Accuracy of Two Three-term and Three Two-term Recurrences for Krylov Space Solvers

Gutknecht, M. H.
1997

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Accuracy of Two Three-term and Three Two-term Recurrences for Krylov Space Solvers

Martin H. Gutknecht Christopher C. Paige Zdeněk Strakoš

Technical report No. 764

September 1997

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic
phone: (+4202) 6884244 fax: (+4202) 8585789
e-mail: strakos@uivt.cas.cz

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Accuracy of Two Three-term and Three Two-term Recurrences for Krylov Space Solvers[1]

Martin H. Gutknecht[2] Christopher C. Paige[3]
Zdeněk Strakoš[4]

Technical report No. 764
September 1997

Abstract

It has been widely observed that Krylov space solvers based on two three-term recurrences can give significantly less accurate residuals than mathematically equivalent solvers implemented with three two-term recurrences. In this paper we attempt to justify this difference theoretically by analyzing the gap between the recursively and the explicitly computed residuals. It is shown that, in contrast with the two-term recurrences analyzed by Greenbaum (*SIAM J. Matrix Anal. Appl., 18 (1997), pp. 535–551*), in the two three-term recurrences the local roundoff error contributions to the analyzed gap may dramatically amplify while propagating through the algorithm. For the conjugate gradient method, such a devastating behavior is, however, not observed frequently in practical computations, where the difference between three two-term and two three-term implementations is usually moderate or small. This can be explained by our results. We emphasize that in general there is no inherent weakness in the three term recurrence for the residual, the difficulty occurs when the iterate is also computed via a three term recurrence.

Keywords

Linear system of equations, iterative method, Krylov space method, conjugate gradient method, three-term recurrence, accuracy, roundoff

[1]This work was supported by the ASCR Grant A2030706 and by the GA CR Grant 205/96/0921. Part of the work was performed while the third author visited the Swiss Center for Scientific Computing (CSCS/SCSC).

[2]Swiss Center for Scientific Computing, ETH-Zentrum, CH-8092 Zürich, Switzerland, E-mail: mhg@scsc.ethz.ch

[3]School of School of Computer Science, McGill University, Montreal, Quebec, Canada H3A 2A7, E-mail: chris@cs.mcgill.ca

[4]Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2, 182 07 Praha 8, Czech Republic

1 Introduction

Among the Krylov space solvers for linear systems $Ax = b$ (with A an $N \times N$ nonsingular matrix and b an N-vector) there are quite a number that are based on three-term recurrences for both the *residuals* r_n and the *iterates* x_n (we give full derivations in the next section). With x_0 given, $r_0 := b - Ax_0$, and while $\gamma_n \neq 0$,

$$r_1 := (Ar_0 - r_0\alpha_0)/\gamma_0\,; \qquad r_{n+1} := (Ar_n - r_n\alpha_n - r_{n-1}\beta_{n-1})/\gamma_n\,,\ n = 1, 2, \ldots\,, \quad (1.1)$$

$$x_1 := -(r_0 - x_0\alpha_0)/\gamma_0\,; \qquad x_{n+1} := -(r_n + x_n\alpha_n + x_{n-1}\beta_{n-1})/\gamma_n\,,\ n = 1, 2, \ldots\,. \quad (1.2)$$

The above recurrences will only ensure (see for example § 4.3 of [13], and Section 2 here)

$$r_n = b - Ax_n\,, \quad n = 0, 1, \ldots \quad (1.3)$$

if the scaling coefficients γ_n are chosen to satisfy

$$\gamma_0 := -\alpha_0\,; \quad \gamma_n := -(\alpha_n + \beta_{n-1}), \quad n = 1, 2, \ldots\,. \quad (1.4)$$

The analysis will hold for any implementation using (1.1), (1.2) and (1.4), whether A is symmetric or not. Some such methods for unsymmetric problems may use other recurrences as well, but for brevity, here we will refer to all such methods as "two three-term recurrence methods". In particular, the list of such algorithms includes the Chebyshev iteration [26, 24, 18], the second-order Richardson iteration [24] (which is the stationary form of the Chebyshev iteration), the three-term version of the conjugate-gradient (CG) method [26, 2, 15], and the three-term version (BIORES) of the unsymmetric or two-sided Lanczos method [17, 13] (which is a variation of the biconjugate gradient or BICG method); see also [15].

CG and BICG have better known versions that are based on three two-term recursions which involve in addition to the iterates and their residuals also direction vectors p_n:

$$r_{n+1} := r_n - Ap_n\omega_n\,, \quad (1.5)$$

$$x_{n+1} := x_n + p_n\omega_n\,, \quad (1.6)$$

$$p_{n+1} := r_{n+1} + p_n\psi_n\,, \quad (1.7)$$

for $n \geq 0$, with $p_0 := r_0$. Other methods like ORTHOMIN [27] use the first two of these recursions, but have a more complex update formula for the direction vectors. The version (1.5)–(1.7) can be obtained from the version (1.1)–(1.2) by an LU decomposition of the tridiagonal matrix with coefficients β_{n-1}, α_n, and γ_n in the $(n+1)$st column, see for example [4, 13] and Section 2 here. The folklore — confirmed by many experiments — is that implementations based on three two-term recursions are less affected by roundoff than the same methods based on two three-term recursions. We will analyze the extent to which this is true.

A recent result by Greenbaum [9, 10] shows that under the sole assumption that the first two recursions (1.5), (1.6) hold, there is a limitation on the accuracy of the iterates computed in finite precision arithmetic, and the corresponding values $b - Ax_n$ do not decrease below a certain level. (A similar, but somewhat weaker result was given

by Sleijpen, van der Vorst, and Fokkema [25].) This maximum expectable accuracy depends primarily on the largest norm of an approximate solution x_n that has been generated during the iteration, but it does not depend directly on how the coefficients ω_n and ψ_n are determined. Since, for example, the BiCG method may produce very large intermediate iterates and residuals, this result is of great importance in practice. In contrast, related work on GMRES showed that the size of intermediate iterates does not play a role [3, 11]. In this paper we investigate and answer the question why methods based on recursions of the form (1.1)–(1.2) often produce less accurate residuals than those based on the form (1.5)–(1.7). Note this says nothing of methods which use the equivalent of the first recurrence (1.1), but compute the iterates some other way than (1.2). We show there is no inherent weakness in (1.1) alone.

It should be noticed that iterative methods based on (1.1)–(1.2) or (1.5)–(1.7) typically produce recursively computed residuals r_n whose norms eventually decrease even beyond the norm of the roundoff occurring in finite precision arithmetic when the exact solution x is inserted into $b - Ax$. This means that from a certain n on, these residuals have nothing to do with the true residuals.

2 Background Theory

In this paper we will use both the vector and matrix forms of certain equations. We use the vector forms to understand iterative computations and to derive their local rounding error effects, while we use the equivalent matrix forms to show for example how three two-term recurrences can be obtained from two three-term recurrences and vice versa, and later to derive and understand global effects of local rounding errors.

This paragraph needs fixing when we know what we've put in. It would be possible to carry through the error analysis for the residual error using vector forms only, by subtle use of difference equations, see [14]. Readers with a background in polynomials or ordinary differential equations may be interested in that reference. However the approach based on matrix forms can be applied with minimal further effort in other situations, as is done for examining the errors in the iterates here, so we use it here. It will help to cast everything in matrix terms, so we develop the theory in these terms to make the paper as self-contained as possible.

2.1 Deriving three-term recurrences for both r_n and x_n

Given an initial approximation x_0, the initial residual is computed via

$$r_0 := b - Ax_0. \tag{2.1}$$

The general three-term Krylov recurrence for the residuals is (1.1), so if we define the matrices

$$X_n \equiv [x_0, \ldots, x_n], \qquad R_n \equiv [r_0, \ldots, r_n], \qquad T_{n+1,n} \equiv \begin{bmatrix} \alpha_0 & \beta_0 & & \\ \gamma_0 & \alpha_1 & \cdot & \\ & \gamma_1 & \cdot & \beta_{n-1} \\ & & \cdot & \alpha_n \\ & & & \gamma_n \end{bmatrix}, \tag{2.2}$$

we see (1.1) corresponds to the columns of

$$AR_n = R_{n+1}T_{n+1,n}. \tag{2.3}$$

But then (1.3) gives, with e the vector of ones of the required dimension,

$$R_n = be^T - AX_n = A^{-1}R_{n+1}T_{n+1,n} = (xe^T - X_{n+1})T_{n+1,n}. \tag{2.4}$$

Since x is unknown, each column of this gives a consistent and computable recurrence for x_n, $n = 1, 2, \ldots$ if and only if the x term in (2.4) disappears. That is, if and only if

$$e^T T_{n+1,n} = 0^T, \tag{2.5}$$

which is just (1.4), and gives for (2.4)

$$R_n + X_{n+1}T_{n+1,n} = 0, \tag{2.6}$$

whose columns show the derivation of (1.2), (see also Section 4.3 of [14], for example).

If (2.5) does not hold, we could still drop the term in (2.4) involving x, giving an inconsistent iteration, see Section 4.5 of [13], but we will not consider this here.

We will see the residual relation (1.3) is implied by (2.1), (2.3), (2.5) and (2.6). This implication leads to the main weakness of using two three-term recurrences in finite precision, so we derive it here. Since $\gamma_i \neq 0$, $T_{n+1,n}$ in (2.2) has full column rank, so (2.5) implies the left null space of $T_{n+1,n}$ is span(e^T). Now (2.3) and (2.6) give

$$(AX_{n+1} + R_{n+1})T_{n+1,n} = 0, \tag{2.7}$$

so $AX_{n+1} + R_{n+1} = ce^T$ for some vector c. Multiplying this on the right by $e_0 \equiv [1, 0, \ldots, 0]^T$ gives $Ax_0 + r_0 = c$, so $c = b$ from (2.1), and

$$R_{n+1} = be^T - AX_{n+1}, \tag{2.8}$$

which is (1.3). Note (2.7) gives an *implicit* definition of the residual relation. We will show because of this the finite precision residual can be poor, depending on the properties of $T_{n+1,n}$.

2.2 Deriving three two-term recurrences from two three-term recurrences

Algorithm (1.5)–(1.7) was introduced as the three two-term recurrence variant of CG, but here we show that it gives the three two-term recurrence variant of any two three-term recurrence (1.1)–(1.2). Define

$$\begin{aligned} & L_{n+1,n}D_nU_n \qquad\qquad (2.9)\\ & \equiv \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & \cdot & \cdot & & \\ & & -1 & 1 & \\ & & & -1 & \end{bmatrix} \begin{bmatrix} -\gamma_0 & & & \\ & \cdot & & \\ & & \cdot & \\ & & & -\gamma_n \end{bmatrix} \begin{bmatrix} 1 & -\beta_0/\gamma_0 & & \\ & \cdot & \cdot & \\ & & 1 & -\beta_{n-1}/\gamma_{n-1} \\ & & & 1 \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \cdot & \cdot & \\ & & -1 & 1 \\ & & & -1 \end{bmatrix} \begin{bmatrix} -\gamma_0 & \beta_0 & & \\ & \cdot & \cdot & \\ & & -\gamma_{n-1} & \beta_{n-1} \\ & & & -\gamma_n \end{bmatrix}$$

$$= \begin{bmatrix} -\gamma_0 & \beta_0 & & & \\ \gamma_0 & -\beta_0 - \gamma_1 & \beta_1 & & \\ & \cdot & \cdot & \cdot & \\ & & \gamma_{n-1} & -\beta_{n-1} - \gamma_n & \\ & & & \gamma_n & \end{bmatrix} = T_{n+1,n}$$

in (2.2) when we use (1.4) (which is (2.5)). We see $e^T L_{n+1,n} = 0$ necessarily. We now use this LDU factorization of $T_{n+1,n}$ to derive (1.5)–(1.7), and give their matrix formulations.

Note that $L_{n+1,n}$ differences columns, so we first define the matrix of direction vectors

$$P_n \equiv [p_0, p, \ldots, p_n] \equiv -X_{n+1} L_{n+1,n} D_n, \tag{2.10}$$

and with this we will show the equivalents of (1.5)–(1.7) are:

$$R_{n+1} L_{n+1,n} = A P_n D_n^{-1} \quad \text{or} \quad r_{n+1} = r_n - A p_n \omega_n, \quad \omega_n \equiv -\gamma_n^{-1}, \tag{2.11}$$

$$P_n = -X_{n+1} L_{n+1,n} D_n \quad \text{or} \quad x_{n+1} = x_n + p_n \omega_n, \tag{2.12}$$

$$R_n = P_n U_n \quad \text{or} \quad p_n = r_n + p_{n-1} \psi_{n-1}, \quad \psi_{n-1} \equiv \beta_{n-1}/\gamma_{n-1}. \tag{2.13}$$

First the columns of (2.10) give $p_n = (x_n - x_{n+1})\gamma_n$, which with $\omega_n \equiv -\gamma_n^{-1}$ gives (2.12), see (1.6). Then (2.8) multiplied by $L_{n+1,n}$ implies

$$R_{n+1} L_{n+1,n} = -A X_{n+1} L_{n+1,n} = A P_n D_n^{-1},$$

whose columns give $r_n - r_{n+1} = -A p_n \gamma_n^{-1}$ which is (2.11), see (1.5). Finally (2.6) implies

$$R_n = -X_{n+1} L_{n+1,n} D_n U_n = P_n U_n,$$

whose columns give $r_n = p_n - p_{n-1}\beta_{n-1}/\gamma_{n-1}$ which with $\psi_n \equiv \beta_n/\gamma_n$ is (2.13), see (1.7).

Of course the three-term recurrences can be derived from the two-term. In fact each of (2.3), (2.6), (2.8) can be obtained from just two of (2.11), (2.12), (2.13), see for example [22, 4, 1, 13]. In particular (2.11) and (2.12) give

$$(A X_{n+1} + R_{n+1}) L_{n+1,n} = 0, \tag{2.14}$$

so $R_{n+1} + A X_{n+1} = c e^T$, where multiplying by e_0 gives $c = b$, which is (2.8). Here the residual relation is implicitly defined in terms of necessarily well-behaved $L_{n+1,n}$, and we will show this leads to good finite precision recursive residuals for (1.5)–(1.7).

3 Local roundoff

Here we use r_n, x_n *etc.* to denote computed quantities. In finite precision arithmetic, recurrences (1.1)–(1.2) have to be replaced by

$$\begin{aligned} r_{n+1} &= (Ar_n - r_n\alpha_n - r_{n-1}\beta_{n-1} + g_n)/\gamma_n\,, \\ x_{n+1} &= -(r_n + x_n\alpha_n + x_{n-1}\beta_{n-1} - h_n)/\gamma_n\,, \end{aligned} \tag{3.1}$$

where g_n and h_n contain all the local errors produced at the step $n+1$.

The first step of the analysis consists in estimating these local errors.

We make the assumption that the floating-point arithmetic with roundoff unit ϵ satisfies

$$\mathrm{fl}(a \pm b) = a(1+\epsilon_1) \pm b(1+\epsilon_2), \quad |\epsilon_1|, |\epsilon_2| \le \epsilon, \tag{3.2}$$

$$\mathrm{fl}(a \ \mathrm{op}\ b) = (a \ \mathrm{op}\ b)(1+\epsilon_3), \quad |\epsilon_3| \le \epsilon, \quad \mathrm{op} \equiv *, /, \tag{3.3}$$

so that the roundoff in the matrix-vector multiplication is bounded according to

$$|\mathrm{fl}(Ap) - Ap| \le m\ \epsilon\ |A|\ |p| + \mathcal{O}(\epsilon^2) \tag{3.4}$$

when A has at most m nonzeros in any row, and the matrix-vector product is computed in the standard way. Assuming that the first and the third terms in (1.1)–(1.2) are summed first, we get via these rules

$$|g_n| \le \epsilon[(m+3)|A|\,|r_n| + 3|r_n\alpha_n| + 4|r_{n-1}\beta_{n-1}|\,] + \mathcal{O}(\epsilon^2), \tag{3.5}$$

$$|h_n| \le \epsilon[\,3|r_n| + 3|x_n\alpha_n| + 4|x_{n-1}\beta_{n-1}|\,] + \mathcal{O}(\epsilon^2). \tag{3.6}$$

We need not take norms or simplify these further, although it is obvious how to do so, since it will be sufficient to know g_n and h_n are bounded by ϵ times reasonable factors. Note in reasonable methods we expect these recursively computed residuals r_n to become smaller and smaller in norm, and the bound on g_n will decrease correspondingly, but that on h_n will not.

In the following estimates we assume that the computed coefficients α_n, β_{n-1}, and γ_n satisfy in analogy to (1.4)

$$\gamma_0 = -\alpha_0, \quad \gamma_n = -(\alpha_n + \beta_{n-1}) + \varepsilon_n, \quad \text{so} \quad e^T T_{n+1,n} = c_n^T \equiv (0, \varepsilon_1, \ldots, \varepsilon_n) \tag{3.7}$$

with error terms ε_n (note that this is another symbol than ϵ) that are bounded by

$$|\varepsilon_n| \le (|\alpha_n| + |\beta_{n-1}|)\,\nu\,\epsilon\,, \quad n \ge 1, \tag{3.8}$$

where ν is a suitable small constant. Note that $\nu = 1$ when γ_n is computed using (1.4).

We want to estimate the size of the difference between true and recursive residuals, hence, of

$$f_n \equiv b - Ax_n - r_n\,.$$

Since we take $r_0 = fl(b - Ax_0)$, $f_0 = b - Ax_0 - fl(b - Ax_0)$ is easily bounded giving

$$|f_0| \le [(m+1)|A|\,|x_0| + |b|\,]\,\epsilon + \mathcal{O}(\epsilon^2). \tag{3.9}$$

We will show that local errors in three-term recurrences are similar to those in three two-term recurrences. However, as we will see in the next section, the two three-term recurrences may suffer from a large amplification of the local errors.

4 The difference between the iterated and true residuals (vector recursion)

I suggest to consider including a short section of that type; the following text should be rewritten and significantly shortened, the present state is copied from the original version of the paper (I had no time to modify it). There might be readers for whom the vector recursions would be easier to follow and including this section will lead them into the matrix formulations and analysis in a natural way. Moreover, it will demonstrate the advantages of matrix analysis explicitly. In this way, some people who would not read the "pure matrix analysis" paper might be motivated to read the matrix analysis here

CCP: I'm still not convinced this should be in here. Can't we just refer to the Tech report? I'll leave it to you, but I won't check or alter this section.

Inserting the recursions (3.1) and the equality (3.7) we have

$$\begin{aligned} e_{n+1} &= b + (Ar_n + Ax_n\alpha_n + Ax_{n-1}\beta_{n-1})\,\frac{1}{\gamma_n} - Af_n \\ &\quad - (Ar_n - r_n\alpha_n - r_{n-1}\beta_{n-1})\,\frac{1}{\gamma_n} - g_n \\ &= -(b - Ax_n - r_n)\,\frac{\alpha_n}{\gamma_n} - (b - Ax_{n-1} - r_{n-1})\,\frac{\beta_{n-1}}{\gamma_n} - b\frac{\varepsilon_n}{\gamma_n} - Af_n - g_n \\ &= -\left(e_n\frac{\alpha_n}{\gamma_n} + e_{n-1}\frac{\beta_{n-1}}{\gamma_n} + b\frac{\varepsilon_n}{\gamma_n} + Af_n + g_n\right). \end{aligned} \tag{4.1}$$

Let us gather the last three terms, the local error (or local contribution) in the recursion for the investigated global difference e_n in

$$l_n := b\frac{\varepsilon_n}{\gamma_n} + Af_n + g_n\,.$$

Then, by inserting the estimates (3.5)–(3.8) we get

$$\begin{aligned} ||l_n|| \;\le\; &\Big[||b||\,(|\alpha_n| + |\beta_{n-1}|) + (\mu+4)\,||A||\rho_n + 3|\beta_{n-1}|\,(||A||\,||x_{n-1}|| + \rho_{n-1}) \\ &+ 2|\alpha_n|\,(||A||\,||x_n|| + \rho_n)\Big]\,\frac{\epsilon}{|\gamma_n|} + ||A||\,||x_{n+1}||\epsilon + \rho_{n+1}\epsilon + \mathcal{O}(\epsilon^2). \end{aligned}$$

This justifies to define

$$\varphi_n := ||b|| + ||A||\,||x_n|| + \rho_n$$

so that

$$||l_n|| \le [(\mu+4)\,||A||\rho_n + 2\varphi_n|\alpha_n| + 3\varphi_{n-1}|\beta_{n-1}|]\,\frac{\epsilon}{|\gamma_n|} + \varphi_{n+1}\epsilon + \mathcal{O}(\epsilon^2). \tag{4.2}$$

In [10] the local error term at the step k is essentially bounded by $\mathcal{O}(\epsilon)||A||\,\max_{1\le j\le k}||x_j||$. In our case, the similar term in the bound for $||l_k||$ that can be derived from (4.2) is

multiplied by the factor $(2|\alpha_k| + 3|\beta_{k-1}|)/|\gamma_k|$, which can be substantially larger than 1. We see that local errors in three-term recurrences are potentially larger than those in the coupled two-term recurrences. In addition to that, as we will see in the next section, the three-term recurrences may suffer from an additional large amplification of the local errors.

The recursions (3.1) and (3.7) are valid for $n \geq 1$, while for $n = 0$ they simplify since $\beta_{-1} := 0$. Consequently, some of the estimates simplify also. In particular, $\gamma_0 = -\alpha_0$, and thus $\varepsilon_0 = 0$. Thus in view of (4.1) we find the second order difference equation

$$e_1 = e_0 - l_0\,, \quad e_{n+1} = -\left(e_n \frac{\alpha_n}{\gamma_n} + e_{n-1}\frac{\beta_{n-1}}{\gamma_n} + l_n\right) \quad (n \geq 1), \tag{4.3}$$

with $e_1 = e_0 - l_0$. These recurrences describe the propagation of the local rounding errors l_k, $k = 0, \ldots, n$. We see that the global gap between the recursively computed residual and the true residual after n steps, e_n, is determined by a inhomogeneous second order difference equation. This is in sharp contrast with the error behavior of the coupled two-term recurrences, where the global error after n steps is just a simple sum of local errors; see [10].

Our analysis of the inhomogeneous second order difference equation (4.3) satisfied by the global rounding errors is based on the observation that we can write n steps of (4.3) as the superposition of $n + 1$ homogeneous recurrence relations. In a different context this idea has been used by Grcar [7]. Considering this superposition, we will use the special relation (3.7) between the recurrence coefficients and significantly simplify the formula for the global error e_n.

For the moment, assume that the term ε_n in (3.7) vanishes, *i.e.*, that

$$-\frac{\alpha_n}{\gamma_n} - \frac{\beta_{n-1}}{\gamma_n} = 1 \tag{4.4}$$

holds even in finite precision arithmetic. Denote by $z_{n+1} = \mathcal{D}(z_{n-m+1}, z_{n-m}; m)$ the result of m steps of the recurrence

$$z_{k+1} = -z_k\frac{\alpha_k}{\gamma_k} - z_{k-1}\frac{\beta_{k-1}}{\gamma_k}\,, \quad k = n-m+1, \ldots, n, \tag{4.5}$$

started at the step $n-m+1$. Note that due to (4.4), $z_{n-m+k+1} = \mathcal{D}(z_{n-m+1}, z_{n-m}; k) = z_{n-m}$ for all k whenever $z_{n-m+1} = z_{n-m}$. Our discussion will heavily rely on this fact.

First, we derive how the error e_{n+1} is affected by e_0. Clearly, the part of this error that depends on e_0 is given by

$$\mathcal{D}(e_0, e_0; n) = e_0\,,$$

i.e., e_0 is not amplified in the process. Next we have to analyze the dependence of e_{n+1} on the elementary errors born in the first step of the algorithm. Clearly, due to (4.4),

$$-l_0\frac{\alpha_1}{\gamma_1} = l_0 + l_0\frac{\beta_0}{\gamma_1}.$$

Therefore, the contribution of l_0 to the error e_{n+1} consists of

$$\mathcal{D}(-l_0, -l_0; n-1) = -\, l_0\,,$$

and the part depending on the modified local error of the first step,

$$lt_1 := \; l_0\frac{\beta_0}{\gamma_1} + l_1\,,$$

which has yet to be analyzed. Repeating the same idea for the steps 2 through n, we can conclude that e_{n+1} can be superposed as

$$\begin{aligned} e_{n+1} = e_0 \; & - l_0 \\ & - l_0\frac{\beta_0}{\gamma_1} - l_1 \\ & - l_0\frac{\beta_0\beta_1}{\gamma_1\gamma_2} - l_1\frac{\beta_1}{\gamma_2} - l_2 \\ & \;\;\vdots \\ & - l_0\frac{\beta_0\beta_1\cdots\beta_{n-1}}{\gamma_1\gamma_2\cdots\gamma_n} - \ldots - l_{n-1}\frac{\beta_{n-1}}{\gamma_n} - l_n\,. \end{aligned} \tag{4.6}$$

Now we describe how the picture changes when the coefficients α_n, β_{n-1}, and γ_n are computed imprecisely, *i.e.*, when (4.4) is replaced by (3.7). We can follow the analysis described above with the only difference that we should add the effect of the quantity $e_0\varepsilon_n$ propagating through $n-1$ steps of the recurrence (4.5) with $z_1 = \mathbf{o}$, the effect of $lt_1\varepsilon_n$ propagating through $n-2$ steps of (4.5) with $z_2 = \mathbf{o}$, and so on. As long as the constant ν is small and ε_n is close to the machine precision ϵ, these modifications will only cause effects proportional to $\mathcal{O}(\epsilon^2)$. In (4.6) we should therefore add terms $\mathcal{O}(\epsilon^2)$ to individual terms of the sum. However, once we will consider the size of these terms, the new $\mathcal{O}(\epsilon^2)$ contribution can be thought of being incorporated in the $\mathcal{O}(\epsilon^2)$ terms already present in our bounds (3.9) and (4.2) for e_0, l_0, $\ldots$, l_n. Therefore, in the further analysis, we can use (4.6) with no change and no limitation.

5 The difference between the iterated and the true residuals (matrix analysis)

I did not like denoting the submatrices (with omitted first rows) using hats. I found the text with many different matrices difficult to follow. There is no ideal notation (I do not like my notation much better, but it seems to me a bit easier to read). Perhaps we should think about some better notation.

CCP: Unfortunately because we are starting with 0 rather than 1, I suggest we change all your 2s to 1s, as follows.

It was shown in [19, 20, 21] how effective it is to treat the errors of Krylov methods using matrix forms, so we follow this approach here. In accordance with indexing

commencing at 0, see (2.2), we will use e_i to denote the $(i+1)$-st column of the unit matrix (in fact the i-th when we count from 0), and $e \equiv (1,\ldots,1)^T$.

For ease of reference we repeat (2.3), (2.6) and (2.5), the results with exact arithmetic:

$$AR_n = R_{n+1}T_{n+1,n}, \qquad R_n + X_{n+1}T_{n+1,n} = 0, \qquad e^T T_{n+1,n} = 0^T.$$

When we include the local errors, see (3.1) and (3.7), we get for the computed values, with $G_n \equiv [g_0,\ldots,g_n]$, $H_n \equiv [h_0,\ldots,h_n]$ and $L_{n+1,n}$, D_n, U_n exactly as in (2.9)

$$\begin{aligned} AR_n + G_n &= R_{n+1}T_{n+1,n}, \quad R_n + X_{n+1}T_{n+1,n} = H_n, \\ T_{n+1,n} &= L_{n+1,n}D_nU_n + \begin{bmatrix} \mathrm{diag}(0,\varepsilon_1,\ldots,\varepsilon_n) \\ 0 \end{bmatrix} \end{aligned} \tag{5.1}$$

We wish to bound $f_n \equiv b - Ax_n - r_n$. In general write

$$[f_0, F_n] \equiv [f_0, f_1, \ldots, f_{n+1}] = be^T - AX_{n+1} - R_{n+1}. \tag{5.2}$$

But from (5.1) $AR_n = AH_n - AX_{n+1}T_{n+1,n} = R_{n+1}T_{n+1,n} - G_n$, so

$$(AX_{n+1} + R_{n+1})T_{n+1,n} = G_n + AH_n, \tag{5.3}$$

giving with the definition in (3.7)

$$[f_0, F_n]T_{n+1,n} = be^T T_{n+1,n} - (AX_{n+1} + R_{n+1})T_{n+1,n} = bc_n^T - G_n - AH_n.$$

Writing in accordance with the partitioning in the first matrix in (5.2)

$$\begin{pmatrix} e_0^T \\ L_{1:n+1,n} \end{pmatrix} \equiv L_{n+1,n}, \qquad \begin{pmatrix} t_{0\cdot} \\ T_{1:n+1,n} \end{pmatrix} \equiv T_{n+1,n}, \tag{5.4}$$

(remembering we are counting from 0), we have

$$F_nT_{1:n+1,n} = S_n \equiv bc_n^T - G_n - AH_n - f_0t_{0\cdot},$$

where S_n can be simply bounded, and $T_{1:n+1,n}$ is a known nonsingular upper tridiagonal matrix, giving

$$F_n = S_nT_{1:n+1,n}^{-1}. \tag{5.5}$$

Using (5.1) we have

$$F_n\hat{E} = S_n(D_nU_n)^{-1}L_{1:n+1,n}^{-1}, \tag{5.6}$$

CCP: I was unable to derive this. Can you elaborate?
where $\hat{E}$ is the unit matrix with the first updiagonal perturbed to $(\varepsilon_1,\ldots,\varepsilon_n)$. A very similar result also follows from (5.1), (5.2), (5.3) and $e^TL_{n+1,n} = 0$:

$$\begin{aligned} [f_0, F_n]L_{n+1,n}D_nU_n &= -(AX_{n+1} + R_{n+1})\left\{T_{n+1,n} - \begin{bmatrix} \mathrm{diag}(0,\varepsilon_1,\ldots,\varepsilon_n) \\ 0 \end{bmatrix}\right\} \\ &= \hat{S}_n \equiv (AX_n + R_n)\mathrm{diag}(0,\varepsilon_1,\ldots,\varepsilon_n) - G_n - AH_n, \end{aligned} \tag{5.7}$$

where $\hat{S}_n$ is easily bounded via (3.5)–(3.8). It follows that

$$F_n = [-f_0 e_0^T + \hat{S}_n (D_n U_n)^{-1}] L_{1:n+1,n}^{-1},$$

$$L_{1:n+1,n}^{-1} = -\begin{pmatrix} 1 & 1 & \cdot & 1 \\ & 1 & \cdot & 1 \\ & & \cdot & \cdot \\ & & & 1 \end{pmatrix}, \quad (D_n U_n)^{-1} = -\begin{pmatrix} \frac{1}{\gamma_0} & \frac{\beta_0}{\gamma_0}\frac{1}{\gamma_1} & \cdot & \frac{\beta_0}{\gamma_0}\frac{\beta_1}{\gamma_1}\cdots\frac{\beta_{n-1}}{\gamma_{n-1}}\frac{1}{\gamma_n} \\ & \frac{1}{\gamma_1} & \cdot & \frac{\beta_1}{\gamma_1}\cdots\frac{\beta_{n-1}}{\gamma_{n-1}}\frac{1}{\gamma_n} \\ & & \cdot & \cdot \\ & & & \frac{1}{\gamma_n} \end{pmatrix}. \tag{5.8}$$

Finally using these gives

$$f_{n+1} = f_0 - \hat{S}_n (D_n U_n)^{-1} e, \tag{5.9}$$

which shows how the local rounding errors (f_0 in (3.9), and the g_j, h_j and ε_j which make up $\hat{S}_n$) can be magnified and accumulated to give the global error f_{n+1} in the recurred residual r_{n+1}, see (5.2). Since $(D_n U_n)^{-1}$ is upper triangular, it is the leading principal submatrix of $(D_m U_m)^{-1}$, $m > n$. Thus any large element in $(D_n U_n)^{-1}$ will always appear in the expression for f_{m+1}. It follows that if a recurred residual loses significant accuracy, it is almost certain that all later residuals will have similar or worse absolute errors.

Thus the accuracy of the recurred residual in methods implemented via (1.1) and (1.2) is heavily dependent on the sizes of the elements in $(D_n U_n)^{-1}$. These will necessarily be reasonable in only very few methods — the Chebyshev iteration being one — but when they are not, these implementations should be avoided. This is especially so since (1.5)–(1.7) does not have this deficiency, as we will show in Section 6. To compare the two behaviors we give a quick bound here on $\hat{s}_n$ in (5.9), see (5.7):

$$\begin{aligned} \hat{s}_n &\equiv (Ax_n + r_n)\epsilon_n - g_n - Ah_n, \\ |\hat{s}_n| &\leq \epsilon\left[(|\alpha_n| + |\beta_{n-1}|)\,(|A||x_n| + |r_n|) + (m+3)|A||r_n|\right. \\ &\quad \left. +3|r_n\alpha_n| + 4|r_{n-1}\beta_{n-1}| + |A|\,(3|r_n| + 3|x_n\alpha_n| + 4|x_{n-1}\beta_{n-1}|)\right] + \mathcal{O}(\epsilon^2), \\ &\leq \epsilon\left[(m+6)|A||r_n| + 4\,(|A||x_n| + |r_n|)\,|\alpha_n|\right. \\ &\quad \left. +4\,(|A||x_{n-1} + |r_{n-1}|)\,|\beta_{n-1}|\right] + \mathcal{O}(\epsilon^2). \end{aligned} \tag{5.10}$$

6 Comparison with three two-term recurrences

In our notation, Greenbaum's error term [10] for the three two-term recurrences (1.5)–(1.7) is

$$f_{n+1}^G = f_0 - \sum_{j=0}^{n} s_j^G, \quad \text{where } s_j^G \equiv Ah_j^G + g_j^G, \tag{6.1}$$

with g_n^G and h_n^G denoting the local roundoff errors in the evaluation of the first two recurrences of (1.5)–(1.7), analogously to g_n and h_n in (3.1). Because no confusion is possible, we drop the superscript G in the rest of this section. As it is straight forward and brief, we repeat the analysis here using the same approach as above. Using the

theory of Section 3 we see that with finite precision arithmetic the recurrences (1.5)–(1.7) have to be replaced by (see [9])

$$\begin{aligned}
r_{n+1} &= r_n - Ap_n\omega_n - g_n,\\
x_{n+1} &= x_n + p_n\omega_n - h_n,\\
p_{n+1} &= r_{n+1} + p_n\psi_n + k_{n+1},\\
|g_n| &\le \epsilon\,[(m+2)|A||p_n\omega_n| + |r_n|] + \mathcal{O}(\epsilon^2),\\
|h_n| &\le \epsilon\,(|x_n| + 2|p_n\omega_n|) + \mathcal{O}(\epsilon^2),\\
|k_{n+1}| &\le \epsilon\,(|r_{n+1}| + 2|p_n\psi_n) + \mathcal{O}(\epsilon^2).
\end{aligned}$$

Let

$$L_{n+1,n} \equiv \begin{pmatrix} e_0^T \\ L_{1:n+1,n} \end{pmatrix}, \quad L_{1:n+1,n} \equiv \begin{pmatrix} -1 & 1 & & \\ & \cdot & \cdot & \\ & & -1 & 1 \\ & & & -1 \end{pmatrix}, \quad U_n \equiv \begin{pmatrix} 1 & -\psi_0 & & \\ & \cdot & \cdot & \\ & & 1 & -\psi_{n-1} \\ & & & 1 \end{pmatrix}$$

and $D_n^{-1} \equiv \mathrm{diag}(\omega_0, \ldots, \omega_n)$, see (2.9), and the definitions in (2.13) and (2.11). Remember (2.11)–(2.13) for the case with no errors:

$$R_{n+1}L_{n+1,n} = AP_nD_n^{-1}, \qquad X_{n+1}L_{n+1,n} = -P_nD_n^{-1}, \qquad P_{n+1}U_{n+1} = R_{n+1}.$$

With the above local rounding errors these become

$$R_{n+1}L_{n+1,n} = AP_nD_n^{-1} + G_n,\ X_{n+1}L_{n+1,n} = -P_nD_n^{-1} + H_n,\ P_{n+1}U_{n+1} = R_{n+1} + K_{n+1}. \tag{6.2}$$

But as in (5.2) write $[f_0, F_n] = be^T - AX_{n+1} - R_{n+1}$ for this algorithm, giving

$$[f_0, F_n]L_{n+1,n} = AP_nD_n^{-1} - AH_n - AP_nD_n^{-1} - G_n = -G_n - AH_n,$$

and with the notation in (5.4) and using (5.8)

$$\begin{aligned}
F_n &= -(G_n + AH_n + f_0[1, 0, \ldots, 0])L_{1:n+1,n}^{-1} \equiv f_0e^T + S_n\hat{L}_{1:n+1,n}^{-1}, \quad \text{say, so}\\
f_{n+1} &= f_0 - S_ne,
\end{aligned}$$

just as in (6.1), where now we can bound $s_n = -g_n - Ah_n$ by using $p_n\omega = x_{n+1} - x_n + h_n$

$$\begin{aligned}
|s_n| &\le \epsilon\,[(m+2)\,|A|\,(|x_{n+1}| + |x_n|) + |r_n| + |A||x_n| + 2|A|\,(|x_{n+1}| + |x_n|)] + \mathcal{O}(\epsilon^2)\\
&\le \epsilon\,[(m+5)|A|\,(|x_{n+1}| + |x_n|) + |r_n|] + \mathcal{O}(\epsilon^2).
\end{aligned} \tag{6.3}$$

In any sensible algorithm the bounds (5.10) and (6.3) will be comparable, and the difference in the errors in the recursive residuals will be determined largely by the growth factors. Clearly the local error can get blown up more in two three-term recurrences (the factor $(D_nU_n)^{-1}e$ in (5.9)) than in the three two-term recurrences (1.5)–(1.7) (the factor e above), indicating the general superiority of the latter over the former for computing residuals, see Section 10.

7 Reliability of the three-term residual recurrence

Because in finite precision the (1.1), (1.2) combination can give a significant difference between the actual and recursive residuals, in contrast to (1.5)–(1.7), one might superficialy concluded that (1.1) by itself is unreliable. This is wrong, and to support this argument we will prove the recurrence (1.1) gives a recursive residual obeying a similar relation to that from (1.5)–(1.7).

At first this seems contradictory, but it is easily accepted once it is understood that it is the *actual*, not the recursive, residual that can cause the main trouble in (1.1)–(1.2). This will be seen in the examples, where this combination leads to an actual residual which does not converge nearly as well as that from three two-term recurrences. Thus two three-term recurrences can lead to x_n iterates which cause the actual residual $r_n = b - Ax_n$ to be significantly worse than necessary. This can happen even though in CG these x_n iterates have typically not much greater error than those from three two-term recurrences. From (5.1), see also (3.1), we saw the three-term recurrence (1.1) for the residual gives with finite precision computation

$$AR_n + G_n = R_{n+1}T_{n+1,n}, \tag{7.1}$$

with the columns of G_n bounded as in (3.5). For the three two-term recurrences with rounding errors, (1.5) and (1.7) may be written, see (6.2), but using superscript G for distinction,

$$R_{n+1}L_{n+1,n}D_n = AP_n + G_n^G D_n, \qquad P_nU_n = R_n + K_n^G.$$

Combining these we see

$$AR_n = -AK_n^G + AP_nU_n = R_{n+1}L_{n+1,n}D_nU_n - AK_n^G - G_n^G D_nU_n. \tag{7.2}$$

Comparing this with (7.1), and noting (5.1) and (2.9), shows the recursive residuals in the two implementations satisfy very similar global equations, suggesting the recursive residual computed by (1.1) is not intrinsically worse than that computed by (1.5)–(1.7). Thus implementations based on (1.1) need not have significantly different numerical behavior to mathematically identical implementations based on three two-term recurrences.

This does not say the recursive residuals will necessarily be good using an algorithm based on (1.1), as the properties will also depend on how the coefficients α_n, β_n and γ_n are computed. Thus to show the recursive residuals for an algorithm based on (1.1) are as good as those of a mathematically equivalent algorithm based on (1.5)–(1.7) would require a more complete analysis.

8 Rutishauser's variant of the recurrences

It is useful to note that the above numerical difficulties encountered using (1.1) and (1.2) can also be avoided by using an elegant technique suggested by H. Rutishauser [24]. His ideas apply not only to CG, which was considered in [24], but to any method based on (1.1) and (1.2), so we derive his variant directly from these.

The idea is to replace the recurrence for the residual by a recurrence for the residual *increment*, and to replace the recurrence for the iterate by a recurrence for the *increment* in the iterate. Thus subtracting r_n from both sides of (1.1), and using (1.4), gives for $n = 1, 2, \ldots$

$$\begin{aligned} \Delta r_n &:= (Ar_n + \Delta r_{n-1}\beta_{n-1})/\gamma_n, \\ r_{n+1} &:= r_n + \Delta r_n. \end{aligned} \tag{8.1}$$

A similar approach to (1.2) gives for $n = 1, 2, \ldots$

$$\begin{aligned} \Delta x_n &:= (-r_n + \Delta x_{n-1}\beta_{n-1})/\gamma_n, \\ x_{n+1} &:= x_n + \Delta x_n. \end{aligned} \tag{8.2}$$

Note for $n = 0$ the resulting equations are $r_1 - r_0 = Ar_0/\gamma_0$, $x_1 - x_0 = -r_0/\gamma_0$, so setting

$$\Delta r_{-1} := \Delta x_{-1} := 0, \qquad r_0 := b - Ax_0, \tag{8.3}$$

allows the above recurrences to hold for $n = 0, 1, 2, \ldots$. Thus Rutishauser's variant uses four two-term recurrences, but does not introduce p_n. The relationship with p_n follows from (1.6), (2.11) and (8.2):

$$\begin{aligned} \Delta x_n &= p_n\omega_n = (-r_n + \Delta x_{n-1}\beta_{n-1})/\gamma_n, \\ p_n &= r_n - \Delta x_{n-1}\beta_{n-1}, \qquad \beta_{-1} \equiv 0. \end{aligned} \tag{8.4}$$

As before, it will simplify the analysis if we express the four recurrences (8.1)–(8.2) in matrix form. With $L_{n+1,n}$ and D_nU_n as in (2.9), and

$$\Delta R_n \equiv [\Delta r_0, \ldots, \Delta r_n], \qquad \Delta X_n \equiv [\Delta x_0, \ldots, \Delta x_n],$$

we see (with (8.3) that (8.1) corresponds to

$$\Delta R_n = -R_{n+1}L_{n+1,n}, \qquad AR_n = -\Delta R_n D_n U_n, \tag{8.5}$$

while (8.2) corresponds to

$$\Delta X_n = -X_{n+1}L_{n+1,n}, \qquad R_n = \Delta X_n D_n U_n. \tag{8.6}$$

We can draw similar conclusions to those in Section 2. In particular

$$\begin{aligned} \Delta R_n &= -AR_n(D_nU_n)^{-1} = -A\Delta X_n, \\ (R_{n+1} + AX_{n+1})L_{n+1,n} &= -\Delta R_n - A\Delta X_n = 0, \end{aligned} \tag{8.7}$$

so $R_{n+1} + AX_{n+1} = ce^T$, and if $r_0 = b - Ax_0$ then (2.8) holds.

9 A particular case: Conjugate Gradients

So far the results have held for any α_n, $\beta_n - 1$, γ_n satisfying $\gamma_n \neq 0$ and (1.4). The choice of these determines the particular method. For our numerical computations in Section 10 and our discussion here we restrict ourselves to symmetric positive definite matrices A and to the method of conjugate gradients (CG). We will briefly indicate the relevant theory, then develop our rounding error analysis of Sections 3–6 to handle this case.

9.1 Conjugate Gradients: Theory

Conjugate gradients [16] has many interesting properties, but it is probably easiest to develop it here from the fact that it produces orthogonal $r_0, r_1, \ldots$ in (2.3), see the Lanczos process [17]. Then from (2.3)

$$R_n^T A R_n = [\mathrm{diag}(r_i^T r_i), 0] T_{n+1,n}$$

is symmetric, so with (2.2), $r_n^T r_n \gamma_{n-1} = r_{n-1}^T r_{n-1} \beta_{n-1}$ and $\alpha_n = r_n^T A r_n / r_n^T r_n$. By using (1.4) we see that the coefficients can be computed via: $\beta_{-1} = 0$, and for $n = 0, 1, \ldots$

$$\alpha_n := \frac{r_n^T A r_n}{r_n^T r_n}, \tag{9.1}$$

$$\gamma_n := -\alpha_n - \beta_{n-1}, \tag{9.2}$$

$$\beta_n := \gamma_n \frac{r_{n+1}^T r_{n+1}}{r_n^T r_n}, \tag{9.3}$$

which is what Rutishauser [24] used.

Let L_n denote the matrix obtained by omitting the last row of $L_{n+1,n}$. Then for the three two-term recurrence methods we have from (2.11) and (2.13)

$$P_n^T A P_n = U_n^{-T} \mathrm{diag}(r_i^T r_i) L_n D_n,$$

which is both symmetric and lower triangular, and so is diagonal. From $U_n^T P_n^T A P_n = \mathrm{diag}(r_i^T r_i) L_n D_n$ we then see $p_n^T A p_n = -r_n^T r_n \gamma_n$, so with the definitions in (2.11) and (2.13), and with (9.3)

$$\omega_n = \frac{r_n^T r_n}{p_n^T A p_n}, \qquad \psi_n = \frac{r_{n+1}^T r_{n+1}}{r_n^T r_n}, \tag{9.4}$$

which is what Hestenes and Stiefel [16] used.

9.2 Conjugate Gradients: Practice

We saw in (5.9) how the local rounding errors could be blown up in the two three-term recurrence leading to a seriously inaccurate residual. We will show here that even so, in CG the *error* has no worse a bound than the three two-term recurrence implementation. This might be one reason this weakness of the two three-term variant of CG was not so obvious, so it is useful to show it. However since we are now unlikely to use this variant, we will only develop the ideas briefly.

early CG work.

Next we want to discuss the size of the multiplicative factors

$$\prod_{j=i}^{k} \frac{\beta_{j-1}}{\gamma_j},$$

where from now on we assume, without repeating it, that $1 \le i \le k$. For this discussion we restrict ourselves to symmetric positive definite matrices A and to the method of conjugate gradients (CG), where, in exact arithmetic,

$$\omega_n = \frac{\langle r_n, r_n \rangle}{\langle p_n, Ap_n \rangle}, \quad \psi_n = \frac{\langle r_{n+1}, r_{n+1} \rangle}{\langle r_n, r_n \rangle}. \tag{9.5}$$

Both ω_n and ψ_n are positive. Without a specific knowledge about A and r_0 we cannot say anything more about their values. More precisely, given any two sequences of positive numbers, $\omega_0, \ldots, \omega_{N-1}$ and $\psi_0, \ldots, \psi_{N-1}$, there are a symmetric positive definite matrix A and a vector r_0 such that the CG algorithm applied to A with the initial residual r_0 generates the given coefficients; see Theorem 18:3 in [16]. The β_n and γ_n then satisfy, see (2.11), (2.13) and (1.4)

$$\gamma_n = -\frac{1}{\omega_n} < 0, \quad \frac{\alpha_n}{\gamma_n} = -1 - \frac{\psi_{n-1}\omega_n}{\omega_{n-1}} \le -1, \quad \frac{\beta_{n-1}}{\gamma_n} = \frac{\psi_{n-1}\omega_n}{\omega_{n-1}} \ge 0, \tag{9.6}$$

where $\psi_{-1} := 0$, $\omega_{-1} := 1$, and where the equality is attained in the last two cases only if $x_n = x$, *i.e.*, if we have reached the solution. We conclude that the multiplicative factors have the form

$$\prod_{j=i}^{k} \frac{\beta_{j-1}}{\gamma_j} = \frac{\omega_k}{\omega_{i-1}} \prod_{j=i}^{k} \psi_{j-1}\,, \tag{9.7}$$

and therefore, they may exhibit, in general, an arbitrary behavior.

For a given matrix A and an initial residual r_0 it is possible to relate the size of the multipliers to the condition number of A and the convergence of the CG process measured by the norm of the residuals. Still assuming exact arithmetic we receive, when rewriting the multipliers in the form

$$\prod_{j=i}^{k} \frac{\beta_{j-1}}{\gamma_j} = \frac{\omega_k}{\omega_{i-1}} \, \frac{||r_k||^2}{||r_{i-1}||^2}$$

and using Theorem 5:5 from [16], the following bound:

$$\frac{1}{\kappa(A)} \, \frac{||r_k||^2}{||r_{i-1}||^2} \le \prod_{j=i}^{k} \frac{\beta_{j-1}}{\gamma_j} \le \kappa(A) \, \frac{||r_k||^2}{||r_{i-1}||^2}, \tag{9.8}$$

where $\kappa(A)$ is the spectral condition number of the matrix A. Note that

$$\frac{||r_k||^2}{||r_{i-1}||^2} \le \frac{||A^{1/2}\, A^{1/2}\,(x - x_k)||^2}{||A^{1/2}\, A^{1/2}\,(x - x_{i-1})||^2} \le \frac{||A||}{\sigma_{\min}(A)} \, \frac{||x - x_k||_A^2}{||x - x_{i-1}||_A^2} \le \kappa(A)$$

due to the monotonicity in the reduction of the A-norm of the error. Consequently,

$$\prod_{j=i}^{k} \frac{\beta_{j-1}}{\gamma_j} \le \kappa^2(A).$$

Moreover, the results of [8] and [12] imply then that in finite precision arithmetic the following slightly relaxed bounds hold:

$$(1-\vartheta)\,\frac{1}{\kappa(A)}\,\frac{||r_k||^2}{||r_{i-1}||^2} \le \prod_{j=i}^{k}\frac{\beta_{j-1}}{\gamma_j} \le (1+\vartheta)\,\kappa(A)\,\frac{||r_k||^2}{||r_{i-1}||^2}\,, \tag{9.9}$$

$$\prod_{j=i}^{k}\frac{\beta_{j-1}}{\gamma_j} \le (1+\vartheta)\,\kappa^2(A)\,, \tag{9.10}$$

where $0 \le \vartheta \ll 1$ (here, we make the usual assumption about the numerical nonsingularity of the matrix A; for details see the references mentioned above).

In the nonsymmetric case no bound similar to (9.9) can be expected to hold.

One can make a natural comment: if the multipliers become very large, then the two three-term recurrence solvers are likely to exhibit a dramatically worse residual behavior than the three two-term ones. For the CG method, Hestenes and Stiefel [16] show in their Theorem 18:3 how to construct examples having any given set of multipliers. However, if the matrix A is reasonably well conditioned and if the CG method converges almost monotonically, then the bounds (9.9) for the multipliers show that no dramatic amplification of the local errors can be expected. On the other hand, if, for some k and i, the factor $||r_k||^2/||r_{i-1}||^2$ is large, then we may indeed expect a large difference in the residuals of the two three-term versus the three two-term recurrences. It is important to note that any significant local oscillations of the residual norms are potentially dangerous for the CG implementation based on three-term recurrences (1.1)–(1.2), even those for which $||r_k||$ is much smaller than $||r_0||$. As illustrated by the numerical experiments in Section 10, local oscillations may cause an incurable damage to the final residual and the process may never recover despite the smooth convergence in the subsequent steps.

Here the analysis by Chris must be incorporated, possibly leading to the necessary changes of the text above

10 Numerical experiments

The construction of our numerical experiments follows ideas from [16]. We consider $N = 48$ and aim at the following values of the coefficients (9.5) for the three two-term recurrences of the CG method:

$$\begin{aligned}
\omega_0 &= \omega_1 = \ldots = \omega_{47} = 1,\\
\psi_0 &= 10, \quad \psi_1 = \psi_3 = \ldots = \psi_{43} = 0.01, \quad \psi_2 = \ldots = \psi_{44} = 100,\\
\psi_{45} &= 10^{-2}, \quad \psi_{46} = 10^{-3}, \quad \psi_{47} = 10^{-4}.
\end{aligned}$$

Using the formulas for the elements of $N \times N$ matrix T, see (9.6)

$$T(1,1) = \frac{1}{\omega(0)}\,,$$

$$
\begin{aligned}
T(i,i) &= \frac{1}{\omega(i-1)} + \frac{\psi(i-2)}{\omega(i-2)}\,, \\
T(i,i-1) &= T(i-1,i) = \frac{(\psi(i-2))^{1/2}}{\omega(i-2)}\,, \quad i = 2,\ldots,N,
\end{aligned}
$$

we construct a symmetric positive definite N by N tridiagonal matrix T. For any unitary N by N matrix V, the CG method (1.5)–(1.7), (9.5) applied to the system $Ax = b$ with $A = VTV^\star$ and $r_0 = b - Ax_0 = Ve_0$ then generates in steps 1 to N the prescribed coefficients ω_j, ψ_j, $j = 0,\ldots,N-1$. Then for the generated residual norms

$$
\begin{aligned}
\|r_j\|_2 &= 10^{1/2} \quad \text{for} \quad j = 1,3,\ldots,43, \\
\|r_j\|_2 &= 10^{-1/2} \quad \text{for} \quad j = 2,4,\ldots,44,
\end{aligned}
$$

and the residual norm is sharply decreasing in the steps 45 through 48. For an initial residual different from Ve_0 the behavior of the residual norms will be different, but we still may expect some oscillations.

We have used the construction described above, choosing V as the unitary matrix resulting from the QR decomposition of a randomly generated N by N matrix; in MATLAB notation $[V,R] = qr(\text{randn}(N,N))$. Furthermore, we have chosen $x = (1,\ldots,1)^\top$, $b = Ax$, $x_0 = \mathbf{o}$, $r_0 = b$. Experiments were performed on an SGI Indigo Workstation using MATLAB 5.0, $\epsilon = 1.11 \times 10^{-16}$. Three implementations of the CG method have been compared: solid lines represent results of the Hestenes-Stiefel variant (HS) given by (1.5)–(1.7) and (9.5), dashed lines those of the Rutishauser variant (R) described in [24], see also [23], and dotted lines those of the three term CG implementation presented in [15], p. 143, and denoted here as (HY). Note this last is described in [15] as coming from the same book as [24], but we were unable to find it there.

Norms of recursively computed residuals are compared in Figure 1. We can see the oscillations followed by the fast convergence for n around 70.

True residual norms, computed as $\|b - Ax_n\|_2$, are compared in Figure 2. We see that the final accuracy of the (R) and (HS) variants are comparable. However, residual norms of the (HY) variant stagnate at a significantly worse level than those of the (HS) variant, as predicted by our theoretical analysis. But we should also mention that surprisingly, despite the differences in the true residuals, all three variants give comparable error norms.

In our experiments, the described behavior was typical. A detailed analysis of the Rutishauser implementation, which is not of the form (1.1)–(1.2) or (1.5)–(1.7), and of the behavior of the error in all variants requires further work.

11 Conclusions

We have proven that implementations of Krylov space methods based on two three-term recurrences (1.1)–(1.2) potentially produce less accurate residuals than the corresponding implementations based on three two-term recurrences of the form (1.5)–(1.7) and that this difference may be significant.

For the conjugate-gradient method for example, the difference between the recursive and the true residuals in implementations using two three-term recurrences is affected not only by the maximum size of the intermediate iterates x_n, but also by oscillations of the squared norms of the residuals, that is the quantities $||r_k||^2/||r_{i-1}||^2$, $1 \leq i \leq k$.

Note that many useful algorithms are effectively based on the first recurrence in (1.1)–(1.2), but not the second, see for example SYMMLQ in [22]. We have shown that this first recurrence by itself does not necessarily cause the kind of difficulty with the residual described here, and indeed SYMMLQ does not exhibit this difficulty. The approach that has been used here for analyzing the rounding error behavior of recurrence methods can presumably also be applied to methods such as these. Other important algorithms like the three-term and the coupled two-term QMR methods [5, 6] are not implemented in the form (1.1)–(1.2) and (1.5)–(1.7). The propagation of elementary roundoff in these algorithms can presumably also be analyzed in a way analogous to the approach described in this paper.

Acknowledgment. The authors would like to thank Anne Greenbaum and Gerard Meurant for their helpful comments.

Bibliography

[1] S. F. Ashby and M. H. Gutknecht, *A matrix analysis of conjugate gradient algorithms*, in Advances in Numerical Methods for Large Sparse Sets of Linear Systems, M. Natori and T. Nodera, eds., no. 9 in Parallel Processing for Scientific Computing, Keio University, Yokahama, Japan, 1993, pp. 32–47.

[2] P. Concus, G. H. Golub, and D. P. O'Leary, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, in Proceedings of the Symposium on Sparse Matrix Computations, J. R. Bunch and D. J. Rose, eds., New York, 1975, pp. 309–332.

[3] J. Drkošová, A. Greenbaum, M. Rozložník, and Z. Strakoš, *Numerical stability of GMRES method*, BIT, 35 (1995), pp. 308–330.

[4] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis, Dundee, 1975, G. A. Watson, ed., vol. 506 of Lecture Notes in Mathematics, Springer, Berlin, 1976, pp. 73–89.

[5] R. W. Freund and N. M. Nachtigal, *QMR: a quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.

[6] ——, *An implementation of the QMR method based on coupled two-term recurrences*, SIAM J. Sci. Comput., 15 (1994), pp. 313–337.

[7] J. F. Grcar, *Analyses of the Lanczos algorithm and of the approximation problem in Richardson's method*, PhD thesis, University of Illinois at Urbana-Champaign, 1981. Report No. UIUCDCS-R-81-1074.

[8] A. Greenbaum, *Predicting the behavior of finite precision Lanczos and conjugate gradient computations*, Linear Algebra Appl., 113 (1989), pp. 7–63.

[9] ——, *Accuracy of computed solutions from conjugate-gradient-like methods*, in Advances in Numerical Methods for Large Sparse Sets of Linear Systems, M. Natori and T. Nodera, eds., no. 10 in Parallel Processing for Scientific Computing, Keio University, Yokahama, Japan, 1994, pp. 126–138.

[10] ——, *Estimating the attainable accuracy of recursively computed residual methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 535–551.

[11] A. Greenbaum, M. Rozložník, and Z. Strakoš, *Numerical behaviour of the modified Gram-Schmidt GMRES implementation*, BIT, 37 (1997), pp. 706–719.

[12] A. Greenbaum and Z. Strakoš, *Predicting the behavior of finite precision Lanczos and conjugate gradient computations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 121–137.

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

[14] M. H. Gutknecht and Zdeněk Strakoš, *Accuracy of three-term and two-term recurrences for Krylov space solvers*, Swiss Center for Scientific Computing, 1997. Technical Report No. TR-97-21.

[15] L. Hageman and D. Young, *Applied Iterative Methods*, Academic Press, Orlando, 1981.

[16] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bureau Standards, 49 (1952), pp. 409–435.

[17] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bureau Standards, 45 (1950), pp. 255–281.

[18] T. A. Manteuffel, *The Tchebyshev iteration for nonsymmetric linear systems*, Numer. Math., 28 (1977), pp. 307–327.

[19] C. C. Paige, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, PhD Thesis, University of London, England 1971.

[20] C. C. Paige, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, J. Inst. Math. Appl., 18 (1976), pp. 341–349.

[21] C. C. Paige, *Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem*, Linear Algebra Appl., 34 (1980), pp. 235–258.

[22] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[23] J. K. Reid, *On the method of conjugate gradients for solution of large sparse systems of linear equations*, in Large Sparse Sets of Linear Equations, J. K. Reid, ed., Academic, London, 1971, pp. 231–254.

[24] H. Rutishauser, *Theory of gradient methods*, in Refined Iterative Methods for Computation of the Solution and the Eigenvalues of Self-Adjoint Boundary Value Problems, Mitt. Inst. angew. Math. ETH Zürich, Nr. 8, Birkhäuser, Basel, 1959, pp. 24–49.

[25] G. L. G. Sleijpen, H. A. van der Vorst, and D. R. Fokkema, *BiCGstab(l) and other hybrid Bi-CG methods*, Numerical Algorithms, 7 (1994), pp. 75–109.

[26] E. Stiefel, *Relaxationsmethoden bester Strategie zur Lösung linearer Gleichungssysteme*, Comm. Math. Helv., 29 (1955), pp. 157–179.

[27] D. M. Young and K. C. Jea, *Generalized conjugate-gradient acceleration of nonsymmetrizable iterative methods*, Linear Algebra Appl., 34 (1980), pp. 159–194.