SIAM J. SCI. COMPUT.
Vol. 14, No. 5, pp. 1020–1033, September 1993

VARIANTS OF BICGSTAB FOR MATRICES WITH COMPLEX SPECTRUM*

MARTIN H. GUTKNECHT†

Abstract. Recently Van der Vorst [*SIAM J. Sci. Statist. Comput.*, 13 (1992), pp. 631–644] proposed for solving nonsymmetric linear systems $Az = b$ a biconjugate gradient (BiCG)-based Krylov space method called BiCGStab that, like the biconjugate gradient squared (BiCGS) method of Sonneveld, does not require matrix–vector multiplications with the transposed matrix A^T, and that has typically a much smoother convergence behavior than BiCG and BiCGS. Its nth residual polynomial is the product of the one of BiCG (i.e., the nth Lanczos polynomial) with a polynomial of the same degree with real zeros. Therefore, nonreal eigenvalues of A are not approximated well by the second polynomial factor. Here, the author presents for real nonsymmetric matrices a method BiCGStab2 in which the second factor may have complex conjugate zeros. Moreover, versions suitable for complex matrices are given for both methods.

Key words. Lanczos algorithm, biconjugate gradient algorithm, conjugate gradient squared algorithm, BiCGStab, formal orthogonal polynomial, nonsymmetric linear system, Krylov space method

AMS subject classification. 65F10

1. From BiCG to complex BiCGStab. The *biconjugate gradient method* (BiCG) of Lanczos [7] and Fletcher [1] is a Krylov space method for solving (real or complex) non-Hermitian linear system $Az = b$, where A is, say, a nonsingular $N \times N$ matrix. (Typically, this matrix will be the result of applying a preconditioner to the original system matrix.) Starting from some initial guess z_0 for the solution, BiCG generates a sequence z_n with the property that the nth residual $r_n := b - Az_n$ lies in the Krylov space generated by A from r_0, i.e.,

$$r_n \in \mathcal{K}_{n+1} := \text{span}\,(r_0, Ar_0, \ldots, A^n r_0), \tag{1}$$

and is orthogonal to another Krylov space generated from some other initial vector y_0 by the Hermitian transpose A^H

$$r_n \perp \mathcal{L}_n := \text{span}\,(y_0, A^H y_0, \ldots, (A^H)^{n-1} y_0). \tag{2}$$

The sequence of residual polynomials ρ_n, which are implicitly defined by

$$r_n = \rho_n(A) r_0, \tag{3}$$

is in view of (2) a sequence of formal orthogonal polynomials: if we define a linear functional Φ on the space of polynomials with complex coefficients by setting $\Phi(\zeta^k) := y_0^H A^k x_0$, the formal orthogonality relation $\Phi(\pi_k \rho_n) = 0$ holds for every polynomial π_k of degree $k < n$; see [5], [3] for further details and references. As a consequence of the consistency condition for polynomial acceleration methods, these residual polynomials are normalized by $\rho_n(0) = 1$. They are often called *Lanczos polynomials*. In general, neither these polynomials nor the residuals satisfy a minimality condition, in contrast to the case in which A is Hermitian positive definite and $y_0 = r_0$, where the method reduces to the classical conjugate gradient method. Theoretically, the BiCG algorithm terminates in at most $\nu(A, r_0)$ steps if this number denotes the degree of the minimal polynomial of the restriction of A to the maximum Krylov space generated by A from

*Received by the editors September 9, 1991; accepted for publication (in revised form) August 17, 1992.
†Interdisciplinary Project Center for Supercomputing (IPS), ETH Zürich, ETH-Zentrum, CH-8092 Zürich, Switzerland (mhg@ips.ethz.ch).

r_0. In practice this property is often irrelevant because $\nu(A, r_0)$ may be very large and the orthogonality (2) is partly lost due to roundoff. What counts in practice is the good convergence behavior and the small memory requirement of the method.

The algorithm can break down, namely, due to

$$r_n \perp \mathcal{L}_{n+1}; \tag{4}$$

but in most cases such a breakdown or a corresponding near-breakdown can be overcome by a look-ahead step, see [8], [3], [4], [2] and further references cited there.

In the standard version of the BiCG method, which we call BioMin, a second sequence $\{\sigma_n\}$ of formal orthogonal polynomials plays a role. The algorithm is based on the following mixed recurrence formulas (in which we suppress the independent variable ζ by writing ρ_n instead of $\rho_n(\zeta)$, and σ_n instead of $\sigma_n(\zeta)$):

$$\rho_{n+1} := \rho_n - \omega_n \zeta \sigma_n, \tag{5a}$$

$$\sigma_{n+1} := \rho_{n+1} - \psi_{n+1} \sigma_n, \tag{5b}$$

where the coefficients ω_n and ψ_{n+1} are computed as indicated below. These formulas are implicitly used to generate two pairs of finite vector sequences

$$x_n := \rho_n(A)x_0, \quad y_n := \overline{\rho_n}(A^H)y_0, \tag{6a}$$

$$u_n := \sigma_n(A)x_0, \quad v_n := \overline{\sigma_n}(A^H)y_0, \tag{6b}$$

satisfying the biorthogonality conditions

$$y_n^H x_m = \delta_n \delta_{m,n}, \tag{7a}$$

$$v_n^H A u_m = \delta_n' \delta_{m,n}. \tag{7b}$$

(In (6), $\overline{\rho_n}$ and $\overline{\sigma_n}$ have the complex conjugate coefficients of ρ_n and σ_n, and in (7), $\delta_n \neq 0$, $\delta_n' \neq 0$ are constants and $\delta_{m,n}$ is the Kronecker symbol. The scaling of the vectors y_n and v_n is, theoretically, irrelevant; but in practice, a scaling different from the one chosen here may be more appropriate, e.g., one may choose y_n and v_n to have unit length.)

Theoretically one proceeds until the algorithm terminates or breaks down; in practice, one stops when the residual is sufficiently small. Here, $x_n = r_n$ is actually the nth residual vector of BiCG, but as we will see, there are other algorithms that are also based on the recursions (5) yet have other residual vectors. In any case, the recurrences (5) are not used to generate the polynomials, but are translated into recurrences for the vectors specified by (6). For the iterates z_n, an additional recurrence related to the one for the residual vectors $x_n = r_n$ is easily derived. The coefficients ω_n and ψ_{n+1} in (5) are determined from the biorthogonality conditions (7) and the consistency condition $\rho_n(0) = 1$ (see [5]); in fact, only the case $m = n + 1$ of (7) is enforced:

$$\omega_n := \delta_n/(y_n^H A u_n) = \delta_n/(v_n^H A u_n), \tag{8a}$$

$$\delta_{n+1} := y_{n+1}^H x_{n+1}, \tag{8b}$$

$$\psi_{n+1} := (y_{n+1}^H A u_n)/(v_n^H A u_n) = -\delta_{n+1}/\delta_n. \tag{8c}$$

(In the equalities in (8a) and (8c), one makes use of the formal orthogonality of the polynomials and of the fact that, in view of (5), ρ_{n+1}, σ_{n+1}, and $-\omega_n \zeta \sigma_n$ have the same leading coefficient.)

It is the great advantage of the Lanczos approach based on (5) that observing the conditions (7) for $m = n+1$ implies theoretically that they hold for all $m \leq n$. The short recurrences (5) mean very small memory requirements. Moreover, although in practice the biorthogonality with respect to the earlier constructed vectors (i.e., for large $|m-n|$) tends to get lost due to roundoff, it is this biorthogonality that accounts for the normally observed gradual decrease of the residual length. The biorthogonality is related to a Padé approximation problem. However, the convergence is still not well understood, and in practice is often far from monotonic. But for very large problems, Lanczos-type methods, due to their small memory requirements, are often the best choice.

One disadvantage of the unsymmetric Lanczos process and BiCG is that the construction of y_n requires one application of A^H in each step in addition to the one application of A to compute z_n and r_n. (In BiCG, u_n and v_n are then found without further matrix–vector products.) In 1984 Sonneveld [11] with his *(bi)conjugate gradient squared* (CGS or BiCGS) *method* proposed a way to get around A^H. He uses a set of recurrences that allows him to compute other iterates z_n whose residuals $r_n := b - Az_n$ satisfy

$$r_n = \rho_n^2(A)r_0, \tag{9}$$

where the polynomials ρ_n are still the Lanczos polynomials. In fact, the iterates and their residuals, together with the vectors defined by

$$p_n := \rho_n(A)\sigma_n(A)r_0, \quad q_n := \rho_n(A)\sigma_{n-1}(A)r_0, \quad s_n := \sigma_n^2(A)r_0, \tag{10}$$

can be generated by the Krylov space analogs of the polynomial recurrences

$$\rho_{n+1}\sigma_n = \rho_n\sigma_n - \omega_n\zeta\sigma_n^2, \tag{11a}$$

$$\rho_{n+1}\sigma_{n+1} = \rho_{n+1}^2 - \psi_{n+1}\rho_{n+1}\sigma_n, \tag{11b}$$

$$\rho_{n+1}^2 = \rho_n^2 - \omega_n\zeta(\rho_n\sigma_n + \rho_{n+1}\sigma_n), \tag{11c}$$

$$\sigma_{n+1}^2 = \rho_{n+1}\sigma_{n+1} - \psi_{n+1}\rho_{n+1}\sigma_n + \psi_{n+1}^2\sigma_n^2, \tag{11d}$$

which are readily derived from (5). Due to the squaring of the residual polynomial, the sometimes erratic convergence behavior of BiCG is even more pronounced here.

Of course, the recurrences (11) need to be complemented by formulas for determining ω_n and ψ_{n+1}. Using (6) and (10), the expressions (8) are readily transformed into

$$\omega_n = \delta_n/(y_0^H As_n), \tag{12a}$$

$$\delta_{n+1} = y_0^H r_{n+1}, \tag{12b}$$

$$\psi_{n+1} = -\delta_{n+1}/\delta_n. \tag{12c}$$

As mentioned before, these coefficients ω_n and ψ_{n+1} result basically from observing the biorthogonality conditions (7) for $m = n+1$, i.e.,

$$y_n^H x_{n+1} = 0, \qquad v_n^H Au_{n+1} = 0. \tag{13}$$

At this point it is worth noting that for $m \neq n$, (7) is equivalent to

$$y_n \perp \mathcal{K}_n, \ x_n \perp \mathcal{L}_n, \tag{14a}$$

$$A^H v_n \perp \mathcal{K}_n, \ Au_n \perp \mathcal{L}_n. \tag{14b}$$

In view of (6) these conditions are not independent; those on the left-hand side are equivalent to those on the right-hand side. If $\mathcal{P}_n$ denotes the set of polynomials of degree at most n, the latter can be expressed as

$$y_0^H \pi_n(A) x_{n+1} = 0 \ (\forall \pi_n \in \mathcal{P}_n), \tag{15a}$$

$$v_0^H \tilde{\pi}_n(A) A u_{n+1} = 0 \ (\forall \tilde{\pi}_n \in \mathcal{P}_n). \tag{15b}$$

The conditions (13) are just the special case of (15) with $\pi_n = \rho_n$ and $\tilde{\pi}_n = \sigma_n$. But the same restrictions can be taken into account by choosing for π_n and $\tilde{\pi}_n$ any other polynomial of exact degree n. Van der Vorst [14] (following an earlier proposal by Sonneveld) discovered that a simple implementation and an often excellent convergence behavior can be attained by choosing for π_n and $\tilde{\pi}_n$ polynomials τ_n that are built up in factored form

$$\tau_n(\zeta) = (1 - \chi_0 \zeta)(1 - \chi_1 \zeta) \cdots (1 - \chi_{n-1} \zeta) \tag{16}$$

by adding a suitable new zero $1/\chi_n$ at each step. Clearly,

$$\tau_{n+1} = (1 - \chi_n \zeta) \tau_n, \tag{17}$$

so that

$$\rho_{n+1} \tau_{n+1} = (1 - \chi_n \zeta) \rho_{n+1} \tau_n. \tag{18}$$

Multiplication of (5a) and (5b) by τ_n and τ_{n+1} yields the further recurrences

$$\rho_{n+1} \tau_n = \rho_n \tau_n - \omega_n \zeta \sigma_n \tau_n, \tag{19}$$

$$\sigma_{n+1} \tau_{n+1} = \rho_{n+1} \tau_{n+1} - \psi_{n+1}(1 - \chi_n \zeta) \sigma_n \tau_n. \tag{20}$$

Equations (18)–(20) are a set of recurrence relations that after the translation from polynomials to Krylov space vectors can be used to build up the vector sequences

$$\tilde{r}_n := \rho_n(A) \tau_n(A) r_0, \quad \tilde{w}_n := \rho_n(A) \tau_{n-1}(A) r_0, \quad \tilde{s}_n := \sigma_n(A) \tau_n(A) r_0. \tag{21}$$

Van der Vorst [14], assuming real data (i.e., a real matrix A and $b, z_0, y_0 \in \mathbb{R}^N$), chooses in the nth step of BICGSTAB the new zero $1/\chi_n$ so that the new residual norm $||\tilde{r}_{n+1}|| = ||\tilde{w}_{n+1} - A\tilde{w}_{n+1}\chi_n||$ is minimized over $\chi_n \in \mathbb{R}$. Here and in the following, the norm is the Euclidean one. Our first aim is to generalize this approach to complex data. In particular, we allow $\chi_n \in \mathbb{C}$ now. Starting from

$$||\tilde{r}_{n+1}||^2 = ||\tilde{w}_{n+1} - A\tilde{w}_{n+1}\chi_n||^2 = ||\tilde{w}_{n+1}||^2 - 2\,\mathrm{Re}\,\{\tilde{w}_{n+1}^H A \tilde{w}_{n+1} \chi_n\} + \chi_n^2 ||A\tilde{w}_{n+1}||^2, \tag{22}$$

one can conclude by inserting $\chi_n = \xi_n + i\eta_n$, and computing the partial derivatives $\partial/\partial\xi_n$ and $\partial/\partial\eta_n$, that this minimization leads to

$$\chi_n := \frac{(A\tilde{w}_{n+1})^H \tilde{w}_{n+1}}{||A\tilde{w}_{n+1}||^2}. \tag{23}$$

This is no surprise, since one obtains the same formula easier by noting that the minimization problem can be solved by orthogonal projection of $\tilde{w}_{n+1}$ onto $A\tilde{w}_{n+1}$ in $\mathbb{C}^N$; the image of this projection must be equal to $A\tilde{w}_{n+1}\chi_n$. (If we had restricted χ_n to be real, we would have obtained $\chi_n := \mathrm{Re}\,\{(A\tilde{w}_{n+1})^H \tilde{w}_{n+1}\}/||A\tilde{w}_{n+1}||^2$ instead.)

Formulas for the coefficients ω_n and ψ_{n+1} are found by modifying (12) for taking the leading coefficient $(-1)^n\chi_0\cdots\chi_{n-1}$ of τ_n into account. Noting that σ_n and ρ_n have leading coefficient $(-1)^n\omega_0\cdots\omega_{n-1}$ and setting $\tilde{\delta}_n := \delta_n(\chi_0\cdots\chi_{n-1})/(\omega_0\cdots\omega_{n-1})$, we get

$$\omega_n = \frac{\delta_n}{y_0^H A s_n} = \frac{\delta_n}{y_0^H A\sigma_n^2(A)x_0} = \frac{\tilde{\delta}_n}{y_0^H A\sigma_n(A)\tau_n(A)x_0} = \frac{\tilde{\delta}_n}{y_0^H A\tilde{s}_n}, \tag{24a}$$

$$\tilde{\delta}_{n+1} = y_0^H\tilde{r}_{n+1}, \tag{24b}$$

$$\psi_{n+1} = -\frac{\delta_{n+1}}{\delta_n} = \frac{\tilde{\delta}_{n+1}\omega_n}{\tilde{\delta}_n\chi_n}. \tag{24c}$$

Altogether, one obtains the following complex version of BiCGStab. If applied to real data, it is identical with Van der Vorst's algorithm. The letter o denotes the zero vector.

ALGORITHM 1 (BiCGStab). *For solving $Az = b$ choose an initial approximation $z_0 \in \mathbb{C}^N$ and set $\tilde{r}_0 := \tilde{s}_0 := b - Az_0$. Choose $y_0 \in \mathbb{C}^N$ such that $\tilde{\delta}_0 := y_0^H\tilde{r}_0 \neq 0$ and $\varphi_0 := y_0^H A\tilde{s}_0/\tilde{\delta}_0 \neq 0$. Then compute for $n = 0, 1, \ldots$*

$$\omega_n := 1/\varphi_n\,, \tag{25a}$$

$$\tilde{w}_{n+1} := \tilde{r}_n - A\tilde{s}_n\omega_n\,, \tag{25b}$$

$$\chi_n := (A\tilde{w}_{n+1})^H\tilde{w}_{n+1}/\|A\tilde{w}_{n+1}\|^2\,, \tag{25c}$$

$$\tilde{r}_{n+1} := \tilde{w}_{n+1} - A\tilde{w}_{n+1}\chi_n\,, \tag{25d}$$

$$z_{n+1} := z_n + \tilde{s}_n\omega_n + \tilde{w}_{n+1}\chi_n\,, \tag{25e}$$

$$\tilde{\delta}_{n+1} := y_0^H\tilde{r}_{n+1}\,, \tag{25f}$$

$$\psi_{n+1} := -\omega_n\tilde{\delta}_{n+1}/(\tilde{\delta}_n\chi_n)\,, \tag{25g}$$

$$\tilde{s}_{n+1} := \tilde{r}_{n+1} - (\tilde{s}_n - A\tilde{s}_n\chi_n)\,\psi_{n+1}\,, \tag{25h}$$

$$\varphi_{n+1} := y_0^H A\tilde{s}_{n+1}/\tilde{\delta}_{n+1}\,. \tag{25i}$$

If $\tilde{r}_{n+1} = o$, the process terminates and z_{n+1} is the solution of $Az = b$; if $\tilde{r}_{n+1} \neq o$ but $\tilde{\delta}_{n+1} = 0$ or $\varphi_{n+1} = 0$, the algorithm breaks down.

We need to comment on the conditions under which this algorithm breaks down. Clearly, in exact arithmetic, a breakdown of BiCG, caused by $\varphi_n = 0$ or $\delta_n = 0$, see [5], is paralleled by a breakdown of BiCGStab, caused by $\varphi_n = 0$ or $\tilde{\delta}_n = 0$, since $\tilde{\delta}_n := \delta_n(\chi_0\cdots\chi_{n-1})/(\omega_0\cdots\omega_{n-1})$. However, in the above formulation, BiCGStab obviously also breaks down if $\chi_n = 0$. A closer look shows that ψ_{n+1} may still be finite in this case, cf. (24c), so that one might try to find an alternative formula for (25g). However, if $\chi_n = 0$, then $\tilde{r}_{n+1} = \tilde{w}_{n+1} = \rho_{n+1}(A)\tau_n(A)r_0$, and it follows from (15a) that $\tilde{\delta}_{n+1} := y_0^H\tilde{r}_{n+1} = y_0^H\rho_{n+1}(A)\tau_n(A)r_0 = y_0^H\tau_n(A)x_{n+1} = 0$. Consequently, $\chi_n = 0$ implies $\tilde{\delta}_{n+1} = 0$, so it suffices to check the two conditions $\tilde{\delta}_{n+1} = 0$ and $\varphi_{n+1} = 0$, although this conceals the fact that $\tilde{\delta}_{n+1} = 0$ can be caused by $\delta_{n+1} = 0$ or $\chi_n = 0$. The new pitfall here is that we do not advance in the Krylov space if $\chi_n = 0$. Then $\tilde{r}_{n+1} \in \mathcal{K}_{2n+2}$, but $\tilde{r}_{n+1} = \rho_{n+1}(A)\tau_{n+1}(A)r_0$ should be in $\mathcal{K}_{2n+3}\backslash\mathcal{K}_{2n+2}$.

2. Two-dimensional local minimization: BiCGStab2. As is well known and is indicated by (3), the convergence of a polynomial acceleration method like BiCG hinges on the damping properties of the residual polynomials ρ_n. Ideally, the lemniscates $|\rho_n(\zeta)| = \varepsilon$ should already for some small ε embrace the spectrum of A or, in the case of a nonnormal A, rather the pseudospectrum [12]. The residual polynomials $\rho_n\tau_n$ of BiCGStab

combine the damping properties of the Lanczos polynomials ρ_n (which not only depend on A but also on $r_0 = x_0$ and y_0) with those of τ_n. If one had $\rho_n \equiv 1$ for all n (which is impossible since ρ_n has exact degree n), the polynomials τ_n would be the residual polynomials of GMRES(1) [10], i.e., of GMRES restarted at each step. The factor ρ_n causes a modification of the restart vector, but in any case, one cannot expect τ_n to have an excellent global damping effect. Although the success of BICGSTAB shows that τ_n is normally good enough to level out the irregular convergence of the Lanczos polynomials ρ_n, it has an obvious deficiency when the method (in its original version proposed by Van der Vorst) is applied to a real nonsymmetric system: while generally a nonsymmetric real matrix A has a complex spectrum, all the zeros of τ_n are real if z_0 and y_0 are real vectors.[1] It is therefore natural to try to modify the method so that in the real case τ_n may have pairs of complex conjugate zeros. This is the basic idea for the method BICGSTAB2 defined next. We formulate it also for complex data, but the reader must keep in mind that it also brings a major improvement in the real case.

Let us redefine the polynomials τ_n according to $\tau_0 :\equiv 1$ and the recurrences

$$\tau_{2m+1} := (1 - \chi_m\zeta)\tau_{2m}, \tag{26a}$$

$$\tau_{2m+2} := (1 - \xi_m)\tau_{2m} + (\xi_m + \eta_m\zeta)\tau_{2m+1}, \tag{26b}$$

with $\chi_m, \xi_m, \eta_m \in \mathbb{C}$. (In the case of real data, $\chi_m, \xi_m, \eta_m \in \mathbb{R}$.) Note that $\tau_n(0) = 1$ (for all n) by induction. Clearly, τ_{2m+1} has, as in the original BICGSTAB, the new zero $1/\chi_m$, but in the next step τ_{2m+2} is chosen as a linear combination of τ_{2m}, τ_{2m+1} and $\zeta\tau_{2m+1}$, restricted only by $\tau_{2m+2}(0) = 1$. Hence, the zero $1/\chi_m$ is dismissed and τ_{2m} is supplemented by two new zeros, i.e., there holds

$$\tau_{2m+2} = (1 - \zeta_{2m}\zeta)(1 - \zeta_{2m+1}\zeta)\tau_{2m}, \tag{27}$$

where in the case of real data ζ_{2m} and ζ_{2m+1} are either real or complex conjugate. These zeros can, but need not, be computed. The parameters χ_m, ξ_m, and η_m are again chosen to minimize the residual locally (i.e., within a one- or two-dimensional subspace, respectively). But first, we want to look at the recurrences, which are actually independent of these choices.

Clearly, (19) is again valid for all n, and an analogous relation follows by multiplying (5a) with τ_{n-1} instead of τ_n:

$$\rho_{n+1}\tau_n = \rho_n\tau_n - \omega_n\zeta\sigma_n\tau_n, \tag{28}$$

$$\rho_{n+1}\tau_{n-1} = \rho_n\tau_{n-1} - \omega_n\zeta\sigma_n\tau_{n-1} \quad \text{if } n \geq 1. \tag{29}$$

Equations (18) and (20) remain correct for even n after replacing χ_n by χ_m:

$$\rho_{n+1}\tau_{n+1} = (1 - \chi_m\zeta)\rho_{n+1}\tau_n \quad \text{if } n = 2m, \tag{30}$$

$$\sigma_{n+1}\tau_{n+1} = \rho_{n+1}\tau_{n+1} - \psi_{n+1}(1 - \chi_m\zeta)\sigma_n\tau_n \quad \text{if } n = 2m. \tag{31}$$

When n is odd, one obtains instead by using (26b):

$$\rho_{n+1}\tau_{n+1} = (1 - \xi_m)\rho_{n+1}\tau_{n-1} + \xi_m\rho_{n+1}\tau_n + \eta_m\zeta\rho_{n+1}\tau_n \quad \text{if } n = 2m + 1. \tag{32}$$

$$\begin{aligned}\sigma_{n+1}\tau_{n+1} &= \rho_{n+1}\tau_{n+1} - \psi_{n+1}\sigma_n\tau_{n+1} \\ &= \rho_{n+1}\tau_{n+1} - \psi_{n+1}[(1 - \xi_m)\sigma_n\tau_{n-1} + (\xi_m + \eta_m\zeta)\sigma_n\tau_n] \\ &\qquad \text{if } n = 2m + 1.\end{aligned} \tag{33}$$

[1] This is analagous to the fact that Newton's method will never find a complex zero of a real polynomial when started on the real axis.

Finally, multiplying (5b) by τ_n, we get for all n

$$\sigma_{n+1}\tau_n = \rho_{n+1}\tau_n - \psi_{n+1}\sigma_n\tau_n. \tag{34}$$

The relations (28)–(34) are a set of recurrence formulas for the products $\rho_n\tau_n$, $\rho_n\tau_{n-1}$, $\rho_n\tau_{n-2}$, $\sigma_n\tau_n$, and $\sigma_n\tau_{n-1}$. Hence, we must build up the vector sequences

$$\text{(35a)}\quad \tilde{r}_n := \rho_n(A)\tau_n(A)r_0, \quad \tilde{w}_n := \rho_n(A)\tau_{n-1}(A)r_0, \quad \tilde{\tilde{w}}_n := \rho_n(A)\tau_{n-2}(A)r_0,$$
$$\text{(35b)}\quad \tilde{s}_n := \sigma_n(A)\tau_n(A)r_0, \quad \tilde{t}_n := \sigma_n(A)\tau_{n-1}(A)r_0.$$

The coefficients ω_n and ψ_{n+1} are found by adapting (24). Since (19) and (28) are identical, i.e., since $\tilde{w}_{n+1}$ is given by the same formula (25b) as in BiCGStab, (24a) and (24b) still hold. Because the leading coefficient of τ_n is now $(-1)^m\chi_0\eta_0\cdots\chi_{m-1}\eta_{m-1}$ if $n = 2m$ and $(-1)^m\chi_0\eta_0\cdots\chi_{m-1}\eta_{m-1}\chi_m$ if $n = 2m+1$, (24c) transforms into

$$\psi_{n+1} = -\frac{\delta_{n+1}}{\delta_n} = \begin{cases} -(\tilde{\delta}_{n+1}\omega_n)/(\tilde{\delta}_n\chi_m) & \text{if } n = 2m, \\ +(\tilde{\delta}_{n+1}\omega_n)/(\tilde{\delta}_n\eta_m) & \text{if } n = 2m+1. \end{cases} \tag{36}$$

Finally, we need to give formulas for the parameters χ_m, ξ_m, and η_m, which are determined by a one- and a two-dimensional minimization problem, respectively, cf. (26), (30), (32), and (35):

$$\text{(37a)}\quad \|\tilde{r}_{n+1}\| = \|\tilde{w}_{n+1} - A\tilde{w}_{n+1}\chi_m\| = \min \quad \text{if } n = 2m,$$
$$\text{(37b)}\quad \|\tilde{r}_{n+1}\| = \|\tilde{\tilde{w}}_{n+1} + (\tilde{w}_{n+1} - \tilde{\tilde{w}}_{n+1})\xi_m + A\tilde{w}_{n+1}\eta_m\| = \min \quad \text{if } n = 2m+1.$$

Of course, these minimization problems are again solved by orthogonal projection: as before in (23),

$$\chi_m := \frac{(A\tilde{w}_{n+1})^H\tilde{w}_{n+1}}{\|A\tilde{w}_{n+1}\|^2} \quad \text{if } n = 2m. \tag{38}$$

For the projection onto the two-dimensional subspace spanned by $\tilde{w}_{n+1} - \tilde{\tilde{w}}_{n+1}$ and $A\tilde{w}_{n+1}$, we define the two-column matrix

$$B_{m+1} := \left[\tilde{w}_{n+1} - \tilde{\tilde{w}}_{n+1} \,|A\tilde{w}_{n+1}\right] \quad \text{if } n = 2m+1, \tag{39a}$$

in terms of which the projection of $\tilde{\tilde{w}}_{n+1}$ is given by $B_{m+1}(B_{m+1}^HB_{m+1})^{-1}B_{m+1}^H\tilde{\tilde{w}}_{n+1}$. Hence, the optimal coefficients ξ_m and η_m for (37b) are

$$\begin{bmatrix} \xi_m \\ \eta_m \end{bmatrix} := -(B_{m+1}^HB_{m+1})^{-1}B_{m+1}^H\tilde{\tilde{w}}_{n+1} \quad \text{if } n = 2m+1. \tag{39b}$$

The 2×2 matrix $B_{m+1}^HB_{m+1}$ can be singular, namely, when the vectors $\tilde{w}_{n+1} - \tilde{\tilde{w}}_{n+1}$ and $A\tilde{w}_{n+1}$ are linearly dependent. The solution (ξ_m, η_m) of the minimization problem (37b) is then not unique, and one might want to compute a particular one in a regularized fashion. However, since $\tilde{w}_{n+1} - \tilde{\tilde{w}}_{n+1} \in \mathcal{K}_{2n+2}$, this can only happen if $A\tilde{w}_{n+1}$ is in the same space, which again means that we do no longer advance in the Krylov space. Here, assuming no previous breakdown, we can even conclude that $\mathcal{K}_{2n+3} = \mathcal{K}_{2n+2}$ since the polynomial $\zeta\rho_{n+1}(\zeta)\tau_n(\zeta)$ that corresponds to $A\tilde{w}_{n+1}$ has exact degree $2n+2$. In view

of $\tilde{r}_{n+1} \in \mathcal{K}_{2n+2}$, we conclude as in the discussion at the end of §1 that $\tilde{\delta}_{n+1} = 0$ due to (15a). The same is true if the minimization problem (37b) has a unique solution, but the second component η_m happens to be zero.

Summarizing we obtain the following two-step algorithm BiCGStab2 for real nonsymmetric or complex linear systems.

ALGORITHM 2. (BiCGStab2) *For solving $Az = b$ choose an initial approximation $z_0 \in \mathbb{C}^N$ and set $\tilde{r}_0 := \tilde{s}_0 := b - Az_0$. Choose $y_0 \in \mathbb{C}^N$ such that $\delta_0 := y_0^H \tilde{r}_0 \neq 0$ and $\varphi_0 := y_0^H A\tilde{s}_0/\delta_0 \neq 0$. Then compute for $n = 0, 1, \ldots$*

(40a) $\omega_n := 1/\varphi_n,$

(40b) $\tilde{\tilde{w}}_{n+1} := \tilde{w}_n - A\tilde{t}_n\omega_n \quad (\text{if } n \geq 1),$

(40c) $\tilde{w}_{n+1} := \tilde{r}_n - A\tilde{s}_n\omega_n;$

if n is even, set $m := n/2$, compute χ_m by (38) *and let*

(40d) $\tilde{r}_{n+1} := \tilde{w}_{n+1} - A\tilde{w}_{n+1}\chi_m,$

(40e) $z_{n+1} := z_n + \tilde{s}_n\omega_n + \tilde{w}_{n+1}\chi_m,$

(40f) $\tilde{\delta}_{n+1} := y_0^H \tilde{r}_{n+1},$

(40g) $\psi_{n+1} := -\omega_n\tilde{\delta}_{n+1}/(\tilde{\delta}_n\chi_m),$

(40h) $\tilde{s}_{n+1} := \tilde{r}_{n+1} - (\tilde{s}_n - A\tilde{s}_n\chi_m)\psi_{n+1},$

else set $m := (n-1)/2$, compute ξ_m and η_m by (39) *and let*

(40i) $\tilde{r}_{n+1} := \tilde{\tilde{w}}_{n+1}(1 - \xi_m) + \tilde{w}_{n+1}\xi_m + A\tilde{w}_{n+1}\eta_m,$

(40j) $z_{n+1} := [z_{n-1} + \tilde{s}_{n-1}\omega_{n-1} + \tilde{t}_n\omega_n](1 - \xi_m) + [z_n + \tilde{s}_n\omega_n]\xi_m - \tilde{w}_{n+1}\eta_m,$

(40k) $\tilde{\delta}_{n+1} := y_0^H \tilde{r}_{n+1},$

(40l) $\psi_{n+1} := \omega_n\tilde{\delta}_{n+1}/(\tilde{\delta}_n\eta_m),$

(40m) $\tilde{s}_{n+1} := \tilde{r}_{n+1} - [\tilde{t}_n(1 - \xi_m) + \tilde{s}_n\xi_m + A\tilde{s}_n\eta_m]\psi_{n+1}$

endif;

(40n) $\tilde{t}_{n+1} := \tilde{w}_{n+1} - \tilde{s}_n\psi_{n+1},$

(40o) $A\tilde{t}_{n+1} := A\tilde{w}_{n+1} - A\tilde{s}_n\psi_{n+1},$

(40p) $\varphi_{n+1} := y_0^H A\tilde{s}_{n+1}/\tilde{\delta}_{n+1}.$

If $\tilde{r}_{n+1} = o$, the process terminates and z_{n+1} is the solution of $Az = b$; if $\tilde{r}_{n+1} \neq o$ but $\tilde{\delta}_{n+1} = 0$ or $\varphi_{n+1} = 0$, the algorithm breaks down.

Like BiCGS and BiCGStab this algorithm requires two applications of A per step, i.e., one per degree of the residual polynomial, namely for $A\tilde{s}_n$ in (40c) and for $A\tilde{w}_{n+1}$ in (40d) or (40i). $A\tilde{t}_{n+1}$ is then obtained in (40o). In case of real data all the computation remains real. But even in the complex case BiCGStab2 is an improvement over BiCGStab since one performs a two-dimensional residual minimization in each other step. Of course, one could try to accomplish even higher dimensional minimizations in this framework, but clearly this would further complicate the algorithm and increase the memory requirement.

The methods BiCGStab and BiCGStab2 are examples of a class of methods that one might call *product methods* and which are characterized by residual polynomials that are the product of residual polynomials emerging from two different methods. Here, one factor is a Lanczos polynomial, while the other one comes from a one- or two-step minimal residual approach. Such product methods are truly *hybrid* methods, and deserve

this name much more than many of the other approaches to which this notion has been applied. Note that the recursions of BiCGStab hold whenever the polynomials τ_n can be updated according to (16), and those of BiCGStab2 are valid whenever these polynomials satisfy (26). Only the definitions of the parameters χ_m, ξ_m, and η_m must be replaced.

For example, one might try a hybrid method that first applies another Krylov space method, say, GMRes, until it becomes too expensive; after computing a real factorization of the obtained residual polynomial τ_M of, say, degree M, one could from then on use this polynomial and its powers to create residuals of the form $r_{kM} = \rho_{(k-1)M}(A)\tau_M^k(A)r_0$ and intermediate ones where not yet all factors of τ_M appear with multiplicity k. The factored form of the GMRes residual τ_M yields recurrences (27) that can be reformulated to conform with (26). First experiments on this approach gave promising results, but not as good as those of BiCGStab2.

Moreover, if in generalization of (26) the polynomials τ_n satisfy a general three-term recursion that takes the condition $\tau_n(0) = 1$ into account, i.e., if

$$\tau_{n+1} := (1 - \xi_n)\tau_{n-1} + (\xi_n + \eta_n\zeta)\tau_n, \tag{41}$$

then the system of recurrences (28), (29), (32)–(34) is still valid if we replace χ_m, η_m by χ_n, η_n in (32)–(34). In the Krylov space these recurrences turn into (40c), (40b), (40i), (40m), and (40n), respectively, and they also yield (40j) and (40o). Consequently, (40a)–(40c) and (40i)–(40p) also yield a realization of a product method in which the Lanczos polynomials are combined with a sequence of polynomials τ_n satisfying a three-term recursion (41). For example, one could use shifted and scaled Chebyshev polynomials here to obtain a *Lanczos–Chebyshev method.*

It is known, see, e.g., [5], that the zeros of ρ_n and σ_n can be determined from the coefficients $\psi_1, \ldots, \psi_{n-1}$ and $\omega_0, \ldots, \omega_{n-1}$. In fact, if

$$L_n := \begin{bmatrix} \omega_0^{-1} & & & \\ -\omega_0^{-1} & \omega_1^{-1} & & \\ & \ddots & \ddots & \\ & & -\omega_{n-2}^{-1} & \omega_{n-1}^{-1} \end{bmatrix}, \quad R_n := \begin{bmatrix} 1 & \psi_1 & & \\ & 1 & \ddots & \\ & & \ddots & \psi_{n-1} \\ & & & 1 \end{bmatrix},$$

then the eigenvalues of the tridiagonal matrix L_nR_n are the zeros of ρ_n, and those of the $n \times n$ leading principal submatrix of the tridiagonal matrix $R_{n+1}L_{n+1}$ are the zeros of σ_n. The zeros of either of the two polynomials are often considered as approximations of eigenvalues of A. (This is, basically, the approach of the unsymmetric Lanczos algorithm for finding eigenvalues of A.) Experiments indicate, however, that the coefficients ψ_n and ω_n produced by BiCGS and BiCGStab are less accurate than those found by BiCG or by the Lanczos biorthogonalization algorithm (the "unsymmetric" Lanczos algorithm).

In BiCGStab2, the zeros $1/\zeta_j$ of the polynomials τ_n are, if $n = 2m$, the zeros of the quadratic factors

$$1 + (\eta_k - \chi_k\xi_k)\zeta - \eta_k\xi_k\zeta^2, \quad k = 0, \ldots, m-1. \tag{42}$$

These zeros can at best in a very vague way be considered as approximations to individual eigenvalues of A. Note in particular that they have no influence on the polynomials ρ_n and σ_n, and, a fortiori, on their zeros.

3. Numerical examples. First, we present several examples with non-Hermitian banded Toeplitz matrices of order $N = 200$. For these matrices, the behavior of the spectrum in the limit $N \to \infty$ is known [6], [13], but this is nearly irrelevant here. What counts for the convergence of iterative methods is the ε-pseudospectrum Λ_ε, which, when N is large and ε is small, is according to Reichel and Trefethen [9] approximately equal to the following union of three sets:

$$\Lambda_\varepsilon \equiv (\Lambda + \Delta_\varepsilon) \cup \Omega_r \cup \Omega^R. \tag{43}$$

Here, $\Lambda + \Delta_\varepsilon$ denotes the exact spectrum with disks of radius ε around each eigenvalue. To describe Ω_r and Ω^R, we need to look at the images $\phi(S_r)$ and $\phi(S_R)$ of the circles of radius r and R, respectively, under the mapping by the symbol ϕ of the Toeplitz matrix. Ω_r and Ω^R contain the points $\zeta \in \mathbb{C}$ with respect to which $\phi(S_r)$ and $\phi(S_R)$ have positive and negative, respectively, winding number. The radii $r < 1$ and $R > 1$ depend on ε and N according to $r := (\varepsilon/c)^{1/N}$ and $R := (\varepsilon/C)^{-1/N}$, where c and C are some constants, which for the plots in [9] have been set to 1. See Reichel and Trefethen [9] for details and for plots corresponding to some of our examples.

Banded Toeplitz matrices are of relevance in applications, since the discretization of partial differential equations often leads to such a matrix or a low-rank modification of one.

Example 1. Let us first consider the tridiagonal matrix

$$A := \begin{bmatrix} 4 & -2 & & \\ 1 & 4 & -2 & \\ & 1 & 4 & \ddots \\ & & \ddots & \ddots \end{bmatrix} \tag{44}$$

with the symbol $\phi(\zeta) = -2\zeta + 4 + \zeta^{-1}$. Its (unimportant) exact spectrum lies on the complex interval $[4 - 2i\sqrt{2}, 4 + 2i\sqrt{2}]$. The image $f(S_1)$, which is the boundary of the (continuous) spectrum of the associated Toeplitz operator with $N = \infty$ is an ellipse with foci $4 \pm 2i\sqrt{2}$, major semiaxis 3 and minor semiaxis 1. The ε-pseudospectrum is the interior of a slightly smaller ellipse with the same foci. Hence, Chebyshev iteration or second-order Richardson iteration, adapted to the family of ellipses with these foci, would have asymptotically optimal linear convergence. It is therefore no surprise that the generalized minimal residual method GMRES converges also approximately linearly.[2] As right-hand side b of the system $Az = b$ and for the initial vectors z_0 and y_0 we choose random vectors. (Of course, the same vectors are used for all the methods tested.)

In Fig. 1 we display the residual norm convergence of BIOMIN(= BICG), BIOMINSQ (= BICGS = CGS), BICGSTAB, the new algorithm BICGSTAB2, and a brute force implementation of GMRES(∞) (= GCR). The numbers on the x-axis give the iteration count, except for GMRES, where only every other iteration is counted. In GMRES each iteration requires just one application of A, in contrast to the other methods where multiplications both by A and A^H are needed; hence, it is fair to divide the numbers of iteration of GMRES by two, although, on the other hand, the long recurrences of GMRES(∞) become very expensive with respect to memory and arithmetic operations when n becomes large.

[2]The author is indebted to L. N. Trefethen for this interpretation of his numerical result.

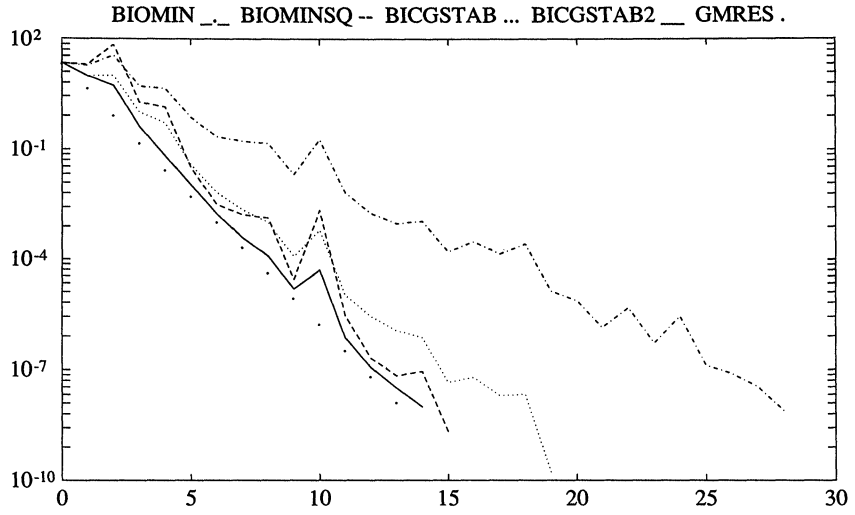

FIG. 1. *The residual norm history (i.e.,* $||r_n||/||r_0||$ *vs.* n*) for a linear system with the matrix* (44) *of order* 200, *solved by* BIOMIN = BICG (*dotted-dashed*), BIOMINSQ = BICGS = CGS (*dashed*), BICGSTAB (*dotted*), BICGSTAB2 (*solid*), *and* GMRES(∞) = GCR (*dotted*). *For* GMRES(∞) *only every other iteration is shown and counted, i.e.,* $||r_{2n}||/||r_0||$ *is plotted.*

In this example the convergence of all methods is rather fast, but it is interesting that the squared method BICGSTAB and BICGSTAB2 are considerably faster than BICG, and nearly as fast as GMRES(∞), which is optimal with respect to the measure of residual norm used in our figures. The same general behavior will be noted in our other examples. In this first example, where the matrix is real but the spectrum is complex, BICGSTAB2 is clearly better than the competing methods that require the same amount of work.

Example 2. As our second example we take

$$A := \begin{bmatrix} 2 & 1 & & & \\ 0 & 2 & 1 & & \\ 1 & 0 & 2 & 1 & \\ & 1 & 0 & 2 & \ddots \\ & & \ddots & \ddots & \ddots \end{bmatrix} \tag{45}$$

with the symbol $\phi(\zeta) = \zeta + 2 + \zeta^{-2}$. Its spectrum and pseudospectrum is three-fold rotationally symmetric with respect to its center at $\zeta = 2$; see Fig. 8 in [9] for a plot. This symmetry increases the chance of breakdown in the Lanczos process, and we noticed in fact many breakdowns in examples of this type. (Breakdowns also depend on the initial vectors z_0 and y_0, but since these are chosen real for this real matrix, they are not really in general position with respect to the eigenvectors.) In this particular example, BICGSTAB broke down in step 26.

Our results are shown in Fig. 2. The convergence is slower and rougher than in Example 1, but the relative performance of the various methods remains the same.

Example 3. While both previous examples are real matrices, we consider now the complex Toeplitz matrix

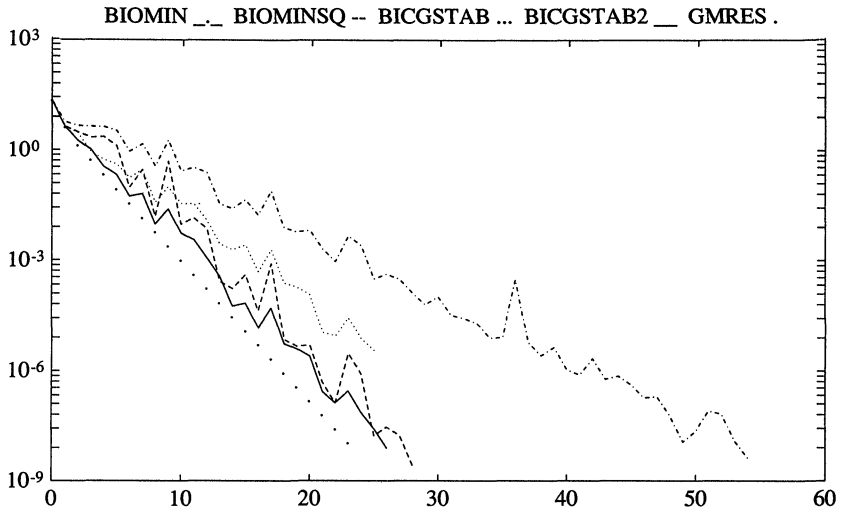

FIG. 2. *The residual norm history for a linear system with the real Toeplitz matrix* (45) *of order* 200. *The selection of methods and the labels of the axes are the same as in Fig.* 1.

$$A := \begin{bmatrix} 4 & 0 & 1 & .7 & & \\ 2i & 4 & 0 & 1 & .7 & \\ & 2i & 4 & 0 & 1 & \ddots \\ & & 2i & 4 & 0 & \ddots \\ & & & 2i & 4 & \ddots \\ & & & & \ddots & \ddots \end{bmatrix} \tag{46}$$

with the symbol $\phi(\zeta) = .7\zeta^3 + \zeta^2 + 4 + 2i\zeta^{-1}$. Its pseudospectrum plot is interpreted as "Picasso's head of a bull" by Reichel and Trefethen; see their Fig. 7 in [9]. Our results are shown in Fig. 3. Both our complex BICGSTAB and the (also complex) BICGSTAB2 do very well, but the difference in their behavior is now, not unexpectedly, much smaller. Again, one needs to point out that the dots for GMRES(∞) represent the optimal residual norm convergence, but that the computational and, in particular, the memory requirements for this method are considerably higher.

Example 4. We finally consider a "real world" example from the Harwell–Boeing collection of large sparse test matrices, namely OILGEN1, a matrix of order 2205 with 14'133 nonzeros, which comes from an oil reservoir simulation on a $21 \times 21 \times 5$ grid. Figure 4 shows the convergence history for BICGSTAB and BICGSTAB2, both applied *without* preconditioning. BICG and BICGS were also tried, but the former converges again only about half as fast (requiring more than 200 iterations to reduce the relative residual to 10^{-5}) and the latter has so many steep peaks in its residual convergence history that the curve would cover up most of what is now shown in Fig. 4. In this difficult example these methods are no longer able to produce a smooth convergence curve, but they do much better than BICG and BICGS.

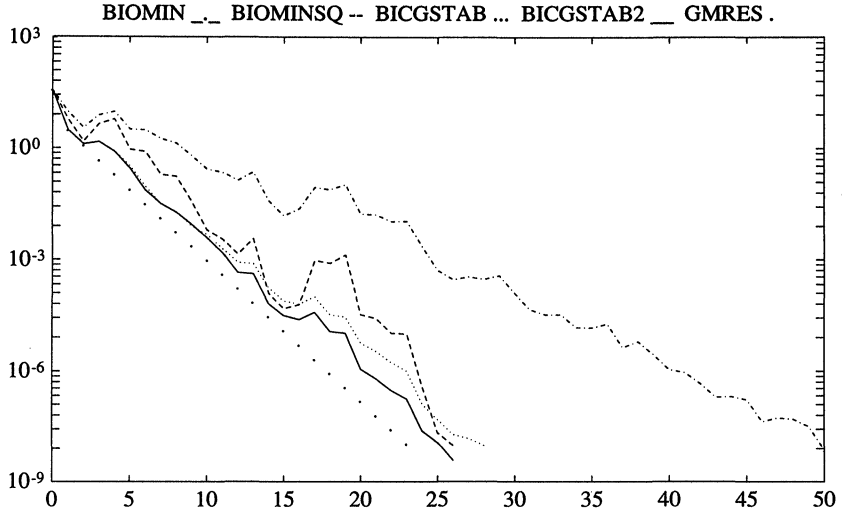

FIG. 3. *The residual norm history for a linear system with the complex Toeplitz matrix* (46) *of order* 200. *The selection of methods and the labels of the axes are the same as in Fig.* 1.

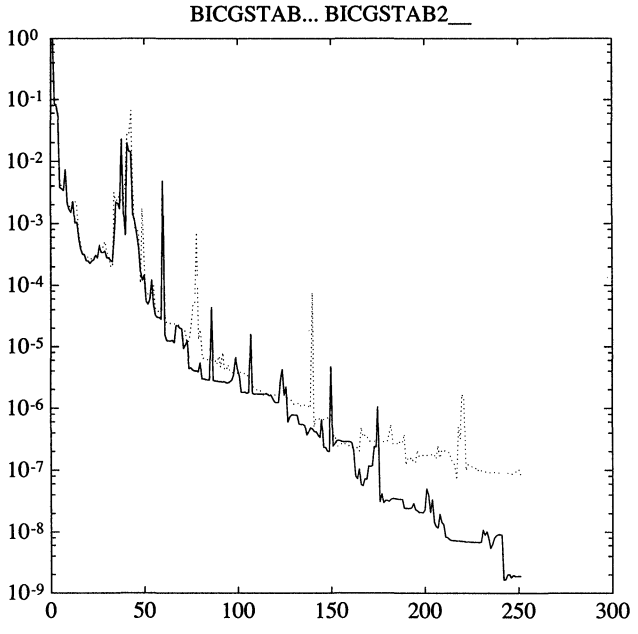

FIG. 4. *The residual norm history (i.e.,* $||r_n||/||r_0||$ *vs.* n*) for a linear system with the* OILGEN1 *matrix of the Harwell–Boeing collection, solved with* BICGSTAB (*dotted*) *and* BICGSTAB2 (*solid*).

Acknowledgment. The author would like to thank Noël Nachtigal for running Example 4 on a test installation of MATLAB 4.

Note added in proof. The algorithm BICGSTAB2 introduced here should not be considered as a black box solver for sparse linear systems. As we have pointed out, there

are situations where BiCGStab2 (as well as BiCG and BiCGStab) can break down or become unstable. A reliable program would have to be able to take appropriate measures in these situations. There are also cases where single steps of BiCGStab should be executed between some of the (double) steps of BiCGStab2; in the present version of the latter, the zeros of the intermediate steps are always dismissed, and this can be a disadvantage.

REFERENCES

[1] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis, G. A. Watson, ed., Vol. 506, Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1976, pp. 73–89.

[2] R. Freund, M. Gutknecht, and N. Nachtigal, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices*, SIAM J. Sci. Comput., 14 (1993), pp. 137–158.

[3] M. H. Gutknecht, *A completed theory of the unsymmetric Lanczos process and related algorithms*, Part I, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 594–639.

[4] ———, *A completed theory of the unsymmetric Lanczos process and related algorithms*, Part II, SIAM J. Matrix Anal. Appl., to appear.

[5] ———, *The unsymmetric Lanczos algorithms and their relations to Padé approximation, continued fractions, and the* qd *algorithm*, Preliminary Proc. Copper Mountain Conference on Iterative Methods (preliminary version), April 2-5, 1990.

[6] I. I. Hirschman, jr., *The spectra of certain Toeplitz matrices*, Illinois J. Math., 11 (1967), pp. 145–159.

[7] C. Lanczos, *Solution of systems of linear equations by minimized iterations*, J. Res. Nat. Bureau Standards, 49 (1952), pp. 33–53.

[8] B. N. Parlett, D. R. Taylor, and Z. A. Liu, *A look-ahead Lanczos algorithm for unsymmetric matrices*, Math. Comp., 44 (1985), pp. 105–124.

[9] L. Reichel and L. N. Trefethen, *Eigenvalues and pseudo-eigenvalues of Toeplitz matrices*, Linear Algebra Appl., 162–164 (1992), pp. 153–185.

[10] Y. Saad and M. H. Schultz, *Conjugate gradient-like algorithms for solving nonsymmetric linear systems*, Math. Comp., 44 (1985), pp. 417–424.

[11] P. Sonneveld, CGS, *a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 36–52.

[12] L. N. Trefethen, *Non-normal matrices and pseudo-eigenvalues*, manuscript.

[13] J. L. Ullman, *Toeplitz matrices associated with a semi-infinite Laurent series*, Bull. Amer. Math. Soc., 73 (1967), pp. 883–885.

[14] H. A. Van der Vorst, Bi-CGSTAB: *A fast and smoothly converging variant of* Bi-CG *for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.