17), we have \(d_K > uN\|A\|\|x_K\| + \|\hat{x}_K\| \geq u\|A\|\|x_K\|\). Furthermore, at the termination step, we have \(\|r_K\| < u\|A\|\|x_K\|\) 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

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

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

\[
\|\hat{x}_n\| = \|x_n - x_m\| = \|A^{-1}((b - Ax_m) - (b - Ax_n))\|
\]

\[
= \|A^{-1}(r_m + \delta_m - r_n - \delta_n)\|
\]

\[
\leq \|A^{-1}\|\|(r_m + ||r_n|| + ||\delta_n - \delta_m||)
\]

\[
\leq \|A^{-1}\|(1 + 2/\epsilon)\hat{d}_K + O(u^2),
\]

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

\[
d_K = uN\|A\|\|x_K - \hat{x}_K\| + u(1 + 2N)\|A\|\Sigma_{i=m+1}^n\|\hat{x}_i\| + u\Sigma_{i=m}^n\|\hat{r}_i\|
\]

\[
\leq uN\|A\|\|x_K\| + uN\|A\|\|\hat{x}_K\| + u(1 + 2N)\|A\|\Sigma_{i=m+1}^n\|\hat{x}_i\| + u\Sigma_{i=m}^n\|\hat{r}_i\|
\]

\[
\leq uN\|A\|\|x_K\| + uK - m + 1)(1 + 2N)\|A\|\|A^{-1}\|(1 + 2/\epsilon)\hat{d}_K
\]

\[
+ u(1 + 2/\epsilon)\hat{d}_K + O(u^2)
\]

\[
\leq uN\|A\|\|x_K\| + uK - m + 1)(1 + 2N)\|A\|\|A^{-1}\|(1 + 3/\epsilon)\hat{d}_K + O(u^2)
\]

\[
\leq uN\|A\|\|x_K\| + uL\hat{d}_K + O(u^2),
\]

which implies

\[
\hat{d}_K \leq uN\|A\|\|x_K\|/(1 - uL) + O(u^2).
\]

Thus the bound follows from

\[
\|b - Ax_K\| \leq \|r_K\| + \|\delta_K\| \leq \|r_K\| + \hat{d}_K + O(u^2).
\]

\[\Box\]

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 \(uN\|A\|\|x_K\|\) at termination. Such an approximate solution is backward stable and it is best one can expect. The condition suggests that \(\epsilon\) 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 \(\epsilon\) 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 \(\epsilon\) 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 \(\epsilon\) around \(\sqrt{u}\) 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 \(u(N\|A\||x_n|| + \|r_m||)\). 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 \(u(N\|A\||x_K||\) 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 \(d_n/\epsilon \sim u(N\|A\||x_n||/\epsilon\). 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 \(d_n > 1.1d_m\) 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 \(\epsilon\); an estimate of \(N\|A\|\);
Set \(r_0 = b - Ax_0\); \(\delta_0 = 0\); \(d_{init} = d_0 = u(\|r_0\|| + N\|A\||\|z_0||),\)
For \(n = 1, 2, \ldots\), until convergence
Generate a correction vector \(q_n\) by the Iterative Method;
\(\hat{x}_n = \hat{x}_{n-1} + q_n\)
\(r_n = r_{n-1} - Aq_n\)
\(d_n = d_{n-1} + uN\|A\||\|\hat{x}_n|| + u\|r_n||\)
if \(d_{n-1} \leq \epsilon\|r_{n-1}||, \ d_n > \epsilon\|r_n|| and d_n > 1.1d_{init}\)
\(z = z + \delta_n\)
\(\hat{x}_n = 0\)
\(r_n = b - Az\)
\(d_{init} = d_n = u(\|r_n\|| + N\|A\||\|z||)\)
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 \(\epsilon\) can be adjusted. We also need to choose this \(\epsilon\). Our extensive numerical testing (see section 5) suggests that \(\epsilon \sim \sqrt{u}\) 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 \(\epsilon\) will regain the convergence to \(O(u)\).

The presented implementation requires one extra matrix-vector multiplication when a 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 \(\epsilon\) 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 up-
date (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 Spare-20 workstation, with $u \approx 10^{-16}$.

In all examples, unless otherwise specified, the replacement threshold $\epsilon$ 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 $\| \hat{d}_n \|$.  

Example 1: The matrix is a finite-difference discretization on a $64 \times 64$ 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 $f(x,y) = x^2y$. 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 $\| \hat{d}_n \| = \| b - Ax_n - r_n \|$ 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 $64 \times 64$ grid for the following convection diffusion equation
\[- \Delta u + \gamma(xu_x + yu_y) + \beta u = f(x,y) \text{ on } (0,1)^2;\]
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 $\| u \|_A \| x \|$ and $\| \hat{d}_n \|$. We also note that the bound $d_n$ captures the behaviour of $\| \hat{d}_n \|$ very closely, although it may be an overestimate for $\hat{d}_n$ 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 $\| u \|_A \| x \|$ so that there is a deviation of the two residuals. For those cases where $b$ is not given, we choose it such that a
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 $O(u)$ with the choice of $\epsilon = 1e-8$. For those cases, a slightly smaller $\epsilon$ 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 $O(u)$ 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.
6 Concluding Remarks

We have presented a new residual replacement scheme for improving the convergence of the true residuals in finite precision implementations of Krylov 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


Table 1: Example 3: Comparison of Normalized Residuals

<table>
<thead>
<tr>
<th>Matrix</th>
<th>res0</th>
<th>res1</th>
<th>res2, n_r</th>
<th>res3, n_f (n_r)</th>
<th>CGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>bescpw06</td>
<td>7e-15</td>
<td>1e-20</td>
<td>1e-19</td>
<td>5(9)</td>
<td>6e-13</td>
</tr>
<tr>
<td>bescpw07</td>
<td>1e-15</td>
<td>2e-17</td>
<td>9e-17</td>
<td>2(6)</td>
<td>2e-12</td>
</tr>
<tr>
<td>bescpw08</td>
<td>2e-15</td>
<td>3e-17</td>
<td>1e-16</td>
<td>5(7)</td>
<td>7e-14</td>
</tr>
<tr>
<td>bescpw09</td>
<td>3e-15</td>
<td>2e-20</td>
<td>6e-20</td>
<td>5(66)</td>
<td>3e-13</td>
</tr>
<tr>
<td>jpw991</td>
<td>1e-16</td>
<td>9e-17</td>
<td>3e-17</td>
<td>1(1)</td>
<td>7e-17</td>
</tr>
<tr>
<td>fs6801</td>
<td>1e-17</td>
<td>7e-17</td>
<td>1e-17</td>
<td>2(1)</td>
<td>8e-18</td>
</tr>
<tr>
<td>fs6802</td>
<td>8e-18</td>
<td>1e-16</td>
<td>8e-18</td>
<td>3(1)</td>
<td>2e-17</td>
</tr>
<tr>
<td>fs6803</td>
<td>6e-18</td>
<td>3e-16</td>
<td>1e-13</td>
<td>11(5)</td>
<td>4e-14</td>
</tr>
<tr>
<td>fs7601</td>
<td>7e-17</td>
<td>7e-17</td>
<td>9e-18</td>
<td>1(1)</td>
<td>5e-15</td>
</tr>
<tr>
<td>jagmesh1</td>
<td>3e-16</td>
<td>4e-15</td>
<td>5e-17</td>
<td>2(1)</td>
<td>1e-12</td>
</tr>
<tr>
<td>nos3</td>
<td>1e-16</td>
<td>3e-16</td>
<td>6e-17</td>
<td>2(1)</td>
<td>6e-17</td>
</tr>
<tr>
<td>nos4</td>
<td>8e-17</td>
<td>2e-16</td>
<td>5e-17</td>
<td>1(1)</td>
<td>6e-17</td>
</tr>
<tr>
<td>nos5</td>
<td>1e-16</td>
<td>3e-16</td>
<td>5e-17</td>
<td>2(1)</td>
<td>6e-17</td>
</tr>
<tr>
<td>nos6</td>
<td>6e-17</td>
<td>4e-16</td>
<td>3e-17</td>
<td>9(1)</td>
<td>8e-17</td>
</tr>
<tr>
<td>1138bas</td>
<td>9e-18</td>
<td>2e-16</td>
<td>1e-17</td>
<td>8(1)</td>
<td>9e-17</td>
</tr>
<tr>
<td>1sirr2</td>
<td>4e-17</td>
<td>1e-17</td>
<td>3(4)</td>
<td>5(6)</td>
<td>5e-14</td>
</tr>
<tr>
<td>1sirr3</td>
<td>2e-16</td>
<td>7e-17</td>
<td>1(1)</td>
<td>7e-15</td>
<td>8e-17</td>
</tr>
<tr>
<td>1sirr4</td>
<td>3e-17</td>
<td>2e-17</td>
<td>3e-17</td>
<td>2(3)</td>
<td>5e-15</td>
</tr>
<tr>
<td>pores5</td>
<td>3e-17</td>
<td>2e-17</td>
<td>3e-17</td>
<td>4(5)</td>
<td>2e-12</td>
</tr>
<tr>
<td>saylr3</td>
<td>3e-17</td>
<td>3e-17</td>
<td>3e-17</td>
<td>1(1)</td>
<td>2e-16</td>
</tr>
<tr>
<td>saylr4</td>
<td>1e-15</td>
<td>8e-17</td>
<td>5e-16</td>
<td>1(1)</td>
<td>2e-15</td>
</tr>
<tr>
<td>sherman1</td>
<td>5e-17</td>
<td>3e-17</td>
<td>3e-17</td>
<td>2(1)</td>
<td>2e-10</td>
</tr>
<tr>
<td>sherman3</td>
<td>6e-19</td>
<td>2e-17</td>
<td>4e-19</td>
<td>9(1)</td>
<td>1e-18</td>
</tr>
<tr>
<td>sherman4</td>
<td>6e-17</td>
<td>2e-17</td>
<td>3e-17</td>
<td>1(1)</td>
<td>2e-12</td>
</tr>
<tr>
<td>sherman5</td>
<td>6e-18</td>
<td>2e-17</td>
<td>4e-18</td>
<td>2(1)</td>
<td>4e-8</td>
</tr>
<tr>
<td>watt1</td>
<td>1e-22</td>
<td>2e-16</td>
<td>4e-23</td>
<td>1(1)</td>
<td>5e-17</td>
</tr>
<tr>
<td>watt2</td>
<td>5e-18</td>
<td>2e-16</td>
<td>4e-18</td>
<td>26(2)</td>
<td>5e-17</td>
</tr>
</tbody>
</table>

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