SIAM J. SCI. COMPUT.
Vol. 15, No. 2, pp. 338–347, March 1994

006

A QUASI-MINIMAL RESIDUAL VARIANT OF THE BI-CGSTAB ALGORITHM FOR NONSYMMETRIC SYSTEMS*

T. F. CHAN†, E. GALLOPOULOS‡, V. SIMONCINI§, T. SZETO†, AND C. H. TONG¶

Abstract. Motivated by a recent method of Freund [*SIAM J. Sci. Comput.*, 14 (1993), pp. 470–482], who introduced a quasi-minimal residual (QMR) version of the conjugate gradients squared (CGS) algorithm, a QMR variant of the biconjugate gradient stabilized (Bi-CGSTAB) algorithm of van der Vorst that is called QMRCGSTAB, is proposed for solving nonsymmetric linear systems. The motivation for both QMR variants is to obtain smoother convergence behavior of the underlying method. The authors illustrate this by numerical experiments that also show that for problems on which Bi-CGSTAB performs better than CGS, the same advantage carries over to QMRCGSTAB.

Key words. conjugate gradients, Lanczos algorithm, iterative methods, BCG, CGS, QRM, Bi-CGSTAB, nonsymmetric linear systems

AMS subject classifications. 65F10, 65Y20

1. Introduction.

In this note we propose a variation of the Bi-CGSTAB algorithm of van der Vorst [18] for solving the linear system

$$Ax = b, \tag{1}$$

where A is a nonsymmetric sparse matrix of order n.

Various attempts have been made in the last forty years to extend the highly successful conjugate gradient (CG) algorithm to the nonsymmetric case [4]. One such natural extension is what is currently called the biconjugate gradient algorithm (BCG) [9], [1]. Although BCG is still quite competitive today, it also has several well-known drawbacks. Among these are (i) the need for matrix-vector multiplications with A^T (which can be inconvenient as well as doubling the number of matrix-vector multiplications compared to CG for each increase in the degree of the underlying Krylov subspace), (ii) the possibility of breakdowns, and (iii) erratic convergence behavior.

Many recently proposed methods can be viewed as improvements over some of these drawbacks of BCG. The most notable of these is the ingenious CGS method proposed by Sonneveld [14], which cures the first drawback mentioned above by computing the square of the BCG polynomial without requiring A^T. Hence when BCG converges, CGS is an attractive, faster converging alternative. However, this relation between the residual polynomials also causes CGS to behave even more erratically than BCG, particularly in near-breakdown situations for BCG [8], [18]. These observations led van der Vorst [18] to introduce Bi-CGSTAB, a more smoothly converging variant of CGS. The main idea is to form a product of the BCG

*Received by the editors May 28, 1992; accepted for publication (in revised form) March 22, 1993.

†Department of Mathematics, University of California at Los Angeles, California 90024 (chan@math.ucla.edu, szeto@math.ucla.edu). The authors' research was partially supported by Department of Energy grant DE-FG03-87ER25037, Office of Naval Research grant N00014-90-J-1695, National Science Foundation grants ASC90-03002 and ASC92-01266, and Army Research Office grant DAAL03-91-G-150.

‡Department of Computer Science and Center for Supercomputing Research and Development, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801 (stratis@csrd.uiuc.edu). This author's research was supported by Department of Energy grant DOE DE-FG02-85ER25001 and National Science Foundation grants NSF CCR-9120105 and CCR-9024554. Additional support was provided by the State of Illinois Department of Commerce and Community Affairs, State Technology Challenge Fund grant SCCA 91-108.

§Center for Supercomputing Research and Development, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801 (simoncin@csrd.uiuc.edu). This author's research was supported by fellowship 2043597 from the Consiglio Nazionale delle Ricerche, Italy.

¶Center for Computer Engineering, Sandia National Laboratories, Livermore, California 94551 (chtong@snll-arpagw.llnl.gov).

polynomial with another, locally defined polynomial. The Bi-CGSTAB method was further refined by Gutknecht [7] to handle complex matrices and also lead to better convergence for the case of complex eigenvalues. Nevertheless, although the Bi-CGSTAB algorithms were found to perform very well compared to CGS in many situations, there are cases where convergence is still quite erratic (see, for example, §4 and [12]).

In a recent paper [3], Freund proposed a new version of CGS, called TFQMR, which "quasi-minimizes" [6] the residual in the space spanned by the vectors generated by the CGS iteration. Numerical experiments show that in most cases TFQMR retains the good convergence features of CGS while correcting its erratic behavior. The transpose-free nature of TFQMR, its low computational cost, and its smooth convergence behavior make it an attractive alternative to CGS. On the other hand, since the square of the residual polynomial for BCG is still in the space being quasi-minimized, in many practical examples CGS and TFQMR converge in about the same number of steps. We note however that in contrast to CGS, the asymptotic behavior of TFQMR has been analyzed [2]. It is also well known that the CGS residual polynomial can be quite polluted by round-off error [16]. One possible remedy would be to combine TFQMR with a look-ahead Lanczos technique as was done for the original QMR method [5]. In this paper, we take an alternative approach by deriving quasi-minimum residual extensions to Bi-CGSTAB. We call the basic method QMRCGSTAB and illustrate its smoothed convergence by means of numerical experiments.

It may appear redundant to combine the local minimization in Bi-CGSTAB with a global quasi-minimization. However, our view is that the local minimization is secondary in nature and is only used as a way of generating residual polynomials in the appropriate Krylov subspace over which the residual is being quasi-minimized. In fact, this view allows us some flexibility in modifying the local minimization step in Bi-CGSTAB, which leads to other quasi-minimal residual variants. Although we use extensively notation introduced in [18] for algorithm Bi-CGSTAB, for the sake of brevity we refer to that paper for a description of the method.

2. The QMRCGSTAB algorithm. The algorithm proposed in this paper is inspired by TFQMR in that it applies the quasi-minimization principle to the Bi-CGSTAB method, in the same way that TFQMR is derived from CGS. During each step of Bi-CGSTAB, the following vector relations hold:

$$s_i = r_{i-1} - \alpha_i A p_i, \qquad r_i = s_i - \omega_i A s_i, \tag{2}$$

where α_i is the same as the analogous coefficient in BCG, and ω_i is chosen by a local steepest descent principle. Note that x_i is completely determined by α_i and ω_i. Instead, our algorithm uses Bi-CGSTAB to generate the vectors p_i and s_i, but chooses x_i by quasi-minimizing the residual over their span. Let $Y_k = \{y_1, y_2, \ldots, y_k\}$, where $y_{2l-1} = p_l$ for $l = 1, \ldots, [(k+1)/2]$ and $y_{2l} = s_l$ for $l = 1, \ldots, [k/2]$ ($[k/2]$ is the integer part of $k/2$). In the same way, let $W_{k+1} = \{w_0, w_1, \ldots, w_k\}$ with $w_{2l} = r_l$ for $l = 0, \ldots, [k/2]$ and $w_{2l-1} = s_l$ for $l = 1, \ldots, [(k+1)/2]$. We also define $\{\delta_1, \delta_2, \ldots, \delta_k\}$, as $\delta_{2l} = \omega_l$ for $l = 1, \ldots, [(k+1)/2]$ and $\delta_{2l-1} = \alpha_l$ for $l = 1, \ldots, [(k+1)/2]$. In this case, for each column of W_{k+1} and Y_k, (2) may be written as

$$A y_j = (w_{j-1} - w_j)\delta_j^{-1}, \qquad j = 1, \ldots, k, \tag{3}$$

or, using matrix notation,

$$A Y_k = W_{k+1} E_{k+1},$$

where E_{k+1} is a $(k+1) \times k$ bidiagonal matrix with diagonal elements δ_j^{-1} and lower diagonal elements $-\delta_j^{-1}$.

It can easily be checked that the degree of the polynomials corresponding to the vectors r_i, s_i, and p_i are $2i$, $2i-1$, and $2i-2$, respectively. Therefore, $\text{span}(Y_k) = \text{span}(W_k) = K_{k-1}$, where K_k is the Krylov subspace of degree k generated by r_0. The main idea in QMRCGSTAB is to look for an approximation to the solution of (1), using the Krylov subspace K_{k-1}, in the form

$$x_k = x_0 + Y_k g_k \quad \text{with } g_k \in \mathbb{R}^n.$$

Hence, we may write the residual $r_k = b - Ax_k$ as

$$r_k = r_0 - AY_k g_k = r_0 - W_{k+1} E_{k+1} g_k.$$

Using the fact that the first vector of W_{k+1} is indeed r_0, it follows that

$$r_k = W_{k+1}(e_1 - E_{k+1} g_k),$$

where e_1 is the first vector of the canonical basis. Since the columns of W_{k+1} are not normalized, it was suggested in [3] to use a $(k+1) \times (k+1)$ scaling matrix Σ_{k+1}=diag$(\sigma_1, \ldots, \sigma_{k+1})$, with $\sigma_i = \|w_i\|$, to make the columns of W_{k+1} to be of unit norm. Then

$$r_k = W_{k+1}\Sigma_{k+1}^{-1}\Sigma_{k+1}(e_1 - E_{k+1}g_k) = W_{k+1}\Sigma_{k+1}^{-1}(\sigma_1 e_1 - H_{k+1}g_k) \tag{4}$$

with $H_{k+1} = \Sigma_{k+1} E_{k+1}$.

The quasi-minimal residual approach consists of the minimization of $\|\sigma_1 e_1 - H_{k+1} g\|$ for some $g \in \mathbb{R}^k$. In §3 we introduce a variant of QMRCGSTAB that generates W_{k+1} with pairwise orthogonal columns.

The least squares minimization of $\|\sigma_1 e_1 - H_{k+1} g\|$ is solved using QR decomposition of H_{k+1}. This is done in an incremental manner by means of Givens rotations. Since H_{k+1} is lower bidiagonal, only the rotation of the previous step is needed. We refer to [3] for a detailed description of the QR decomposition procedure.

The pseudocode for the QMRCGSTAB algorithm is as follows, in which the Givens rotations used in the QR decomposition are written out explicitly:

ALGORITHM QMRCGSTAB(A, b, x_0, ϵ)

(1) Initialization

$r_0 = b - Ax_0$

choose $\tilde{r}_0$ such that $(\tilde{r}_0, r_0) \neq 0$

$p_0 = v_0 = d_0 = 0$

$\rho_0 = \alpha_0 = \omega_0 = 1$; $\tau = \|r_0\|$, $\theta_0 = 0$, $\eta_0 = 0$

(2) for $k = 1, 2, \ldots$ do

$\rho_k = (\tilde{r}_0, r_{k-1})$; $\beta_k = (\rho_k \alpha_{k-1})/(\rho_{k-1}\omega_{k-1})$

$p_k = r_{k-1} + \beta_k(p_{k-1} - \omega_{k-1} v_{k-1})$

$v_k = Ap_k$

$\alpha_k = \rho_k/(\tilde{r}_0, v_k)$

$s_k = r_{k-1} - \alpha_k v_k$

(2.1) First quasi-minimization and update iterate

$\tilde{\theta}_k = \|s_k\|/\tau$; $c = 1/\sqrt{1+\tilde{\theta}_k^2}$; $\tilde{\tau} = \tau\tilde{\theta}_k c$

$\tilde{\eta}_k = c^2 \alpha_k$

$\tilde{d}_k = p_k + \frac{\theta_{k-1}^2 \eta_{k-1}}{\alpha_k} d_{k-1}$

$\tilde{x}_k = x_{k-1} + \tilde{\eta}_k \tilde{d}_k$

(2.2) compute t_k, ω_k and update r_k

$t_k = As_k$

$\omega_k = (s_k, t_k)/(t_k, t_k)$

$r_k = s_k - \omega_k t_k$

(2.3) Second quasi-minimization and update iterate

$\theta_k = \|r_k\|/\tilde{\tau}$; $c = 1/\sqrt{1+\theta_k^2}$; $\tau = \tilde{\tau}\theta_k c$

$\eta_k = c^2\omega_k$

$d_k = s_k + \frac{\tilde{\theta}_k^2\tilde{\eta}_k}{\omega_k}\tilde{d}_k$

$x_k = \tilde{x}_k + \eta_k d_k$

If x_k is accurate enough, then quit

(3) end

To check the convergence, the estimate $\|\hat{r}_k\| \leq \sqrt{k+1}|\tau|$ was used, where $\hat{r}_k$ denotes the QMRCGSTAB residual at step k [3].

Note that the cost per iteration is slightly higher than for Bi-CGSTAB, since two additional inner products are needed to compute the elements of Σ_{k+1}. A more detailed discussion on computational costs is given in §4.

3. Some variants of QMRCGSTAB. The use of quasi-minimization in the "product algorithms" (such as CGS and Bi-CGSTAB) introduces some flexibility. For example, the underlying product algorithm need not be constrained to generate a residual polynomial that has small norm since, presumably, the quasi-minimization step will handle that. Instead, the basic iteration can be viewed as only generating a set of vectors spanning the Krylov subspace over which the quasi-minimization is applied. This leads us to several variants of QMRCGSTAB, which we will briefly describe. Note however that only one of these variants will be used in the numerical experiments.

We make two observations on the QMRCGSTAB method:

1. It is not crucial that the steepest descent step reduces the norm of the residual as long as it increases the degree of the Krylov subspace associated with W_{k+1}.
2. If W_{k+1} were orthogonal, then quasi-minimization becomes true minimization of the residual.

Therefore, it is natural to choose ω_i to make W_{k+1} "more orthogonal." For example, one can choose ω_i to make r_i orthogonal to s_i and W_k pairwise orthogonal. This leads to the formula:

$$\omega_i = \frac{(s_i, s_i)}{(s_i, t_i)},$$

which replaces the corresponding formula in Algorithm QMRCGSTAB. We call this variant QMRCGSTAB2. We note that since the inner-product (s_i, s_i) is already needed to compute $\tilde{\theta}_i$, we save one inner product compared to QMRCGSTAB.

We also note that similarly to Bi-CGSTAB, both QMRCGSTAB and QMRCGSTAB2 break down if $(s_i, t_i) = 0$, which is possible if A is indefinite (in fact it is always true if A is skew symmetric). This is an additional breakdown condition over that of BCG. One possible strategy to overcome this is to set a lower bound for the quantity $|(s_i, t_i)|$. However, for matrices with large imaginary parts, Gutknecht [7] observed that Bi-CGSTAB does not perform well because the steepest descent polynomials have only real roots and thus cannot be expected to approximate the spectrum well. In principle, it is possible to derive a quasi-minimal residual version of Gutknecht's variant of Bi-CGSTAB, but we shall not pursue that here.

4. Numerical experiments. We next compare the performance of the QMRCGSTAB variants with that of Bi-CGSTAB, TFQMR, and CGS.

Table 1 shows the cost per step of the methods under discussion, excluding the cost for computing the residual norm which is the same for all methods.

TABLE 1
Cost per step for each method.

	Inner products	DAXPY operations	Matrix-vector multiplications
Bi-CGSTAB	4	6	2
CGS	2	7**	2
QMRCGSTAB	6	8	2
QMRCGSTAB2	5	8	2
TFQMR	4	10	2

In the sequel we present experiments to show that QMRCGSTAB indeed achieves a smoothing of the residual compared to Bi-CGSTAB. Note however that, because the Bi-CGSTAB method already improves the erratic residual convergence of BCG, the effect of QMRCGSTAB is not as impressive as the one of TFQMR on the residual of CGS.

Unless stated otherwise, in all examples, the right-hand side b was generated as a random vector with values distributed uniformly in (0, 1), and the starting vector x_0 was taken to be zero. All matrices arising from a partial differential operator were obtained using centered, second-order finite differences. The methods were compared on the basis of the number of iterations necessary to achieve relative residual $\|r_k\|/\|r_0\| < 10^{-8}$ with $r_k = b - Ax_k$ being the true residual. Hence, the figures were built with the abscissae representing the number of iterations and the ordinates representing $\|r_k\|/\|r_0\|$ graded with a logarithmic scale. Experiments were conducted using a Beta test version of Matlab 4.0 [10] running on a Sun Sparc workstation.

Example 1. This example was taken from [14] and corresponds to the discretization of the convection-diffusion operator

$$L(u) = -\varepsilon \Delta u + \cos(\alpha) u_x + \sin(\alpha) u_y \tag{5}$$

on the unit square with homogeneous Dirichlet conditions on the boundary and parameters $\varepsilon = 0.1$ and $\alpha = -30°$, using 40 grid points per direction, yielding a matrix of order $n = 1600$. Figure 1 shows the convergence histories, from which we can see the smoothing effect of quasi-minimization on the CGS and Bi-CGSTAB residuals. We see that Bi-CGSTAB and its smoothed counterparts converge slightly faster than CGS and TFQMR, with QMRCGSTAB2 showing the best performance by a small margin.

Example 2. This example was taken from [17] and corresponds to the discretization of

$$-(Du_x)_x - (Du_y)_y = 1 \tag{6}$$

on the unit square with homogeneous boundary conditions. We used a coarser grid than the one considered in [17]; that is, 50 grid points per direction yielding a matrix of order $n = 2500$. Parameter D takes the value $D = 10^5$ in $0 \le x, y \le 0.75$, $D = 0.1$ in $0.75 < x, y \le 1$, and $D = 1$ everywhere else. Left diagonal preconditioning was applied. In [17], this matrix was used to illustrate the better convergence of Bi-CGSTAB over CGS. We see from Fig. 2 that this advantage carries over to the smoothed versions. Furthermore, even though the matrix is symmetric positive definite and hence CG is applicable, as shown in Fig. 2, the method

**Strictly speaking, one of the operations is a simple vector addition. This must be taken into account if floating point operations were to be counted.

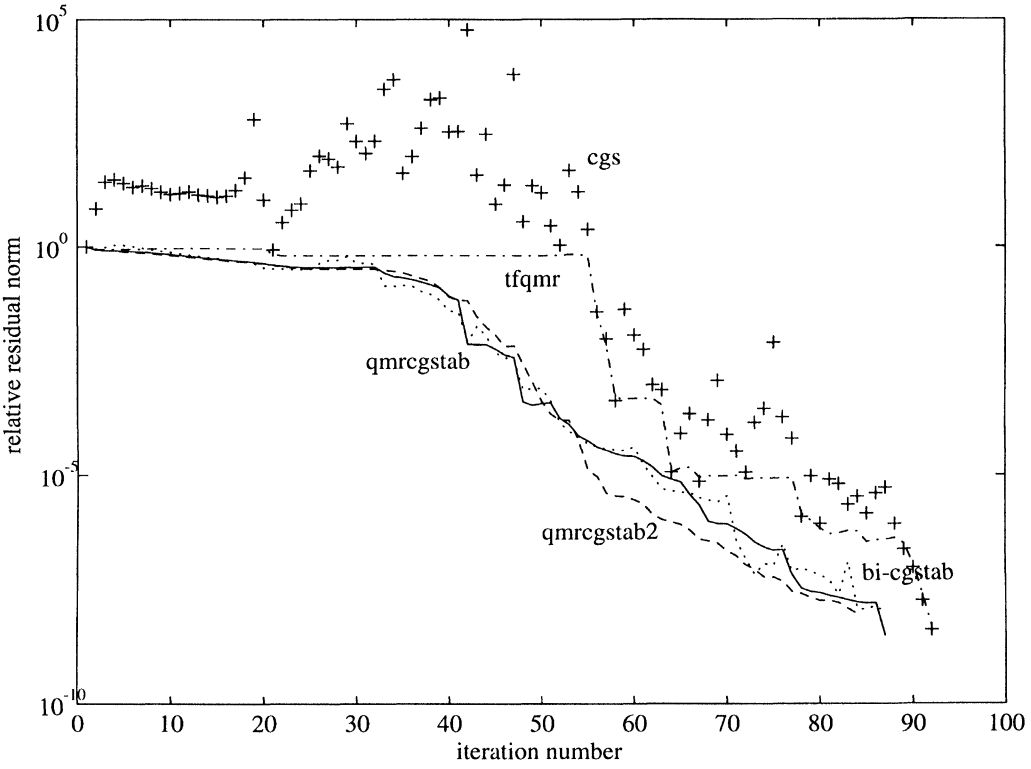

FIG. 1. *Example* 1: *Two-dimensional convection-diffusion operator* (5).

stagnates. This is due to the fact that for this operator, the computed direction vectors of CG methods rapidly lose orthogonality [16]. We note that to make cost comparisons meaningful, the CG curve was plotted so that each "iteration" corresponds to two true CG iterations, i.e., two matrix-vector multiplications.

Example 3. This example comes from the discretization of the convection-diffusion equation

$$L(u) = -\Delta u + \gamma(xu_x + yu_y) + \beta u \tag{7}$$

on the unit square where $\gamma = 100$, $\beta = -100$, for a 63×63 grid, yielding a matrix of order $n = 3969$. No preconditioning was used. In this example, we see the CGS-based methods converge a little faster than Bi-CGSTAB and QMRCGSTAB, but the pairwise orthogonal variant, QMRCGSTAB2, is the fastest. See Fig. 3.

Example 4. Figure 4 shows the results of a three-dimensional version of Example 3 without preconditioning:

$$L(u) = -\Delta u + \gamma(xu_x + yu_y + zu_z) + \beta u \tag{8}$$

on the unit cube where $\beta = -100$, and $\gamma = 50$ for a $15 \times 15 \times 15$ grid, yielding a matrix of order $n = 3375$.

We note that in this example the improvement caused by Bi-CGSTAB over CGS and TFQMR is impressive. Therefore it is not surprising that there is only little additional improvement brought by the variants proposed in this paper. We note that for this operator, the

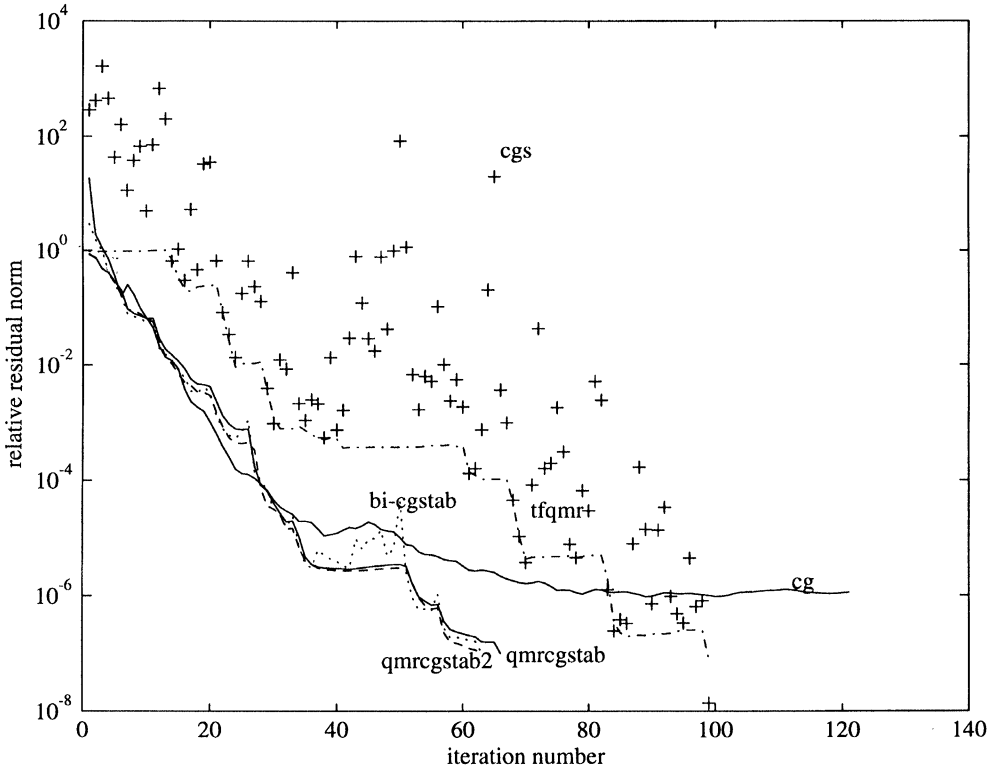

FIG. 2. *Example* 2: *Two-dimensional operator with discontinuous coefficients. Every point on the* CG *curve refers to two CG iterations.*

use of centered differences and large values of γ are unfavorable for Bi-CGSTAB-type methods, since the resulting matrices would have pronounced skew-symmetric components and eigenvalues with large imaginary parts [7]; different discretization methods would be more attractive [13].

Example 5. The next example illustrates how all methods can be affected by the conditioning of the generated polynomial. Matrix A is a modification of an example presented in [11],

$$A = I_{n/2} \otimes \begin{pmatrix} \epsilon & 1 \\ -25 & 100 \end{pmatrix}, \tag{9}$$

i.e., A is an $n \times n$ block diagonal matrix with 2×2 blocks and $n = 40$. We chose $b = (1\ 0\ 1\ 0\ \cdots)^T$ and $\tilde{r}_0 = r_0$. For such a b the norm of the resulting BCG polynomial satisfies $\|\varphi_n\| = \mathcal{O}(\epsilon^{-1})$. Thus, $\|\varphi_n^2\| = \mathcal{O}(\epsilon^{-2})$ in the squared methods and we can foresee numerical problems when ϵ is small.

Each entry of Table 2 shows (i) the number of correct digits, d, in the relative residual obtained after running each algorithm until the relative residual dropped below 10^{-8} but without exceeding 20 matrix vector multiplications, and (ii) in parentheses, the number of matrix vector multiplications, *mv*, that is a number, not greater than 20, needed to achieve a relative residual of 10^{-d}.

In exact arithmetic, finite termination occurs after the second BCG polynomial φ_2 is computed in both the CGS and Bi-CGSTAB algorithms. We see from Table 2 that all methods

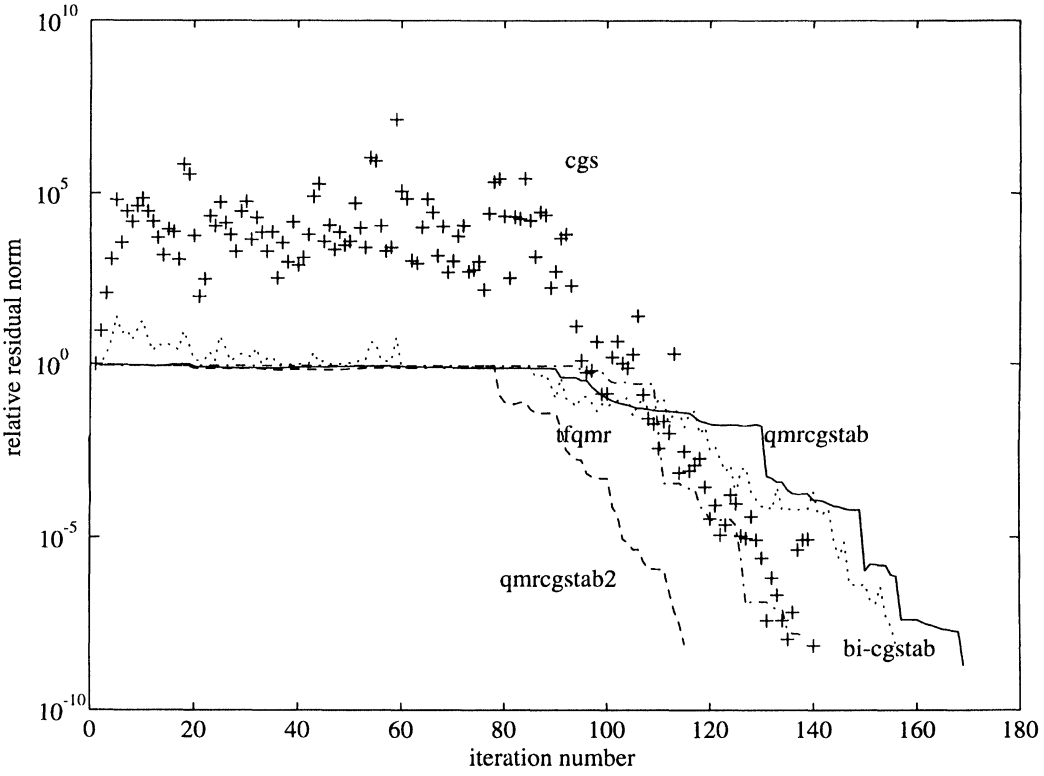

FIG. 3. *Example* 3: *Two-dimensional convection-diffusion operator* (7).

TABLE 2

Example 5: *Correct digits and matrix vector multiplications at termination*: *d(mv). A maximum* 20-*matrix vector multiplications allowed.*

	ϵ			
Method	1.0	10^{-4}	10^{-8}	10^{-12}
CGS	14(4)	5(4)‡	−3(20)*	−1(4)†
TFQMR	13(3)	5(4)‡	1(20)‡	1(20)‡
Bi-CGSTAB	16(3)	12(3)	7(3)‡	3(3)‡
QMRCGSTAB	16(3)	12(3)	7(3)‡	3(3)‡
QMRCGSTAB2	16(3)	12(3)	7(3)‡	3(3)‡

* Oscillatory behavior observed.
‡ Residual stagnates before maximum number of *mv*'s was reached.
† Iterations stopped when division by zero was encountered.

behave equally well for $\epsilon = 1.0$. As ϵ decreases, round-off error causes CGS and TFQMR, which are based on squaring, to fail or not to converge within the expected time. Furthermore, both CGS and TFQMR lose about twice as many digits as Bi-CGSTAB and its quasi-minimal variants. We also mark the instances of the quasi-minimal variants whose residuals stagnate before the maximum number of iterations has been reached. We note that although the example is contrived, it does justify the implementation of a QMRCGSTAB-type method.

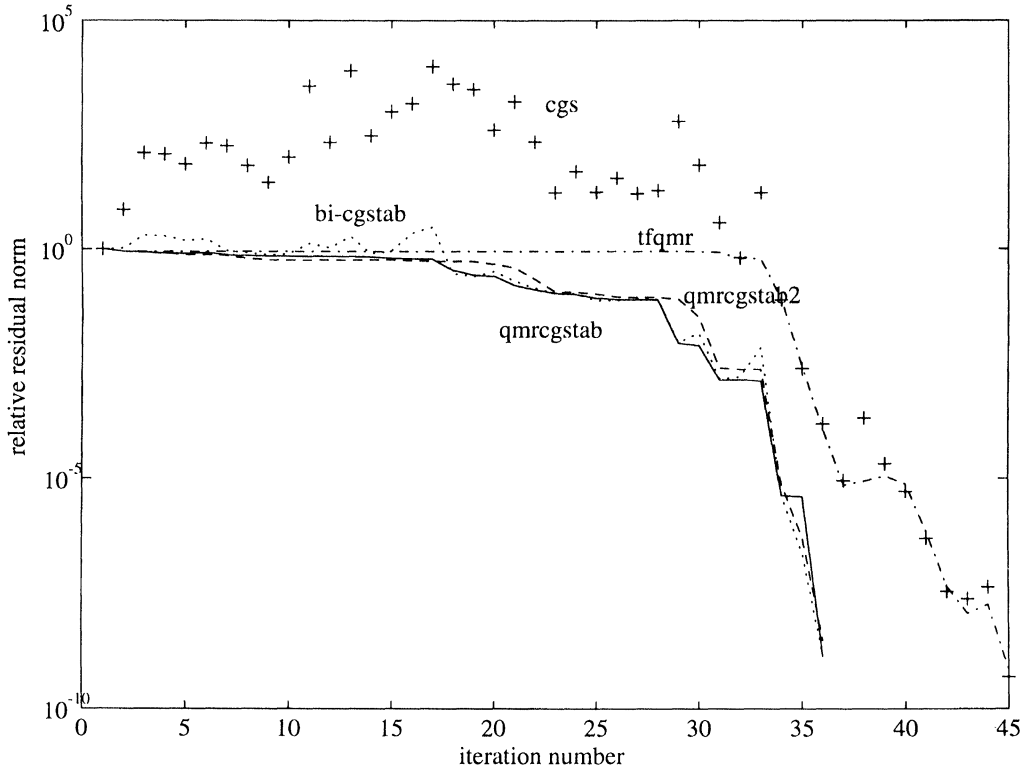

FIG. 4. *Example* 4: *Three-dimensional convection-diffusion operator* (8).

We finally observe that experiments using several of the methods discussed herein, albeit using another naming convention, were presented in [15].

5. Conclusions and future work. We have derived two QMR variants of Bi-CGSTAB. Our motivation for these methods was to inherit any potential improvements on performance that Bi-CGSTAB offers over CGS, while at the same time to provide a smoother convergence behavior. We have shown numerically that this is indeed true for many realistic problems. Although in their present form, the two proposed methods still suffer from some numerical problems, they have many desirable properties: they are transpose-free, they use short recurrences, they make efficient use of matrix-vector multiplications, and they demonstrate smooth convergence behavior.

Acknowledgments. We are grateful to J. M. Hammond, C. Moler, and The Mathworks, Inc. for providing us with the Beta Test version of Matlab 4.0. [10]. We also thank R. Freund and the referees for providing us with very useful suggestions and corrections.

REFERENCES

[1] R. FLETCHER, *Conjugate gradient methods for indefinite linear systems*, in Proc. Dundee Biennial Conf. Numer. Anal., Scotland, G. A. Watson, ed., Lecture Notes in Math., Vol. 506, Springer-Verlag, Berlin, 1976, pp. 73–89.

[2] R. W. FREUND, *Quasi-kernel polynomials and convergence results for quasi-minimal residual iterations*, in Numerical Methods in Approximation Theory, Vol. 9, D. Braess and L. L. Schumaker, eds., Birkhäuser Verlag, Basel, 1992, pp. 77–95.

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

[4] R. W. FREUND, G. H. GOLUB, AND N. M. NACHTIGAL, *Iterative solution of linear systems*, Acta Numerica, 1 (1992), pp. 57–100.

[5] R. W. FREUND, M. H. GUTKNECHT, AND N. M. NACHTIGAL, *An Implementation of the Look-Ahead Lanczos Algorithm for Non-Hermitian Matrices*: *Part* I, Tech. Report 90.45, Research Institute for Advanced Computer Science, Nov. 1990.

[6] 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.

[7] M. H. GUTKNECHT, *Variants of BiCGStab for Matrices with Complex Spectrum*, Tech. Report, Eidgenössische Technische Hochschule, Zürich, Aug. 1991. IPS Research Report 91-14.

[8] W. D. JOUBERT AND T. A. MANTEUFFEL, *Iterative methods for nonsymmetric linear systems*, in Iterative Methods for Large Linear Systems, D. R. Kincaid and L. J. Hayes, eds., Academic Press, Boston, 1990, pp. 149–171.

[9] C. LANCZOS, *Solution of linear equations by minimized iterations*, J. Res. Natl. Bur. Stand., 49 (1952), pp. 33–53.

[10] THE MATHWORKS, INC., *MATLAB User's Guide*, beta 3 ed., Natick, MA, Feb. 1991.

[11] N. M. NACHTIGAL, S. C. REDDY, AND L. N. TREFETHEN, *How fast are nonsymmetric matrix iterations?*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 778–795.

[12] C. POMMERELL AND W. FICHTNER, *PILS*: *An iterative linear solver package for ill-conditioned systems*, in Proc. Supercomputing'91, IEEE, Albuquerque, NM, Nov. 1991, pp. 588–599.

[13] A. SEGAL, *Aspects of numerical methods for elliptic singular perturbation problems*, SIAM J. Sci. Statist. Comput., 3 (1982), pp. 327–349.

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

[15] C. H. TONG, *A comparative study of preconditioned Lanczos methods for nonsymmetric linear systems*, Tech. Report SAND91-8240 UC404, Sandia National Laboratories, Albuquerque, NM, Sept. 1992.

[16] H. A. VAN DER VORST, *The convergence behavior of preconditioned CG and CG-S in the presence of rounding errors*, in Preconditioned Conjugate Gradient Methods, O. Axelsson and L. Y. Kolotilina, eds., Lecture Notes in Math., Vol. 1457, Springer-Verlag, Berlin, 1990, pp. 126–136.

[17] ———, *Bi-CGSTAB*: *A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems*, Tech. Report 633, Dept. of Math., University of Utrecht, the Netherlands, Dec. 1990.

[18] ———, *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.