ELSEVIER

Applied Numerical Mathematics 19 (1995) 235-254

APPLIED NUMERICAL MATHEMATICS

An overview of approaches for the stable computation of hybrid BiCG methods

Gerard L.G. Sleijpen *, Henk A. van der Vorst [1]

Mathematical Institute, University of Utrecht, P.O. Box 80.010, 3508 TA Utrecht, Netherlands

Abstract

It is well known that BiCG can be adapted so that the operations with A^T can be avoided, and hybrid methods with computational complexity almost similar to BiCG can be constructed in a further attempt to improve the convergence behavior. Examples of this are CGS, Bi-CGSTAB, and BiCGstab(l).

In many applications, the speed of convergence of these methods is very dependent on the incorporated BiCG process. The accuracy of the iteration coefficients of BiCG depends on the particular choice of the hybrid method. We will discuss the accuracy of these coefficients and how this affects the speed of convergence. We will show that hybrid methods exist which have better accuracy properties. This may lead to faster convergence and more accurate approximations.

We also discuss look-ahead strategies for the determination of appropriate values for l in BiCGstab(l). These strategies are easily applied for the hybrid part, in contrast to similar techniques for the BiCG part (but of course they do not solve the breakdown problems of the BiCG part).

0. Introduction

By combining BiCG with other Krylov subspace methods, operations with the transpose of the matrix, as in standard BiCG [7], can be avoided. Examples of such *hybrid BiCG methods* are, CGS [23] where BiCG is combined with BiCG itself, Bi-CGSTAB [25] as a combination of BiCG and GMRES(1), BiCGStab2 [12] that incorporates GCR(2) (i.e. GCR restarted every 2nd step), and BiCGstab(l) [19,22] that combines GMRES(l) with each lth step of BiCG.

The method on top of BiCG in hybrid BiCG is used to get an additional reduction of the residual, but it can also be designed for other desirable properties. In [8], for instance, the performance of hybrid BiCG methods as linear solvers in (inexact) Newton schemes for nonlinear problems is

* Corresponding author. E-mail: sleijpen@math.ruu.nl.

[1] E-mail: vorst@math.ruu.nl.

SSDI 0168-9274(95)00085-2

studied. It is shown that the hybrid part can be selected to reduce the number of Newton steps rather than for the reduction of the computational costs for the linear Jacobian systems.

It is not our purpose to compare hybrid BiCG methods with BiCG or QMR [10]. This has been done in, e.g., [16,24]. In the present paper, we will concentrate on approaches that help to reduce the effects of local rounding errors on the BiCG iteration coefficients in hybrid schemes.

Locally accurate BiCG coefficients, i.e., coefficients that ensure at least local bi-orthogonality of the BiCG basis vectors, are important for maintaining the convergence of the incorporated BiCG process in finite precision arithmetic. When the convergence of the hybrid method exhibits a phase of stagnation or a phase of very poor reductions, one often has to rely on the BiCG part in the hope to arrive at the phase where further reduction takes place.

We will see that the hybrid part affects the accuracy of the BiCG part (see Section 4). Our analysis will suggest strategies for improvement. To be more specific, in Section 6 we will explain why BiCGstab(l) often performs much better for $l > 1$ than Bi-CGSTAB (=BiCGstab(1)). GMRES [18] and GCR [6] produce residuals that are minimal in the associated Krylov subspaces, while methods as FOM [17] or GENCG [6] produce residuals that are orthogonal to Krylov subspaces of lower order, and this has been exploited to decrease the effects of rounding errors to the BiCG iteration coefficients. We will call this a stabilizing effect. One can form convex combinations of these methods: the residual at step l of such a combined method is a convex mean of the lth residual of GMRES and the one of FOM. An appropriate convex combination will turn out to lead to a very attractive stabilizing hybrid part for hybrid BiCG (see Sections 5 and 7). Due to more accurate BiCG coefficients, the stabilized BiCGstab(l) method often converges (much) faster than BiCGstab(l), and may lead to convergence in cases where BiCGstab(l) does not converge (see Section 9 on numerical experiments). The stabilizing strategy can be implemented in BiCGstab(l) without significant computational overhead (the additional operations concern vectors of dimension $\leqslant l+1$).

Also for the stabilized BiCGstab(l) the question arises what the most suitable value for l might be. Larger l, say $l = 2$, 4, or 8, may lead to more accurate BiCG coefficients and to faster convergence. On the other hand, a larger l is slightly more expensive per matrix–vector multiplication and this is not *always* compensated by faster convergence. Moreover, the final residual may not quite agree with the final approximation (see [22], also for implementational approaches to minimize this effect). Based on our analysis of the accuracy of BiCG coefficients, we will suggest strategies in Section 8 to determine l automatically. We allow l to have a different value in successive iteration phases. Numerical experiments in Section 9 illustrate how well our strategies may do.

BiCG, CGS, and Bi-CGSTAB play a key role in this paper. We briefly discuss these methods in Sections 1–3.

Our approaches for the improvement of the convergence of BiCG in finite precision arithmetic work *indirectly* on BiCG: we will stabilize the computation through adapting the hybrid part in hybrid BiCG. In other publications (e.g., [1,2,5,10,15]) strategies are discussed that operate *directly* on BiCG. Composite (multiple) BiCG steps are formed in the iterative process whenever standard single steps are expected to introduce large rounding errors (near-breakdown). Here, we will not consider these so-called look-ahead strategies, but we note that these strategies may be combined with our approaches in order to further improve the BiCG part. Our approaches, if applied only, may not cure the negative effects of (near-) breakdown.

Although methods like GMRES and GCR are mathematically equivalent (if GCR does not break

down), GMRES is known to be more stable than GCR. Similarly, as argued in [22], the stability of a hybrid BiCG method also depends on its implementation. For these aspects, we refer to [22]; they will not be discussed in the present paper.

Parts of our presentation in the Sections 4–7 are taken from [20].

1. BiCG

With an initial guess x_0, for the solution x of the equation $Ax = b$, and some "shadow" residual $\tilde{r}_0$, BiCG [7] produces iteratively sequences of approximations x_k, residuals r_k, and search directions u_k by

$$u_k = r_k - \beta_k u_{k-1}, \qquad x_{k+1} = x_k + \alpha_k u_k, \qquad r_{k+1} = r_k - \alpha_k A u_k, \tag{1}$$

where the BiCG coefficients α_k and β_k are such that r_k and Au_k are orthogonal with respect to the shadow Krylov subspace $\mathcal{K}_k(A^{\mathrm{T}}; \tilde{r}_0) := \mathrm{span}(\tilde{r}_0, A^{\mathrm{T}}\tilde{r}_0, \ldots, (A^{\mathrm{T}})^{k-1}\tilde{r}_0)$. If (ψ_k) is some sequence of polynomials of degree k with a nonvanishing leading coefficient θ_k then the vectors $\psi_0(A^{\mathrm{T}})\tilde{r}_0, \ldots, \psi_{k-1}(A^{\mathrm{T}})\tilde{r}_0$ form a basis of $\mathcal{K}_k(A^{\mathrm{T}}; \tilde{r}_0)$ and we have (see [23] or [19]):

$$\beta_k = \frac{\theta_{k-1}}{\theta_k}\frac{\rho_k}{\sigma_{k-1}} \quad \text{and} \quad \alpha_k = \frac{\rho_k}{\sigma_k} \quad \text{where} \quad \begin{cases} \rho_k := (r_k, \psi_k(A^{\mathrm{T}})\tilde{r}_0), \\ \sigma_k := (Au_k, \psi_k(A^{\mathrm{T}})\tilde{r}_0). \end{cases} \tag{2}$$

2. CGS

Sonneveld [23] suggested to rewrite the inner products so as to avoid operations with A^{T}, e.g.,

$$\rho_k = (r_k, \psi_k(A^{\mathrm{T}})\tilde{r}_0) = (\psi_k(A)r_k, \tilde{r}_0) = (\boldsymbol{r}_k, \tilde{r}_0), \tag{3}$$

and to generate recursions for the vectors

$$\boldsymbol{r}_k = \psi_k(A)r_k, \tag{4}$$

hoping that the operator $\psi_k(A)$ would lead to a further reduction of the BiCG residual. More specifically he suggested to take $\psi_k = \phi_k$, with ϕ_k such that for the BiCG residual r_k we have $r_k := \phi_k(A)r_0$, which led to the CGS method: $\boldsymbol{r}_k = \phi_k^2(A)r_0$. The search directions for the corresponding approximations $\boldsymbol{x}_k$ can be constructed in a similar way. In CGS the BiCG residuals and search directions themselves are not computed.

The matrix polynomial $\phi_k(A)$ reduces the initial residual r_0 (in case of convergence) and one may hope that this polynomial reduces the vector $\phi_k(A)r_0$ even further. Unfortunately, especially in the early phases of the process there is often irregular convergence in BiCG, and this is then squared in CGS, so that large residuals and very irregular convergence may be expected [25].

The accuracy $|\|\boldsymbol{r}_k\| - \|b - A\boldsymbol{x}_k\|\|$ in the final approximation $\boldsymbol{x}_k$ for CGS, and other methods to be discussed here, is proportional to the largest residual norm [22]:

$$\left|\|\boldsymbol{r}_k\| - \|b - A\boldsymbol{x}_k\|\right| \leqslant k\bar{\xi}\Gamma \max_{j \leqslant k} \|\boldsymbol{r}_j\|, \tag{5}$$

where $\|\cdot\|$ is the Euclidean norm, $\bar{\xi}$ is the relative machine precision, and Γ is a modest multiple of the condition number of A. In exact arithmetic $|\|r_k\| - \|b - Ax_k\|| = 0$. Apparently it is desirable to avoid large, intermediate, residual norms.

By very irregular convergence we refer to the situation where the norms of successive residual vectors differ in order of magnitude in norm. For instance when $\|r_k\| \ll \|r_{k-1}\|$, the recurrence relation for the computation of r_k may be perturbed by rounding errors that are relatively small with respect to $\|r_{k-1}\|$ but that are relatively large with respect to $\|r_k\|$. Experiments show that such local, and relatively large errors may affect the speed of convergence of the underlying BiCG process, or in other words, local bi-orthogonality of residuals and shadow residuals seems to be essential for the convergence of BiCG (cf. [9,11]).

In [21] several strategies are discussed that minimize the negative effects of large intermediate residuals on the accuracy. These strategies are relatively inexpensive, require only a few additional lines of code, and work for other hybrid BiCG methods as well. Although they lead to more accurate approximations, they leave the speed of convergence (essentially) unchanged. These strategies do not introduce significant new errors, but, since they work a posteriori on residual vectors and approximations, they also do not reduce these errors. Of course such strategies can be combined by our approaches to improve accuracy of the BiCG iteration coefficients.

3. Bi-CGSTAB

The Bi-CGSTAB algorithm in [25] tries to avoid large residuals and irregular convergence by choosing ψ_k as a product of linear factors that minimize the residuals locally in contrast to the BiCG polynomials that "aim" for global minimization. If ψ_k is the polynomial in Bi-CGSTAB in step k then

$$\psi_{k+1}(t) = (1 - \omega_k t)\psi_k(t) \tag{6}$$

where, for

$$\hat{r} := \psi_k(A) r_{k+1}, \tag{7}$$

ω_k minimizes $\|(I - \omega_k A)\hat{r}\|$:

$$\omega_k = \omega_k^{\mathrm{MR}} := \frac{(\hat{r}, A\hat{r})}{(A\hat{r}, A\hat{r})}. \tag{8}$$

Bi-CGSTAB shares the advantages of CGS: its steps are (almost) equally inexpensive and it is transpose free. Moreover, it often converges (much) faster and it is more accurate.

Unfortunately, there are also situations where CGS performs reasonable well, while Bi-CGSTAB converges poorly or even stagnates. This often happens for discretized diffusion-advection equations with large advection terms, when using real arithmetic. In such situations, some ω_k^{MR} may be expected to be relatively small, that is

$$|\hat{\omega}_k| \ll 1 \quad \text{where } \hat{\omega}_k := \frac{(\hat{r}, A\hat{r})}{\|\hat{r}\|\|A\hat{r}\|}. \tag{9}$$

In finite precision arithmetic, this leads to inaccurate BiCG coefficients [12, 19]. As may be expected, and as experiments affirm [22], inaccuracies on these coefficients may seriously deteriorate the speed of convergence. Although this would help to explain the poor performance of Bi-CGSTAB in the situations mentioned above, an inspection of experimental results reveals that stagnation due to inaccurate BiCG coefficients also may occur in situations where none of the ω_k^{MR} is relatively small (say, $|\widehat{\omega}_k| > 0.02$ for all k; cf. Fig. 1 and [20]).

4. Towards accurate BiCG coefficients in hybrid BiCG methods

To understand why the BiCG coefficients can be inaccurate, we concentrate on ρ_k (see (3)). Similar arguments also apply to σ_k (in (2)); for more details, see [20]. The value of ρ_k will be inaccurate if it is relatively small with respect to $\|r_k\|\|\tilde{r}_0\|$. The question is, when does this occur and can it be avoided? As is well known, it will happen if the incorporated Lanczos process nearly breaks down (i.e. $(\phi_k(A)r_0, \psi_k(A^{\mathrm{T}})\tilde{r}_0) \approx 0$ for any polynomial ψ_k of exact degree k). But an "unlucky" choice of ψ_k may lead to a relatively small ρ_k as well. Here, we concentrate on a typical situation for hybrid BiCG, that is, we concentrate on the effect of the chosen polynomials ψ_k. Therefore, we assume that the Lanczos process does not (nearly) break down.

In hybrid methods that exploit (3), like Bi-CGSTAB, the rounding error ε_k in ρ_k can relatively and sharply be bounded by

$$|\varepsilon_k| \leqslant n\bar{\xi}\frac{(|r_k|, |\tilde{r}_0|)}{|(r_k, \tilde{r}_0)|} \leqslant n\bar{\xi}\frac{\|r_k\|\|\tilde{r}_0\|}{|(r_k, \tilde{r}_0)|} \leqslant \frac{n\bar{\xi}}{\widehat{\rho}_k} \quad \text{where } \widehat{\rho}_k := \frac{|(r_k, \tilde{r}_0)|}{\|r_k\|\|\tilde{r}_0\|}, \tag{10}$$

$\bar{\xi}$ is the relative machine precision, and n is the dimension of the problem. For accuracy reasons we would like to have $\widehat{\rho}_k$ (the scaled ρ_k) as large as possible. Since r_k is orthogonal to $\mathcal{K}_k(A^{\mathrm{T}}; \tilde{r}_0)$, we see that (at least in exact arithmetic) ρ_k depends on ψ_k only through its leading coefficient θ_k:

$$\rho_k = (r_k, \psi_k(A^{\mathrm{T}})\tilde{r}_0) = (r_k, \theta_k(A^{\mathrm{T}})^k\tilde{r}_0)$$

(cf. (3)). Hence,

$$\widehat{\rho}_k = \frac{|\theta_k(A^k r_k, \tilde{r}_0)|}{\|r_k\|\|\tilde{r}_0\|} = \frac{|\theta_k|}{\|\psi_k(A)r_k\|} \cdot \frac{|(A^k r_k, \tilde{r}_0)|}{\|\tilde{r}_0\|}. \tag{11}$$

Therefore, we would like to use the polynomial ψ_k for which $\|r_k\|/|\theta_k| = \|\psi_k(A)r_k\|/|\theta_k|$ is minimal. Since ψ_k is a residual polynomial (that is, r_k corresponds to an approximate solution x_k), we have that $\psi_k(0) = 1$. The minimization problem is solved by the *OR polynomial* ψ_k^{OR} that is characterized by the property

$$\psi_k^{\mathrm{OR}}(A)r_k \perp \mathcal{K}_k(A; r_k), \qquad \psi_k^{\mathrm{OR}} \text{ is of degree } k, \text{ and } \psi_k^{\mathrm{OR}}(0) = 1$$

(see [20]).

The OR polynomial defines the residual $s_k^{\mathrm{OR}} = \psi_k^{\mathrm{OR}}(A)s_0$ at the kth step of orthogonal residual methods, like FOM [17] or GENCG [6] (here with initial residual $s_0 = r_k$). On the other hand, the *MR polynomial* ψ_k^{MR} minimizes

$$\|r_k\| = \|\psi_k(A)r_k\|$$

over all polynomials ψ_k of degree k for which $\psi_k(0) = 1$: the MR polynomial defines the residual $s_k^{MR} = \psi_k^{MR}(A)s_0$ at the kth step of minimal residual methods as GMRES [18] or GCR [6] (here also with initial residual $s_0 = r_k$). The MR polynomial seems to be more appropriate for creating residuals r_k that are small with respect to the Euclidean norm. Obviously, we would like to avoid to do k steps of FOM or GMRES for the computation of $r_k = \psi_k(A)r_k$ (using $s_0 = r_k$ as initial residual), since this would be too expensive for large k. The product of MR(1) polynomials (i.e. MR polynomials of degree 1), as in Bi-CGSTAB, is a compromise between the wishes for small residuals and for inexpensive steps. Although OR(1) polynomials (cf. (6), now with $\omega_k = \omega_k^{OR} = (\hat{r}, \hat{r})/(\hat{r}, A\hat{r})$ such that $(I - \omega_k^{OR}A)\hat{r} \perp \hat{r}$) occasionally cure stagnation of Bi-CGSTAB (see Section 5 and also [4]), they also may amplify residuals with respect to $\|\cdot\|$, which may lead to inaccurate approximations (as explained by (5)) or even to overflow.

5. Stabilizing Bi-CGSTAB

In this section, we concentrate on the case where ψ_{k+1} is constructed as the product of polynomials of degree 1. Furthermore, ψ_k, $\hat{r}$ and $\hat{\omega}_k$ are as defined in Section 3.

Since (cf. (11))

$$\hat{\rho}_{k+1} = \frac{|\theta_{k+1}(A^{k+1}r_{k+1}, \tilde{r}_0)|}{\|r_{k+1}\|\|\tilde{r}_0\|} = \frac{|\omega_k|}{\|(I - \omega_k A)\hat{r}\|} \cdot \frac{|\theta_k||(A^{k+1}r_{k+1}, \tilde{r}_0)|}{\|\tilde{r}_0\|}, \tag{12}$$

we may expect to obtain the most accurate BiCG coefficients when using degree 1 factors if we take the ω_k for which $\|(I - \omega_k A)\hat{r}\|/|\omega_k|$ is minimal.

If the angle between $\hat{r}$ and $A\hat{r}$ (see (7)) is larger than 45° (i.e. $|\hat{\omega}_k| < \frac{1}{2}\sqrt{2}$, cf. (9)) then the OR(1) polynomial locally amplifies the residual,

$$\|s_1^{OR}\| > \|\hat{r}\| > \|s_1^{MR}\|, \tag{13}$$

where

$$s_1^{OR} := (I - \omega_k^{OR}A)\hat{r} \quad \text{and} \quad s_1^{MR} := (I - \omega_k^{MR}A)\hat{r}, \tag{14}$$

while the MR(1) polynomial locally amplifies inaccuracies in ρ_{k+1} (cf. (10) and (12)),

$$\frac{\|s_1^{MR}\|}{|\omega_k^{MR}|} = \frac{\|(I - \omega_k^{MR}A)\hat{r}\|}{|\omega_k^{MR}|} > \|A\hat{r}\| > \frac{\|s_1^{OR}\|}{|\omega_k^{OR}|} = \frac{\|(I - \omega_k^{OR}A)\hat{r}\|}{|\omega_k^{OR}|} \tag{15}$$

(see [20] for a detailed explanation). We should worry about such amplifications if they are extremely large or if they occur in a consecutive number of steps as will be the case in a stagnation phase of the process. For different reasons, both choices $r_{k+1} = s_1^{MR}$ and $r_{k+1} = s_1^{OR}$, may slow down the convergence: OR(1) polynomials by amplifying $\|r_{k+1}\|$; MR(1) polynomials by reducing $\hat{\rho}_{k+1}$ (cf. (10)) and thus affecting the accuracy of the BiCG coefficients. The reducing effect of MR(1) polynomials, due to small $|\hat{\omega}_k|$, often seems to accumulate (cf. [20]). In examples, for Bi-CGSTAB, one may observe that $\hat{\rho}_{k+1}$ is more or less proportional to $\prod_{j\leq k}|\hat{\omega}_j|$: $|\hat{\omega}_k|$ is the factor by which the MR(1) polynomial reduces $\hat{\rho}_{k+1}$ as compared with the OR(1) polynomial (see (12) and (16) below). Apparently, Bi-CGSTAB may converge poorly not only if $|\omega_k^{MR}|$ is small but also if in a consecutive number of steps $|\omega_k^{MR}|$ is relatively less than, say 0.7, i.e. $|\hat{\omega}_k| < 0.7$ (cf. (9)).

The following property expresses the opposite effects of the OR(1) polynomial and the MR(1) polynomial (cf. [20, Corollary 3.1]):

$$\|s_1^{\mathrm{OR}}\| = \frac{1}{|\widehat{\omega}_k|}\|s_1^{\mathrm{MR}}\|, \qquad \frac{\|s_1^{\mathrm{OR}}\|}{|\omega_k^{\mathrm{OR}}|} = |\widehat{\omega}_k|\frac{\|s_1^{\mathrm{MR}}\|}{|\omega_k^{\mathrm{MR}}|}, \qquad \sqrt{1-|\widehat{\omega}_k|^2} = \frac{\|s_1^{\mathrm{MR}}\|}{\|\widehat{r}\|}. \tag{16}$$

Especially if in Bi-CGSTAB, the MR(1) part almost stagnates in a consecutive number of steps, it is important to have accurate BiCG coefficients, because in such a case further convergence may only be expected from the BiCG part of Bi-CGSTAB. Unfortunately, as explained above, this is precisely the situation where the MR(1) polynomials would spoil the accuracy of the coefficients. In that case the OR(1) polynomials spoil the accuracy of the approximations or lead to overflow. Therefore, we have to find a cure in other modifications.

With $\widehat{\omega}_k$ as in (9), we propose to take

$$\omega_k = \frac{\widehat{\omega}_k}{|\widehat{\omega}_k|}\max\left(|\widehat{\omega}_k|, 0.7\right)\frac{\|\widehat{r}\|}{\|A\widehat{r}\|}, \qquad r_{k+1} = (I-\omega_k A)\widehat{r}. \tag{17}$$

Since

$$\omega_k^{\mathrm{MR}} = \widehat{\omega}_k\frac{\|\widehat{r}\|}{\|A\widehat{r}\|}, \qquad \omega_k^{\mathrm{OR}} = \frac{1}{\widehat{\omega}_k}\frac{\|\widehat{r}\|}{\|A\widehat{r}\|}, \tag{18}$$

we see how the choice in (17) combines the OR(1) and the MR(1) polynomial. The resulting first degree polynomial $1-\omega_k t$ only mildly amplifies both $\|r_{k+1}\|$ and the rounding errors in the BiCG coefficients in situations where the OR(1) polynomial would strongly amplify $\|r_{k+1}\|$ and where the MR(1) polynomial would strongly amplify the rounding errors. However, although, this often cures our problems, it cannot always prevent completely a poor convergence of Bi-CGSTAB (see Section 9). For a more rigorous explanation for the factor 0.7 in (17), see the end of Section 7.

6. BiCGstab(l)

In BiCGstab(l) [19] (see also [12,22]) the polynomial ψ_k is constructed as a product of polynomials of degree l: for $k = ml$, we have that $\psi_{k+l} = p^{(m)}\cdot\psi_k = p^{(m)}\cdot\ldots\cdot p^{(0)}$ where, for

$$\widehat{r} := \psi_{ml}(A)r_{ml+l} = \psi_{ml}(A)\phi_{ml+l}(A)r_0, \tag{19}$$

the polynomial $p^{(m)} = p_l^{\mathrm{MR}}$ minimizes $\|p^{(m)}(A)\widehat{r}\|$ over all polynomials $p^{(m)}$ of degree l for which $p^{(m)}(0) = 1$. The residual r_{ml+l} at the $(m+1)$st sweep of BiCGstab(l) is of the form $r_{ml+l} = p_l^{\mathrm{MR}}(A)\widehat{r}$, the polynomial ψ_{ml+l} is given by $\psi_{ml+l} = p_l^{\mathrm{MR}}\cdot\psi_{ml}$.

The implementations in [19,22] use simple polynomials like t^i in intermediate steps, that is, for $k = ml+i$, $i \neq 0$, they compute ρ_k from $A^i\widehat{r}$ (cf. (2)). Thus they avoid the breakdown problems in the intermediate steps that would have occurred from degenerated MR polynomials. If, for $i = 1$, the angle between $\widehat{r}$ and $A\widehat{r}$ is larger than 45° then the inequalities in (15) are now also valid. The first inequality in (15) shows that the first degree polynomial t may lead to more accurate ρ_{ml+1} than the MR(1) polynomial. A similar statement can also be made with respect to polynomials of degree $i > 1$. Moreover, even if the $\widehat{\rho}_{ml+i}$ decrease in the intermediate steps, this does not affect the accuracy of the BiCG coefficients in the next sweep. Any intermediate reduction plays no role since

$\widehat{\rho}_{ml+l}$ depends only on the new polynomial $p^{(m)}$ and, of course, the value of $\widehat{\rho}_{ml+l}$ depends on the previous polynomials as well ($\psi_{ml+l} = p^{(m)} \cdot \psi_{ml}$). Apparently, more accurate BiCG coefficients are to be expected from BiCGstab(l) with $l > 1$ than from Bi-CGSTAB (=BiCGstab(1)), also since it is less likely that $p^{(m)}$ will lead to stagnation for larger values of l.

To compare the convergence of Bi-CGSTAB with BiCGstab(l), we should compare the results of m sweeps of BiCGstab(l) with the results of ml sweeps of Bi-CGSTAB: then, for both methods, the residual vectors can be expressed by a polynomial in A of degree $2ml$ applied to r_0. Moreover, also the computational costs are almost equal in this case.

For two reasons, we may expect better convergence for BiCGstab(l):

(1) One sweep of MR(l) may be expected to result in better reduction of the norm of the residual than l steps of MR(1), because of the superlinear convergence of the MR approximations (note that GMRES is an implementation of the MR approach),

(2) l steps of MR(1) may contribute l times to a decrease in $\widehat{\rho}_k$ (hence contributing l times to increasingly larger rounding errors in ρ_k), while one sweep of MR(l) contributes only once; the decreasing effect in each single step of MR(1) may be expected to be comparable or worse than the effect of only one sweep by MR(l).

BiCGstab(l) is an improvement over Bi-CGSTAB both in the MR part as well as in the BiCG part: (1) the MR part will produce residuals with smaller norm and (2) the convergence of exact BiCG will be better maintained. In many applications, the speed of convergence of the BiCGstab methods appears to be determined largely by the underlying BiCG process and then a better recovered BiCG part is much more important than the MR reductions. In these situations, the $\widehat{\rho}_k$ for Bi-CGSTAB may reduce to the order of magnitude of $\overline{\xi}$. In such a case, none of the digits of the BiCG coefficients is correct and BiCG, and consequently Bi-CGSTAB, does not converge. Taking $l = 2$ often suffices to keep $\widehat{\rho}_k$ large enough for convergence, larger than, say, $\overline{\xi}^{1/3}$ (cf. [9, Section 4]). Occasionally, a larger l ($l = 4$ or even $l = 8$) is necessary.

Although the occurrence of MR(l) polynomials with a relatively small leading coefficient may be expected to be less for larger l, almost degenerated MR polynomials or small leading coefficients in a consecutive number of sweeps may still spoil the accuracy of the BiCG coefficients. There are two obvious approaches for avoiding such a situation: select another polynomial $p^{(m)}$ of degree l, or increase l even further. In Section 8, we will discuss strategies for adapting the value of l automatically (that is, the value of l may vary per sweep). In the next section, we will propose polynomials $p^{(m)}$ of degree l that are more appropriate than p_l^{MR}, similar to our approach to stabilize Bi-CGSTAB.

7. Combining MR and OR polynomials

Also for $l > 1$, one may consider OR polynomials as an alternative for (almost) degenerated MR polynomials. That is, one may take $p^{(m)} = p_l^{\mathrm{OR}}$, where p_l^{OR} is the polynomial of degree l for which $p_l^{\mathrm{OR}}(0) = 1$ and $s_l^{\mathrm{OR}} := p_l^{\mathrm{OR}}(A)\widehat{r} \perp \mathcal{K}_l(A;\widehat{r})$. The OR($l$) polynomial minimizes $\|p^{(m)}(A)\widehat{r}\|/|\omega|$, where ω is the leading coefficient of $p^{(m)}$. Therefore, among all polynomials of degree l, the OR(l) polynomials may be expected to lead to the most accurate BiCG coefficients (cf. Section 4 and (12) in Section 5). Similarly as for $l = 1$ (see (16)), the OR polynomial and the MR polynomial have opposite effects: with $s_l^{\mathrm{MR}} := p_l^{\mathrm{MR}}(A)\widehat{r}$, we can state the following result (cf. [20, Corollary 3.2]).

Theorem 1 ([20, Corollary 3.2], see also [3,26]). *For some $\widehat{\omega} \in (0,1]$, we have*

$$\|s_l^{\mathrm{OR}}\| = \frac{1}{\widehat{\omega}}\|s_l^{\mathrm{MR}}\|, \qquad \frac{\|s_l^{\mathrm{OR}}\|}{|\omega^{\mathrm{OR}}|} = \widehat{\omega}\frac{\|s_l^{\mathrm{MR}}\|}{|\omega^{\mathrm{MR}}|}, \qquad \sqrt{1-\widehat{\omega}^2} = \frac{\|s_l^{\mathrm{MR}}\|}{\|s_{l-1}^{\mathrm{MR}}\|}, \tag{20}$$

where ω^{OR} and ω^{MR} are the leading coefficients of p_l^{OR} and, respectively, p_l^{MR}.

The coefficient $\widehat{\omega} \geqslant 0$ is a scaled version of $|\omega^{\mathrm{MR}}|$ and it can be computed in BiCGstab(l) and in the modified version that we will now propose, at almost no additional computational costs.

As for $l = 1$ (cf. (17)), one can compromise between the MR polynomial and the OR polynomial in order to obtain more accurate BiCG coefficients without amplifying the residuals with respect to $\|\cdot\|$. We suggest to take for r_{ml+l},

$$r_{ml+l} = s_{l-1}^{\mathrm{MR}} - \frac{\gamma}{\widehat{\omega}}\left(s_{l-1}^{\mathrm{MR}} - s_l^{\mathrm{MR}}\right), \tag{21}$$

where

$$\widehat{\omega} > 0 \text{ is such that } \sqrt{1-\widehat{\omega}^2} = \frac{\|s_l^{\mathrm{MR}}\|}{\|s_{l-1}^{\mathrm{MR}}\|} \text{ and } \gamma := \max(\widehat{\omega}, 0.7), \tag{22}$$

or, equivalently,

$$p^{(m)} = \frac{1-\widehat{\omega}\gamma}{1-\widehat{\omega}^2}p_l^{\mathrm{MR}} + \frac{\widehat{\omega}\gamma - \widehat{\omega}^2}{1-\widehat{\omega}^2}p_l^{\mathrm{OR}} \text{ where } \widehat{\omega} := \frac{\|s_l^{\mathrm{MR}}\|}{\|s_l^{\mathrm{OR}}\|} \text{ and } \gamma := \max(\widehat{\omega}, 0.7). \tag{23}$$

The relations (21) and (22) express the new residual in terms of MR residuals, while (23) shows that the polynomial $p^{(m)}$ is a convex combination of the MR polynomial and the OR polynomial. The equivalence is not trivial; for a proof we refer to [20, Theorem 3.1]. In [20] also suggestions are made for an efficient and stable implementation. Actually, the additional computational costs that are required for the computation of r_{ml+l} by (21), instead of $r_{ml+l} = s_l^{\mathrm{MR}}$, are negligible: only some operations with vectors and matrices of dimension smaller than $l+1$ are involved.

Observe that for $l = 1$ we have precisely the compromise as suggested in (17), since $s_0^{\mathrm{MR}} = \widehat{r}$.

Of course, r_{ml+l} can also be obtained as a convex combination of the MR(l) residual s_l^{MR} and the OR(l) residual s_l^{OR}: $r_{ml+l} = ((1-\widehat{\omega}\gamma)s_l^{\mathrm{MR}} + (\widehat{\omega}\gamma - \widehat{\omega}^2)s_l^{\mathrm{OR}})/(1-\widehat{\omega}^2)$ (cf. (23)).

If, in the expression for $p^{(m)}$ in (23), we take $\gamma = \widehat{\omega}$, then we obtain for $p^{(m)}$ the MR polynomial, while for $\gamma = 1/\widehat{\omega}$ we have the OR polynomial. The value $\gamma = 1$ would be some kind of average. We suggest to take for γ the maximum of $\widehat{\omega}$ and 0.7 in order to profit from the fact that the MR(l) polynomial reduces $\widehat{r}$ strongly with respect to $\|\cdot\|$ ($\|s_l^{\mathrm{MR}}\| \ll \|\widehat{r}\|$) if $\widehat{\omega} \approx 1$, and we still avoid the situation where the MR polynomial may lead to large rounding errors in the BiCG coefficients in the case of (almost) stagnation in the lth step of MR (then $\|s_l^{\mathrm{MR}}\| \approx \|s_{l-1}^{\mathrm{MR}}\|$ and $\widehat{\omega} \ll 1$). Obviously, the maximum of $\widehat{\omega}$ and any other nonsmall positive constant Ω less than 1 would have a similar effect; in our experiments the choice $\Omega = 0.7$ was quite satisfactorily.

8. Varying l

If, precisely in the lth step, MR does not reduce the residual well with respect to $\|\cdot\|$, then more accurate BiCG coefficients and faster convergence are to be expected for larger l where such a

reduction does happen. If, on the other hand, the lth step of MR reduces well, then the reduction by OR(l) will be comparable (since then $\widehat{\omega} \approx 1$ and $\|s_l^{\mathrm{MR}}\| \approx \|s_l^{\mathrm{OR}}\|$; cf. (20)), and it is not necessary to increase l for improving the accuracy. Counting only the costs associated with n-vectors, i.e. vectors of the dimension of the original problem, BiCGstab(l) requires $2l + 10$ AXPYs (vector updates) and $l + 7$ DOTs (inner products) per 4 MVs (matrix–vector multiplications). Furthermore, storage is required for $2l + 5$ n-vectors (cf. [22, Section 8]). The costs associated with vectors of dimension $\leqslant l + 1$ are negligible. Typical choices for l are $l = 1$ (Bi-CGSTAB), 2, 4, or 8. Therefore, it is only slightly more expensive to use an l that is somewhat larger. Moreover, due to large intermediate residuals (see [22]), much larger l lead to less accurate approximations (cf. (5)). For these reasons, it is advisable to keep l limited (say $\leqslant 8$).

We will discuss briefly several strategies for the dynamical determination of l.

In each sweep, we start with r_k and $l = 1$, and we increase l to 2, or to 3, . . ., up to $l = l_{\max}$, if accurate computation of the BiCG coefficients seems to require such an increase. We try to avoid too small values for $\widehat{\rho}_k$, say less than $\delta := \bar{\xi}^{1/2}$ ($\bar{\xi}$ is the relative machine precision).

As experiments indicate, $l_{\max} = 8$ seems to be an appropriate value.

The choice $\delta = \bar{\xi}^{1/2}$ expresses our aim to keep the local rounding errors in the BiCG coefficients relatively less than this value. We have experimental evidence that the BiCG recurrence relations may be perturbed by errors of this order of magnitude without affecting the convergence of BiCG too much [9, 20]; see [11] for an explanation.

8.1. *Using the size of* $\widehat{\rho}_k$

To avoid a small $\widehat{\rho}_{k+l}$, it is tempting to

$$\text{increase } l \quad \text{if } l \leqslant l_{\max} \text{ and if } \widehat{\rho}_{k+l} = \frac{|(r_{k+l}, \tilde{r}_0)|}{\|r_{k+l}\|\|\tilde{r}_0\|} \leqslant \delta. \tag{24}$$

This criterion might lead to additional costs, since r_{k+l} may not be needed when l has to be decreased. Moreover, we should increase l *before* $\widehat{\rho}_{k+l}$ is too small and not *afterwards*. Although correction afterwards may seem doubtful, numerical experiments indicate that the a posteriori approach works well: it prevents future $\widehat{\rho}_k$ to become much smaller than δ. We do not have a theoretical explanation for this phenomenon.

The idea of using $\widehat{\rho}_k$ for monitoring the performance of Bi-CGSTAB can also be found in [14], where a restart is suggested when $\widehat{\rho}_k$ is less than δ. As has also been observed in [14], a restart is not likely to cure stagnation, since after the restart $\widehat{\rho}_k$ may again rapidly be smaller than δ.

In [2, 13], $\widehat{\sigma}_k$ is used in BiCG (Lanczos), and CGS, to make a decision for look-ahead steps (for the BiCG part).

8.2. *Computational details and costs*

Obviously, we have to compute $\|\tilde{r}_0\|$ only once in the iterative process.

In the implementation of BiCGstab(l) in [19], the vectors $\widehat{r}_i, A\widehat{r}_i, \ldots, A^l\widehat{r}_i$ with $\widehat{r}_i := \psi_k(A)r_{k+i}$ are computed explicitly in the intermediate steps $k + i$, $i = 1, \ldots, l$. This is done very efficiently. The residual r_{k+l} is not formed until $i = l$. Then, for $\boldsymbol{R} := [\widehat{r}_l \mid A\widehat{r}_l \mid \ldots \mid A^l\widehat{r}_l]$ (note that $\widehat{r} = \widehat{r}_l$; cf. (19)), the vector $y^{\mathrm{MR}} \in \mathbb{R}^{l+1}$ is determined for which $r_{k+l} = \boldsymbol{R}y^{\mathrm{MR}}$, and r_{k+l} is computed. Except for the

AXPYs in the computation of r_{k+l} from Ry^{MR} and the DOTs in $V := R^T R$, all other computations for y^{MR} require only operations associated with vectors of dimension $l+1$ at most. For instance, the "minimal residual" y^{MR} will appear as the solution of a least square problem that can be solved from normal equations that can be formulated in terms of the $l \times l$ right lower block of V. A similar observation can be made for the computation of the r_{k+l} given by (21): the vector y and the scalar $\hat{\omega}$ needed for this $r_{k+l} = Ry$, the leading coefficient ω of the polynomial $p^{(m)}$ in (23), and even the norm $\|r_{k+l}\| = \sqrt{y^T V y}$ can be computed from the elements in V (cf. [20]). Since

$$(r_{k+l}, \tilde{r}_0) = \omega(A^l \hat{r}_l, \tilde{r}_0),$$

and since the inner product $(A^l \hat{r}_l, \tilde{r}_0)$ has to be computed anyway, the value $\hat{\rho}_{k+l}$ in (24) can be computed from V (and $(A^l \hat{r}_l, \tilde{r}_0)$). If the present value of l is to be preferred, then no additional costs are involved; otherwise, if l has to be increased then the computation of V requires another $\frac{1}{2}(l+1)(l+2)$ DOTs.

In one sweep of BiCGstab(l), checking the criterion in (24) would require an additional number of $\sum_{1<j\leqslant l} \frac{1}{2} j(j+1) = \frac{1}{6} l(l+1)(l+2) - 1$ DOTs. As compared with Bi-CGSTAB, BiCGstab(2) requires an additional 2 AXPYs and 1 DOT per 4 MVs, while an $l = 1$ step with (24) requires no additional work, and an $l = 2$ step requires another 2 AXPYs and 2.5 DOTs per 4 MVs. Apart from the fact that we do not have to worry about the choice of l if we use a criterion like (24), we may save some computational costs. If $l_{max} = 2$ and when the strategy with variable l converges as fast as BiCGstab(2), then we may save computational costs as soon as 34% of the steps are $l = 1$ steps.

By skipping the odd values for l, we may save computational costs too. Because, for real-valued problems the situation of complex eigenvalues of A with relatively large imaginary parts may be expected to be captured by polynomials of even degree. For such a situation, we expect no improvement by increasing the degree of the polynomial by only 1 to a polynomial of odd degree. Experiments seem to justify this point of view.

8.3. An efficient approach

Although the computation of $\|r_{k+l}\|$ does not require additional AXPYs, especially if larger values of l are allowed ($l_{max} \geqslant 8$), it may become expensive. The following criterion is relatively cheap (per 4 MV, an additional 2 DOTs for computing $\|A^l \hat{r}_l\|$).

$$\text{Increase } l \quad \text{if } l \leqslant l_{max} \text{ and if } \hat{\rho}'_{k+l} := \frac{|(A^l \hat{r}_l, \tilde{r}_0)|}{\|A^l \hat{r}_l\| \|\tilde{r}_0\|} \leqslant \delta. \tag{25}$$

The value of the scalar $\hat{\rho}'_{k+i}$ indicates the accuracy of ρ_{k+i} for intermediate steps where ρ_{k+i} is computed from $A^i \hat{r}_i$ (see the discussion in the second paragraph of Section 6). Since $\|p^{(m)}(A)\hat{r}_i\|/|\omega|$ may expected to be (much) smaller than $\|A^l \hat{r}_l\|$, the value of $\hat{\rho}_{k+l}$ will be larger than δ whenever $\hat{\rho}'_{k+l} \geqslant \delta$. Therefore, as compared with criterion (24), this criterion might lead to sweeps with larger values of l.

8.4. Using the leading coefficients of MR polynomials

The most obvious strategy to choose l would be to increase l as long as $\hat{\omega}$ (cf. (22)) is not large enough. However, it is not clear what "$\hat{\omega}$ is not large enough" means. Obviously we should avoid

extremely small $\widehat{\omega}$ (e.g. $\widehat{\omega} \approx \overline{\xi}$), but we should also avoid consecutive $\widehat{\omega}$'s that are smaller than, say, 0.7. Unfortunately, we only know that $\widehat{\omega}$ is too small when it is too late: if $\widehat{\rho}_{k+l} < \delta$ then, apparently, $\widehat{\omega}$'s from preceding sweeps have been too small.

Now, recall that, for $l = 1$, often $\widehat{\rho}_k$ seems to decrease proportionally to the product of preceding $|\widehat{\omega}_j|$ (see Section 5). A similar observation can be made for larger l ($\widehat{\omega} = |\widehat{\omega}_j|$ for $l = 1$).

If $\widehat{\rho}_k$ is not small (say, of order 0.1) we may permit a significant reduction of $\widehat{\rho}_{k+l}$ (say, with an $\widehat{\omega} \approx 0.1$). On the other hand, if $\widehat{\rho}_k$ is small (say, of order 10^{-6}) we should avoid an additional reduction ($\widehat{\omega}$ should be larger than, say 0.7). Moreover, from a consecutive number of $l = 1$ steps with $\widehat{\omega} \approx |\widehat{\omega}_0|$ we expect a similar reduction as from a consecutive number of l steps with $\widehat{\omega} \approx |\widehat{\omega}_0|^l$. Therefore, for larger l we may allow smaller $\widehat{\omega}$ compared with the $\widehat{\omega}$ for $l = 1$. Our next criterion takes these observations into account.

$$\text{Increase } l \quad \text{if } l \leqslant l_{\max} \text{ and if } \widehat{\omega}^{\nu(l)} \leqslant \left(\frac{\delta}{\widehat{\rho}_{k+l}}\right)^{\upsilon}, \tag{26}$$

with, say $\nu(l) := 2/(l+1)$ and $\upsilon := \frac{1}{8}$.

With $\nu(l) = 1$ we do not allow smaller $\widehat{\omega}$ for larger l, while with $\nu(l) = 1/l$ we may anticipate not to improve the accuracy for larger l. Our suggestion $\nu(l) = 2/(l+1)$ is a compromise.

To apply the criterion in (26), we need $\widehat{\omega}$ and ρ_{k+l}. As explained in Section 8.2, these values can be computed from V.

9. Numerical experiments

In this section, we will focus on the effects of varying l using the criteria (24)–(26), but we will also see numerical illustrations of our observations in the other sections of this paper.

We use the same test problems (linear problems) as in [20].

As we have argued before, the numerical problems in computing the BiCG coefficients typically occur in stagnation phases of the hybrid iteration process. Such phases are quite common in real-life problems. In order to make these also visible for simple model problems we have omitted preconditioning.

All figures display:

- the $\log_{10}$ of the norm of the true residual $b - Ax_k$ for $k = ml$ (———);
- the $\log_{10}$ of $\widehat{\rho}_k = |(r_k, \tilde{r}_0)|/(\|r_k\|\|\tilde{r}_0\|)$, also for intermediate k (– · – ·);
- the $\log_{10}$ of $\widehat{\sigma}_k = |(Au_k, \tilde{r}_0)|/(\|Au_k\|\|\tilde{r}_0\|)$ (· · · · · ·).

If the value of l was allowed to vary per sweep, then the figures also show:

- the value of l per sweep (indicated by +'s: the +'s give the values of $-l$, with l as used at the sweep started at the $2k$th MV).

Finally, Figs. 1–3 show

- the $\log_{10}$ of $\widehat{\omega}$ with $\widehat{\omega}$ as in (22), and $\widehat{\omega} = |\widehat{\omega}_k|$ for $l = 1$ (indicated by •'s).

The scalar $\widehat{\sigma}_k$ is the scaled version of σ_k (cf. (2), (3) and (10)). From the figures, we see that, for our test problems, the coefficients $\widehat{\rho}_k$ and $\widehat{\sigma}_k$ are almost similar and our approaches to minimize rounding errors in ρ_k have about the same effect on σ_k.

All computations were carried out in finite precision arithmetic with relative machine precision $\overline{\xi} = 2.2\,10^{-16}$.

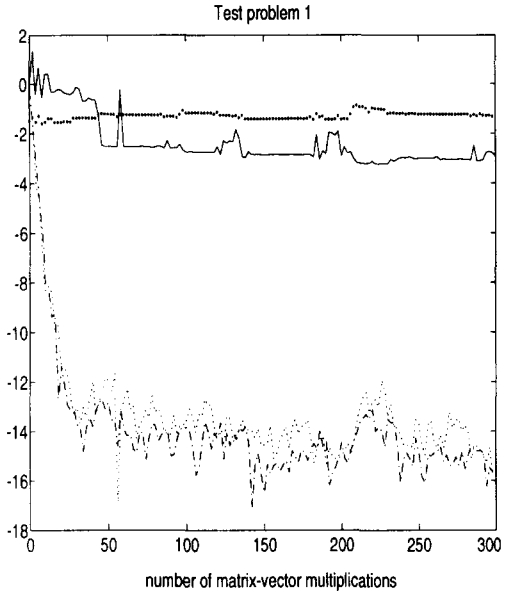

Fig. 1. Convergence Bi-CGSTAB.

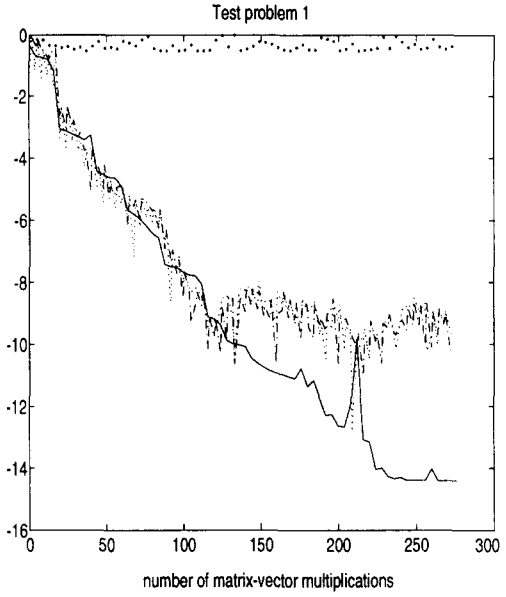

Fig. 2. Convergence BiCGstab(2).

Test problems

For the *first test problem*, we considered the partial differential equation

$$-u_{xx} - u_{yy} - u_{zz} + 1000u_x = f, \tag{27}$$

defined on the unit cube with Dirichlet boundary conditions, where f is such that

$$u = \exp(xyz)\sin(\pi x)\sin(\pi y)\sin(\pi z)$$

is the solution. This equation was discretized by $10 \times 10 \times 10$ finite volumes and central differences for u_x, resulting in a 7-diagonal linear system of order 1000.

For the *second* and *third test problem*, we have discretized

$$-u_{xx} - u_{yy} + a(xu_x + yu_y) + bu = f, \tag{28}$$

on the unit square with Dirichlet boundary conditions, with 63×63 finite volumes, $a = 100$ and $b = -200$, and 66×66 finite volumes, $a = 1000$ and $b = 10$, respectively. The function f is such that the discrete solution is constant 1 on the grid.

9.1. *Taking larger values for l and combining MR with OR polynomials*

In this subsection, we briefly discuss Figs. 1–4, which illustrate the effect of replacing $l = 1$ by $l = 2$ and the effect of combining the MR polynomials with the OR polynomials. For an extensive discussion we refer to [20].

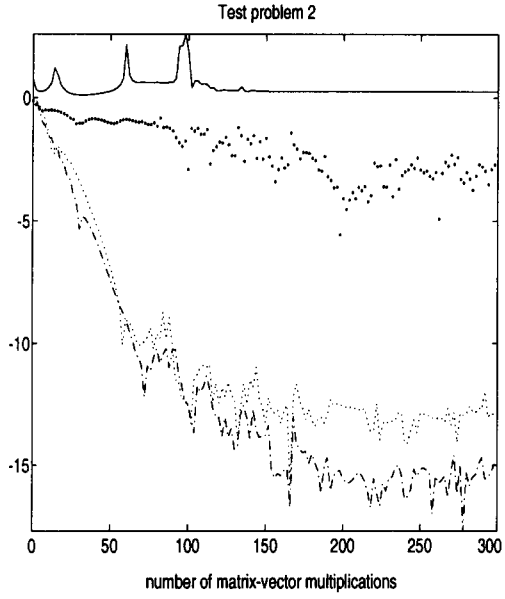

Fig. 3. Convergence Bi-CGSTAB.

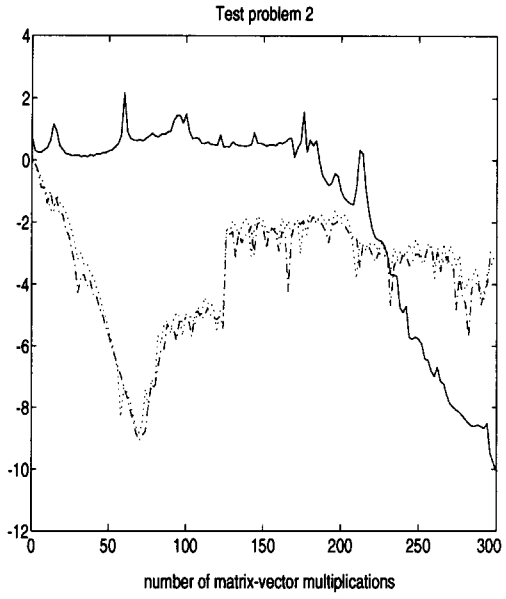

Fig. 4. Convergence stabilized Bi-CGSTAB.

In Figs. 1 and 3 we see that the scalars $\widehat{\rho}_k$ (and $\widehat{\sigma}_k$) decrease to values 10^{-14} or less, although the $|\widehat{\omega}_k|$ are not extremely small. At most two digits of the BiCG coefficients α_k and β_k may be expected to be correct if $\widehat{\rho}_k \approx 10^{-14}$, which explains well the stagnation of the method.

For the first test problem, as shown in Fig. 2, increasing l to $l = 2$, leads to convergence. The $\widehat{\omega}$'s in this test problem of BiCGstab(2) are larger than the $\widehat{\omega}$'s (= $|\widehat{\omega}_k|$'s) of Bi-CGSTAB, and the frequency of a contribution to an increase of the rounding errors is halved. The combination of these effects appears to be enough to prevent $\widehat{\rho}_k$ and $\widehat{\sigma}_k$ from becoming too small. Although, for the second test problem (of which the results are not displayed here), the values of the $\widehat{\rho}_k$ and $\widehat{\sigma}_k$ improve by switching to BiCGstab(2), this is not enough to avoid stagnation: the $\widehat{\omega}$'s for BiCGstab(2) are smaller than for Bi-CGSTAB, but this is not completely compensated for by halving the frequency of contribution.

For the second test problem, Fig. 4 shows the effect of *stabilizing*, that is, of combining MR(1) and OR(1) polynomials as explained in (17) (and (21)). For this second test problem, stabilizing keeps the values of $\widehat{\rho}_k$ and $\widehat{\sigma}_k$ large enough for sufficiently accurate BiCG coefficients. For the first test problem, the improvement from this approach (not shown here) was not enough to maintain convergence of the incorporated BiCG process. The amplification of the residual, though mildly, even led to divergence.

For the third test problem, a combination of increasing l to $l = 2$ with the stabilizing strategy (not shown here) was needed to maintain convergence (of BiCG).

Since the costs involved in stabilizing are negligible and the resulting BiCG coefficients may be significantly more accurate, we will employ this strategy in all our further experiments.

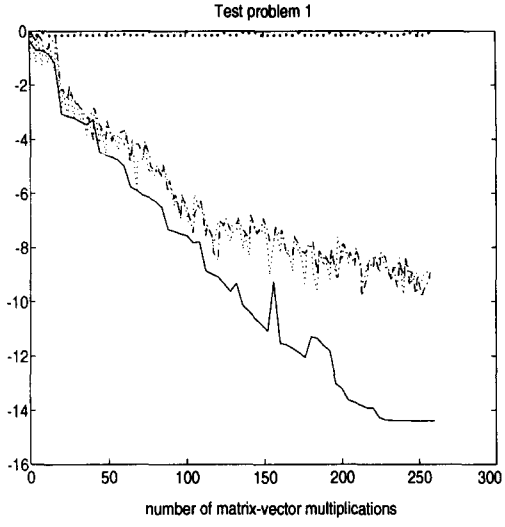

Fig. 5. Stabilized BiCGstab(2).

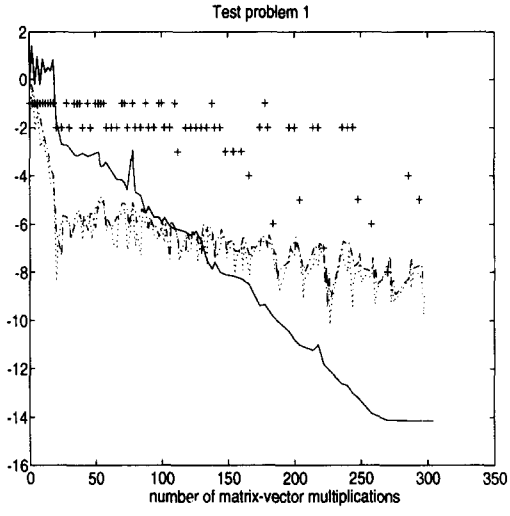

Fig. 6. Stab. BiCGstab($\{1:8\}$) using (24).

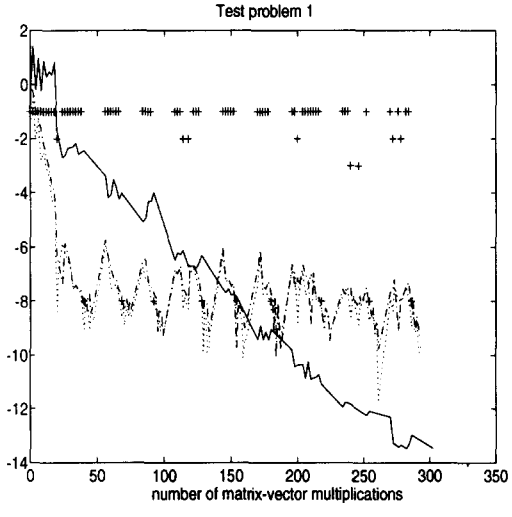

Fig. 7. Stab. BiCGstab($\{1:8\}$) using (25).

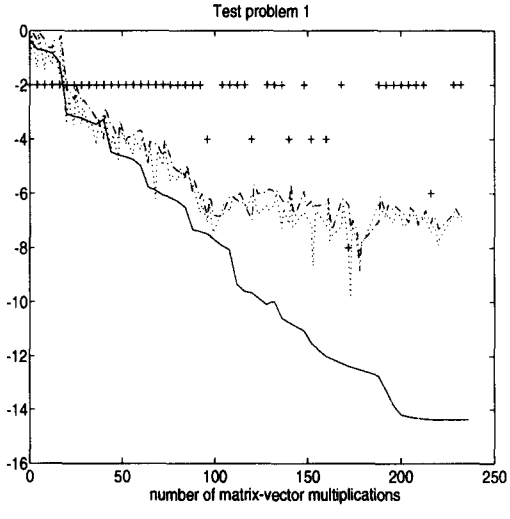

Fig. 8. Stab. BiCGstab($\{1:8\}$) using (26).

9.2. *Various increasing criteria*

For a subset $\mathbb{L}$ of $\mathbb{N}$, BiCGstab($\mathbb{L}$) is the BiCGstab(l) method in which we allow l to have a different value from the subset $\mathbb{L}$ in each sweep. In our experiments, $\mathbb{L} = \{j \in \mathbb{N} \mid j \leqslant 8\} = \{1:8\}$. We followed the strategy as explained in Section 8 using the criteria (24)–(26) with $\delta = \bar{\xi}^{1/2}$ ($\approx 10^{-8}$) and the values for ν and υ as suggested there.

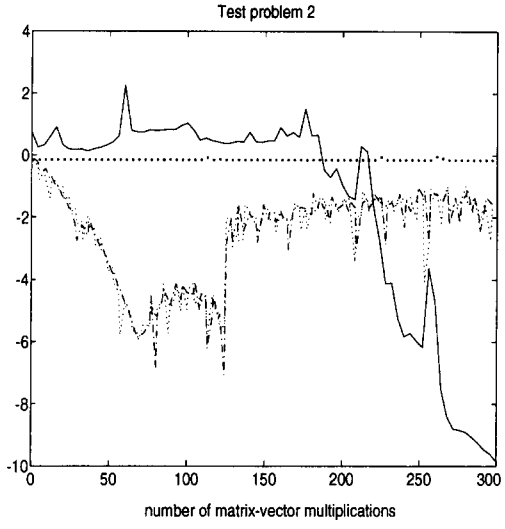

Fig. 9. Stabilized BiCGstab(2).

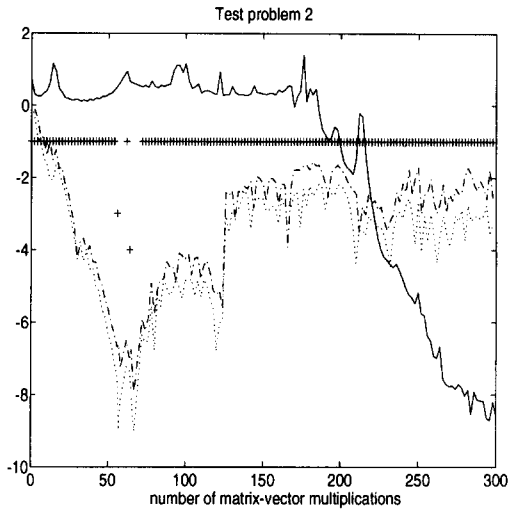

Fig. 10. Stab. BiCGstab($\{1:8\}$) using (24).

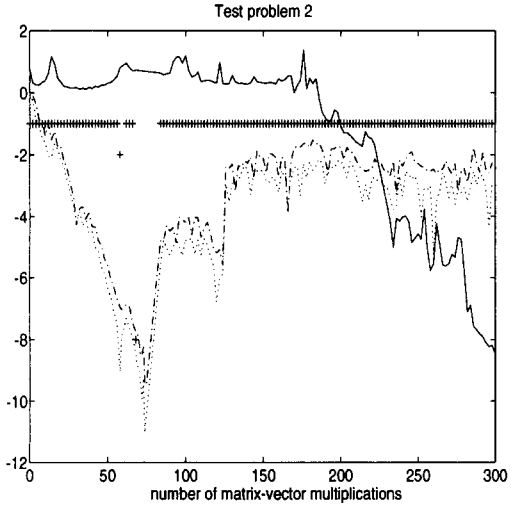

Fig. 11. Stab. BiCGstab($\{1:8\}$) using (25).

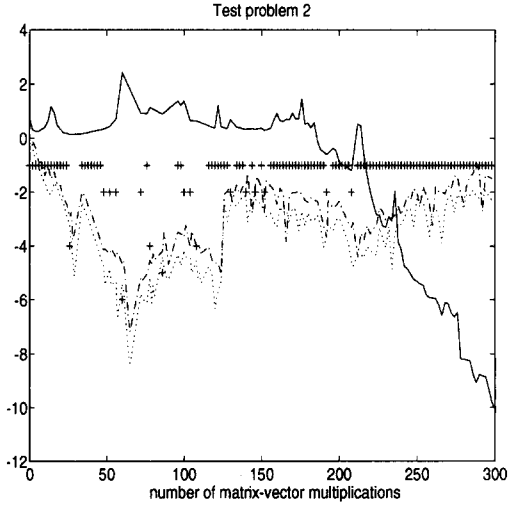

Fig. 12. Stab. BiCGstab($\{1:8\}$) using (26).

The results for the three test problems are shown in the Figs. 5–16 and will be discussed below.

Each of the criteria seems to prevent the values for $\widehat{\rho}_k$ and $\widehat{\sigma}_k$ to decrease (much) below δ. The accuracy in the BiCG coefficients appears to be enough to survive the stagnation phase. All criteria lead to convergence for the test problems considered here, although they did not all have the same speed of convergence. Recall that for convergence, it was not sufficient to stabilize Bi-CGSTAB, for the first and third test problem.

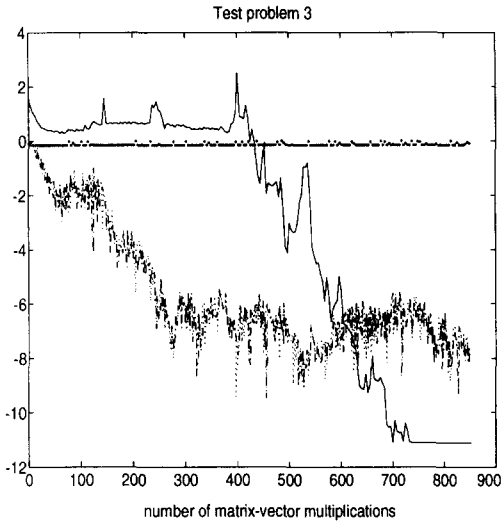

Fig. 13. Stabilized BiCGstab(2).

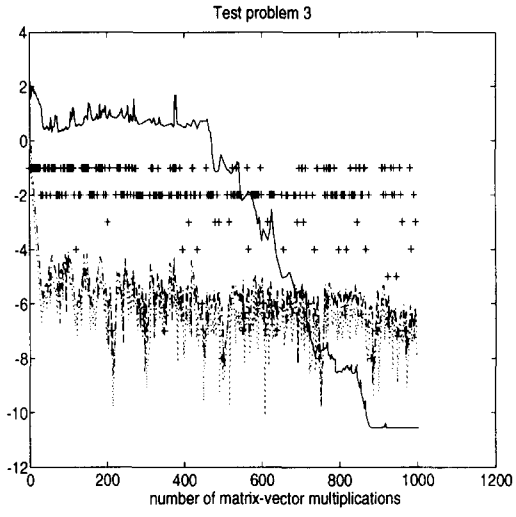

Fig. 14. Stab. BiCGstab($\{1:8\}$) using (24).

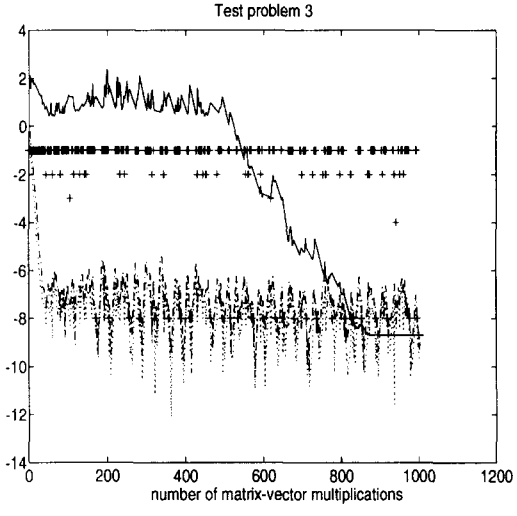

Fig. 15. Stab. BiCGstab($\{1:8\}$) using (25).

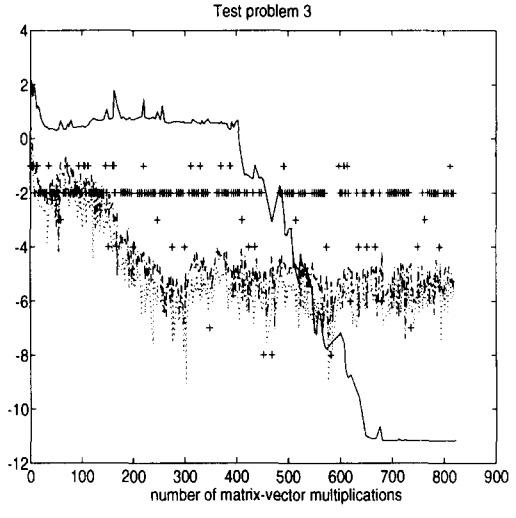

Fig. 16. Stab. BiCGstab($\{1:8\}$) using (26).

Although the criterion (24) is applied a posteriori, this seems to keep the values for $\widehat{\rho}_k$ larger than $\approx \delta$ (see Figs. 6, 10 and 14).

Observe that the minimal values for $\widehat{\sigma}_k$ are also not much smaller than δ. In general the $\widehat{\sigma}_k$ seem to be less than $\widehat{\rho}_k$. Therefore, criteria that try to prevent $\widehat{\sigma}_k$, rather than $\widehat{\rho}_k$, from becoming too small, may be more effective. In [13], in the context of look-ahead strategies for BiCG methods, a similar observation is formulated.

Criterion (25) seems to allow values for $\widehat{\rho}_k$ (and $\widehat{\sigma}_k$) that are somewhat smaller than δ, but still acceptable (see Figs. 7, 11 and 15). In all the test problems, the values for $\widehat{\rho}_k$ associated with criterion (25) are smaller than the ones associated with criterion (24). This seems to contradict our observation in the paragraph following (25). However, in the discussion there, we assumed $p^{(m)}$ to be close to the OR polynomial, as will be the case if MR (=GMRES) reduces well at the lth step with respect to the $(l-1)$st one. For our test problems, MR does not make (much) progress, in the early stage of the process; the value of $\widehat{\omega}$ is (often) less than 0.7 and we will make stabilizing steps. This will lead to polynomials $p^{(m)}$ for which $\|p^{(m)}(A)\widehat{r}\|/|\omega|$ may be slightly larger than $\|A^l\widehat{r}\|$, resulting in over-estimates $\widehat{\rho}'_k$ for $\widehat{\rho}_k$.

The criteria (24) and (25) discussed above, allow the values for $\widehat{\rho}_k$ to decrease rapidly as long as they are larger than δ. This means that, already in an early stage of the process, the BiCG coefficients will be correct to only approximately 7 digits. Although such an accuracy appears to be sufficient for maintaining convergence of BiCG, it may slow down this convergence: for the test problems, the stabilized BiCGstab($\{1:8\}$) methods converge slower than stabilized BiCGstab(2) when using the criteria (24) and (25). Criterion (26) is designed to control the cumulative reduction of $\widehat{\rho}_k$ due to a consecutive number of small $\widehat{\omega}$, and it seems to work well (see Figs. 8, 12 and 16): this criterion leads to a slower decrease of $\widehat{\rho}_k$ than the other two. For the first and third test problem, this is rewarded by faster convergence. The improvements by more accurate BiCG coefficients is paid for by an l that is larger on average. Although, as a consequence, the steps are more expensive, less steps are needed and the overall efficiency is better. A small decrease in the number of iteration steps will already compensate for the small additionals costs due to larger l.

For real-valued problems for which the matrix A has conjugate eigenpairs with relatively large imaginary parts, the MR polynomial of odd degree ($l > 2$) may often be approximately equal to the MR polynomial of even degree $(l-1)$: in that case the odd degree polynomials will have a small leading coefficient. Since criterion (26) controls the size of the leading coefficient, the resulting l (the l of the sweep) will seldomly be odd when larger than 2. Figs. 8 and 12 (but also Fig. 16) nicely illustrate this observation.

The criteria (24) and (26) rarely need the maximum value $l_{\max} = 8$ for l. Criterion (25) is less efficient; it seems to select either minimal values ($l = 1$ or $l = 2$) or the maximum one ($l = 8$).

The criteria (24) and (25) seem to favor low values for l (as $l = 1$). However, for the first and the third test problem, when considering the speed of convergence (also for stabilized Bi-CGSTAB and stabilized BiCGstab(2)), the value $l = 2$ seems to be the optimal one; for the second test problem, $l = 1$ might be more appropriate. These values are precisely the ones that are favored by criterion (26).

The value $l = 4$ is occasionally selected by criterion (26) for the first test problem (Fig. 8), keeping $\widehat{\rho}_k$ larger than δ. The $\widehat{\rho}_k$ for stabilized BiCGstab(2) decreases below δ (Fig. 5), and we see a convergence that is smoother and faster in Fig. 8 than in Fig. 5. The improvement is not impressive since it takes place only at a final stage of the process.

Criterion (26) seems to be attractive for finding optimal values for l.

Figs. 5–16 confirm the importance of having BiCG coefficients that are locally rather accurate: the method with, on average, the most accurate coefficients exhibits the best convergence properties. We conclude that it is helpful to try to keep the $\widehat{\rho}_k$ as large as possible. Although criteria that are directly based on the size of $\widehat{\rho}_k$ may maintain convergence, they may also lead to less efficient computations than criteria that try, in addition, to prevent $\widehat{\rho}_k$ from decreasing too quickly.

10. Conclusions

For convergence of the BiCG part in hybrid BiCG methods, it is important to compute the BiCG coefficients accurately. The polynomial associated with the hybrid part can be selected to minimize locally the rounding errors in these coefficients.

The (dynamical) combination of two strategies for the improvement of the local accuracy seems to be attractive: (i) take a product of degree l polynomials with $l > 1$ ($l = 2$, 4 or 8) as in BiCGstab(l) rather than the product of degree 1 polynomials as in Bi-CGSTAB (cf. Section 6); (ii) try to avoid almost degenerated degree l polynomials by forming occasionally convex combinations of the polynomials associated with l-step MR residuals and l-step OR residuals (cf. Sections 5 and 7).

These approaches often lead to improved convergence and may help to overcome phases of stagnation. The additional computational costs per sweep involved in forming the convex combinations of MR and OR results are negligible. The additional costs associated with larger l are small relatively to the costs of the matrix–vector multiplications.

It is possible to change the value for l in each sweep. We have proposed criteria to identify, per sweep, a small l that leads to locally accurate BiCG coefficients. Especially criterion (26) seems to find (relatively inexpensively) the most appropriate values for l. However, the additional improvements by varying l are small as compared with the improvements by following strategies (i) with fixed $l > 1$ and (ii).

References

[1] R.E. Bank and T.F. Chan, An analysis of the composite step biconjugate gradient method, *Numer. Math.* 66 (1993) 259–319.

[2] C. Brezinski and M. Redivo-Zaglia, Treatment of near breakdown in the CGS algorithm, *Numer. Algorithms* 7 (1994) 33–74.

[3] P.N. Brown, A theoretical comparison of the Arnoldi and GMRES algorithms, *SIAM J. Sci. Statist. Comput.* 12 (1991) 58–78.

[4] T.F. Chan, E. Gallopoulos, V. Simoncini, T. Szeto, and C.H. Tong, A quasi-minimal residual variant of the BI-CGSTAB algorithm for nonsymmetric systems, *SIAM J. Sci. Comput.* 15 (1994) 338–347.

[5] T.F. Chan and T. Szeto, A composite step conjugate gradient squared algorithm for solving nonsymmetric linear systems, *Numer. Algorithms* 7 (1994) 12–32.

[6] S.C. Eisenstat, H.C. Elman and M.H. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, *SIAM J. Numer. Anal.* 20 (1983) 345–357.

[7] R. Fletcher, Conjugate gradient methods for indefinite systems, in: *Lecture Notes in Mathematics* 506 (Springer, Berlin, 1976) 73–89.

[8] D.R. Fokkema, G.L.G. Sleijpen and H.A. van der Vorst, Generalized conjugate gradient squared, Preprint 851, Department of Mathematics, University of Utrecht, Utrecht (1994); also: *J. Comput. Appl. Math.* (to appear).

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

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

[11] A. Greenbaum, Behavior of slightly perturbed Lanczos and conjugate gradient recurrences, *Linear Algebra Appl.* 113 (1989) 7–63.

[12] M.H. Gutknecht, Variants of BiCGStab for matrices with complex spectrum, *SIAM J. Sci. Comput.* 14 (1993) 1020–1033.

[13] W. Joubert, Generalized conjugate gradient and Lanczos methods for the solution of nonsymmetric systems of linear equations, Ph.D. Thesis, Center for Numerical Analysis, University of Austin, Austin, TX (1990).
[14] K.J. Neylon, Numerical experiments with Bi-CGSTAB on an advection-diffusion problem, Numerical Analysis Report 8/93, Department of Mathematics, University of Reading, Reading, MA (1993).
[15] B.N. Parlett, D.R. Taylor, and Z.A. Liu, A look-ahead Lanczos algorithm for unsymmetric matrices, *Math. Comput.* 44 (1985) 105–124.
[16] C. Pommerell, *Solution of Large Unsymmetric Systems of Linear Equations*, Series in Micro Electronics 17 (Hartung-Gorre Verlag, Konstanz, 1992).
[17] Y. Saad, Krylov subspace methods for solving large unsymmetric linear systems, *Math. Comput.* 37 (1982) 105–126.
[18] Y. Saad and M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* 7 (1986) 856–869.
[19] G.L.G. Sleijpen and D.R. Fokkema, BiCGstab(l) for linear equations involving matrices with complex spectrum, *Electron. Trans. Numer. Anal.* 1 (1993) 11–32.
[20] G.L.G. Sleijpen and H.A. Van der Vorst, Maintaining convergence properties of BiCGstab methods in finite precision arithmetic, *Numer. Algorithms* 10 (1995) 203–223.
[21] G.L.G. Sleijpen and H.A. Van der Vorst, Reliable updated residuals in hybrid Bi-CG methods, Preprint 886, Department of Mathematics, University Utrecht, Utrecht (1994); also: *Computing* (to appear).
[22] G.L.G. Sleijpen, H.A. van der Vorst and D.R. Fokkema, BiCGstab(l) and other hybrid Bi-CG methods, *Numer. Algorithms* 7 (1994) 75–109.
[23] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* 10 (1989) 36–52.
[24] C. Tong, A comparative study of preconditioned Lanczos methods for nonsymmetric linear systems, Tech. Rept. SAND91-8240, Sandia National Laboratory, Livermore (1992).
[25] 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) 631–644.
[26] H.A. van der Vorst and C. Vuik, The superlinear convergence behaviour of GMRES, *J. Comput. Appl. Math.* 48 (1993) 327–341.