SIAM J. SCI. COMPUT.
Vol. 19, No. 1, pp. 55–73, January 1998

005

QMR SMOOTHING FOR LANCZOS-TYPE PRODUCT METHODS BASED ON THREE-TERM RECURRENCES*

KLAUS J. RESSEL†‡ AND MARTIN H. GUTKNECHT†

Abstract. For the solution of large, sparse, non-Hermitian linear systems, Lanczos-type product methods that are based on the Lanczos three-term recurrence are derived in a systematic way. These methods either square the Lanczos process or combine it with a local minimization of the residual. For them a quasi-minimal residual (QMR) smoothing is proposed that can also be implemented by short-term recurrences. The practical performance of these methods and the QMR smoothing is demonstrated in a number of numerical examples.

Key words. Lanczos-type product methods, iterative methods, QMR smoothing, non-Hermitian matrices, sparse linear systems

AMS subject classifications. 65F15, 65F10

PII. S1064827596304812

1. Introduction.

Many problems in scientific computing lead to the task of solving a system of linear equations $Ax = b$ with a large, sparse, non-Hermitian system matrix $A \in \mathbb{C}^{N\times N}$. In general, optimal Krylov subspace methods that obtain the solution iteratively by minimizing the residual over a Krylov subspace cannot be based on a short-term recurrence [10], so their work and storage requirements grow linearly with the iteration number. On the contrary, methods that are related to the Lanczos bi-orthogonalization (BiO) process, such as the Lanczos method [20] and the bi-conjugate gradient (BiCG) method [11], [21], are based on a three-term recurrence or coupled two-term recurrences. For this reason, they can be implemented with low and roughly constant work and storage requirements. However, they lack a minimization property over the generated Krylov subspace in a standard fixed norm. Instead, they minimize the error in a variable norm [2], which depends on the initial iterate and is not known explicitly at the beginning of the iteration. Therefore, with respect to the Euclidean norm they typically have a rather irregular convergence behavior with wild oscillations of the residual.

A monotone decreasing residual norm history can be achieved by postprocessing the Lanczos (BiCG) iterates and residuals with the local *minimal residual smoothing* process, proposed by Schönauer [27] and further investigated in [32], [33], [16], [35].

Smooth convergence curves result also from the combination of the Lanczos process with the *quasi-minimal residual* (QMR) smoothing technique, developed by Freund and Nachtigal [15]. Instead of defining the approximate iterates by a Galerkin condition (as in BiCG), in QMR they are obtained by a global quasi-minimization of the residual. In exact arithmetic a general relationship between the residuals of a Galerkin process (like FOM-Arnoldi, BiCG) and a norm-minimizing process (like GMRES, QMR) can be derived [5], [9]. This shows that when the norm-minimizing process is converging rapidly, the residual norm of the corresponding Galerkin process

*Received by the editors June 4, 1996; accepted for publication (in revised form) January 3, 1997. This work was supported by Swiss National Science Foundation grant 20-40737.94.
http://www.siam.org/journals/sisc/19-1/30481.html
†Swiss Center for Scientific Computing (SCSC), ETH Zürich, ETH-Zentrum, CH-8092 Zürich, Switzerland (kjr@scsc.ethz.ch, mhg@scsc.ethz.ch).
‡Current address: German Remote Sensing Data Center (DFD), German Aerospace Center (DLR), D-82234 Oberpfaffenhofen, Germany (kjr@dfd.dlr.de).

exhibits a similar behavior, whereas when the convergence of the norm-minimizing process is stagnating, the residual norm in the corresponding Galerkin process can even grow significantly. Whenever such a peak occurs in the convergence curve of a Galerkin process, there is a plateau in the convergence curve of the corresponding norm-minimizing process beneath it [8]. Thus, (quasi-)minimizing the residual norm cannot accelerate the convergence significantly. However, it provides us with a smoother residual norm plot compared to the one for the corresponding Galerkin method.

An improvement of the Lanczos process in another direction is the development of Lanczos-type product methods (LTPMs), like CGS [29], BiCGStab [31], BiCGStab2 [17], BiCGStab(ℓ) [28], and others [4]. These methods either square the Lanczos polynomial or they combine it with another (*shadow*) polynomial, e.g., for achieving a local minimization of the residual. They have the advantage of converging roughly twice as fast as the plain Lanczos process, and further, they do not require a routine for applying the adjoint system matrix A^H to a vector. For an overview of LTPMs that are based on two-term recurrences for both the Lanczos and the shadow polynomials, we refer to [12] and [34].

LTPMs that are based on the coupled two-term version of the Lanczos process (BiCG) are successfully combined with the QMR approach in the transpose-free QMR (TFQMR) method [13], the QMRCGSTAB method [7], the QMRCGSTAB(k) method [30], and Cao's QMR approach [6]. Nevertheless, these methods inherit from the underlying Lanczos process the danger of breakdown. Although exact breakdowns are very rare in practice, it has been observed that near-breakdowns can slow down or even prevent convergence [14]. Look-ahead techniques for the plain Lanczos process [14], [18], [24] allow us to avoid this problem with very small overhead. However, the look-ahead procedures that have so far been proposed for CGS or other LTPMs either are limited to exact breakdowns or are complicated and have considerable overhead [3], [4]. We have recently developed more efficient look-ahead procedures for LTPMs. This new approach is first applied to LTPMs that are based on the Lanczos three-term recurrences [19], since look-ahead procedures for the three-term version of the Lanczos process are less complicated than those for the coupled two-term recurrences.

In the following work we consider, therefore, LTPMs that are based on the Lanczos three-term recurrence and develop a QMR technique for them. The remainder of this paper is organized as follows. In section 2 we systematically derive different LTPMs from the Lanczos three-term recurrences by introducing the notation of product vectors and of the w-table. Based on a new relationship of these product vectors and their product with the system matrix we develop a QMR version of these LTPMs in section 3. Numerical experiments in section 4 are used to compare the performance of different LTPMs and to demonstrate the smoothing effect of the QMR approach, even for LTPMs that combine the Lanczos process with a local minimization of the residual. Finally, in section 5 we draw some conclusions.

Throughout the paper we use the following notation. If $\alpha \in \mathbb{C}$, then $\overline{\alpha}$ denotes its complex conjugate. For $x, y \in \mathbb{C}^N$, $\langle x, y\rangle := x^H y$ is the standard inner product of $\mathbb{C}^N$ and $\|x\| := \sqrt{\langle x, x\rangle}$ is the corresponding norm.

2. LTPMs. In this section we derive LTPMs that are based on the Lanczos three-term recurrences. We start with the one-sided Lanczos BiO process, from which we can then obtain different LTPMs.

2.1. The one-sided Lanczos BiO process. We recall that the Lanczos BiO algorithm generates a pair of finite sequences, $\{\widetilde{y}_n\}_{n=0}^{\nu}$, $\{y_n\}_{n=0}^{\nu}$, of *left* and *right*

Lanczos vectors, such that

$$\begin{aligned} y_n \in \mathcal{K}_{n+1} &:= \text{span}\,\{y_0, Ay_0, \ldots, A^n y_0\}, \\ \widetilde{y}_n \in \mathcal{L}_{n+1} &:= \text{span}\,\{\widetilde{y}_0, A^H\widetilde{y}_0, \ldots, (A^H)^n\widetilde{y}_0\}, \end{aligned} \tag{2.1}$$

and

$$\widetilde{y}_n \perp \mathcal{K}_n, \quad y_n \perp \mathcal{L}_n. \tag{2.2}$$

This sequence of pairs of Lanczos vectors can be constructed by the following three-term recursion:

$$\begin{aligned} y_{n+1} &= \left(Ay_n - y_n\alpha_n - y_{n-1}\beta_n\right)/\gamma_n, \\ \widetilde{y}_{n+1} &= \left(A^H\widetilde{y}_n - \widetilde{y}_n\overline{\alpha}_n - \widetilde{y}_{n-1}\overline{\beta}_n\right)/\overline{\gamma}_n, \end{aligned} \tag{2.3}$$

with coefficients α_n and β_n that are determined from the orthogonality condition (2.2) and freely choosable nonvanishing scale factors γ_n. The Lanczos vectors can then be written in the form

$$y_n = \rho_n(A)y_0, \qquad \widetilde{y}_n = \overline{\rho}_n(A^H)\widetilde{y}_0, \tag{2.4}$$

where ρ_n denotes the nth Lanczos polynomial. Since we aim here at LTPMs, we consider for the Krylov spaces $\mathcal{L}_n$ more general basis vectors of the form

$$\widetilde{z}_n = \overline{\tau}_n(A^H)\widetilde{z}_0, \tag{2.5}$$

with arbitrarily chosen polynomials τ_n of exact degree n and $\widetilde{z}_0 := \widetilde{y}_0$. In general, $\widetilde{z}_n \perp \mathcal{K}_n$ will no longer hold, but the one-sided bi-orthogonality $y_n \perp \mathcal{L}_n$ can still be attained by enforcing $\widetilde{z}_{n-1} \perp y_{n+1}$ and $\widetilde{z}_n \perp y_{n+1}$, which means choosing the coefficients α_n and β_n in (2.4) in the following way. If $n > 0$ we need

$$0 = \langle \widetilde{z}_{n-1}, y_{n+1}\rangle = \left(\langle \widetilde{z}_{n-1}, Ay_n\rangle - \langle \widetilde{z}_{n-1}, y_n\rangle\,\alpha_n - \langle \widetilde{z}_{n-1}, y_{n-1}\rangle\,\beta_n\right)/\gamma_n,$$

but since $\langle \widetilde{z}_{n-1}, y_n\rangle = 0$ we have

$$\beta_n = \frac{\langle \widetilde{z}_{n-1}, Ay_n\rangle}{\langle \widetilde{z}_{n-1}, y_{n-1}\rangle}, \tag{2.6}$$

and in the case $n = 0$ we set $\beta_0 = 0$. Similarly, from the orthogonality condition $\langle \widetilde{z}_n, y_{n+1}\rangle = 0$ we obtain

$$\alpha_n = \frac{\langle \widetilde{z}_n, Ay_n\rangle - \langle \widetilde{z}_n, y_{n-1}\rangle\,\beta_n}{\langle \widetilde{z}_n, y_n\rangle}. \tag{2.7}$$

Clearly, the recursive process terminates with $y_\nu = 0$ or $\widetilde{z}_\nu = 0$, or it breaks down with $\delta_\nu^\tau := \langle \widetilde{z}_\nu, y_\nu\rangle = 0$ and $y_\nu \neq 0$, $\widetilde{z}_\nu \neq 0$. The look-ahead Lanczos process [14], [18], [24] overcomes such a breakdown if curable. To simplify our presentation, we assume in the following that no breakdown occurs in the underlying Lanczos process. For a description of look-ahead procedures for LTPMs we refer the reader to [19].

2.2. Product methods. LTPMs originate from the ingenious idea, due to Sonneveld [29] and van der Vorst [31], to rewrite the inner products in the definition (2.7) and (2.6) of the coefficients α_n and β_n in the following way:

$$\langle \widetilde{z}_l, y_n \rangle = \langle \overline{\tau}_l(A^H)\widetilde{z}_0, y_n \rangle = \langle \widetilde{z}_0, \tau_l(A) y_n \rangle = \langle \widetilde{z}_0, w_n^l \rangle ,$$
$$\langle \widetilde{z}_l, Ay_n \rangle = \langle \overline{\tau}_l(A^H)\widetilde{z}_0, Ay_n \rangle = \langle \widetilde{z}_0, A\tau_l(A) y_n \rangle = \langle \widetilde{z}_0, Aw_n^l \rangle ,$$

and then to formulate the Lanczos process in terms of the new *product vectors*

$$w_n^l := \tau_l(A) y_n := \tau_l(A)\rho_n(A) y_0, \tag{2.8}$$

without explicitly computing the Lanczos vectors z_n and y_n. This has the advantage that, first, multiplications with the adjoint system matrix A^H are avoided and, second, for an appropriate choice of the polynomials $\tau_l(\zeta)$, smaller new residuals (see below) $r_l := w_l^l = \tau_l(A) y_l$ can be expected because of a further reduction of y_l by the operator $\tau_l(A)$.

Expressing the Lanczos coefficients α_n and β_n from (2.7) and (2.6) in terms of product vectors w_n^l results in

$$\beta_n = \frac{\langle \widetilde{z}_0, Aw_n^{n-1} \rangle}{\langle \widetilde{z}_0, w_{n-1}^{n-1} \rangle}, \qquad \alpha_n = \frac{\langle \widetilde{z}_0, Aw_n^n \rangle - \langle \widetilde{z}_0, w_{n-1}^n \rangle \beta_n}{\langle \widetilde{z}_0, w_n^n \rangle}. \tag{2.9}$$

It remains to obtain recursion formulas for the product vectors w_n^l. Let us arrange the product vectors in a *w-table*, where the n-axis of the table points downwards and the l-axis to the right. The aim of every LTPM is then to move in this table from the upper left corner downwards right (see Figures 2.1 and 2.2). By multiplying the three-term recursion (2.3) for the right Lanczos vector y_{n+1} with $\tau_l(A)$ we obtain for the product vectors the recursion

$$w_{n+1}^l = \left(Aw_n^l - w_n^l \alpha_n - w_{n-1}^l \beta_n \right) / \gamma_n, \tag{2.10}$$

which can be applied to move forward in the vertical direction in the w-table. To obtain a recursion formula for moving forward in the horizontal direction, the recursion formula for the chosen polynomials $\tau_l(\zeta)$ is applied. For example, let us consider polynomials $\tau_l(\zeta)$ that are generated by a *consistent*[1] three-term recurrence of the form

$$\tau_{l+1}(\zeta) = (\xi_l + \eta_l \zeta)\, \tau_l(\zeta) + (1 - \xi_l)\, \tau_{l-1}(\zeta), \tag{2.11}$$

with $\tau_{-1}(\zeta) \equiv 0$, $\tau_0(\zeta) \equiv 1$, and $\xi_0 = 1$. This covers all LTPMs that are considered in this paper except for bi-orthogonalization squared (BiOS), where the Lanczos polynomial is squared. Examples of how to choose the coefficients η_l and ξ_l are given below. Inserting in (2.11) for ζ the system matrix A and multiplying by y_n from the right leads to the recursion

$$w_n^{l+1} = Aw_n^l \eta_l + w_n^l \xi_l + w_n^{l-1}(1 - \xi_l). \tag{2.12}$$

Finally, since the overall aim is to obtain the solution of $Ax = b$, we need to define approximate solution iterates x_n^l. For methods that are based on the Lanczos process,

[1] A recursion for generating a Krylov space basis will be called *consistent* if the associated polynomials all take the value 1 at $\zeta = 0$. In the literature, the term *normalized* has also been used for this property.

we can define iterates in two different ways: the first is by a quasiminimization of the residual over the generated Krylov subspace, which will be considered in section 3. The second is by imposing a Galerkin condition on the residual, which is discussed now. If we choose $\gamma_n := -\alpha_n - \beta_n$, the Lanczos polynomial is also consistent (i.e., $\rho_n(0) = 1$), so that the product $\tau_l(\xi)\rho_n(\xi)$ is a residual polynomial. Thus the product vectors $w_n^l = \tau_l(A)\rho_n(A)y_0$ can play the role of residuals, and therefore we could define the iterates x_n^l in the following way. Starting with an arbitrary initial product iterate $x_0^0 \in \mathbb{C}^N$, we assume that the initial product vector is chosen so that $w_0^0 = b - Ax_0^0$, and for $n, l > 0$ we define the product iterates by the relation $Ax_n^l - b := w_n^l$.

A numerically more stable option is to generalize the idea of inconsistent BiORes, called unnormalized BiORes in [18]. There, the scaling parameters γ_n are used to normalize the Lanczos vectors y_n in order to avoid underflow or overflow. Since the Lanczos vectors y_n are not explicitly computed in an LTPM, we cannot base the choice of γ_n here on the norm of y_n. However, in every LTPM, it is necessary to compute the product vectors w_{n+1}^n (see Figures 2.1 and 2.2). Therefore, we choose here γ_n to normalize the product vector w_{n+1}^n, i.e.,

$$\gamma_n := \left\| Aw_n^n - w_n^n\alpha_n - w_{n-1}^n\beta_n \right\|. \tag{2.13}$$

Then, in general, the Lanczos polynomials $\rho_n(\zeta)$ are no longer consistent. For this reason we introduce the scalars

$$\hat{\rho}_n^l := \tau_l(0)\rho_n(0), \tag{2.14}$$

which can be used to make the Lanczos polynomials implicitly consistent. For an arbitrary initial iterate $x_0^0 \in \mathbb{C}^N$ we define the first product vector w_0^0 as the corresponding residual, i.e., $w_0^0 := b - Ax_0^0 = b\hat{\rho}_0^0 - Ax_0^0$, using the fact that $\tau_0(0) = \rho_0(0) = 1 = \hat{\rho}_0^0$. For $l, n > 0$, $\tau_l(\zeta)\rho_n(\zeta)/\hat{\rho}_n^l$ is consistent, so that $w_n^l/\hat{\rho}_n^l = \tau_l(A)\rho_n(A)y_0/\hat{\rho}_n^l$ can be interpreted as a residual corresponding to an approximation $x_n^l/\hat{\rho}_n^l$ for the solution of $Ax = b$. Therefore, we define the iterates x_n^l and the scalars $\hat{\rho}_n^l$ by the following relation:

$$b - A\left(x_n^l/\hat{\rho}_n^l\right) = w_n^l/\hat{\rho}_n^l \Longleftrightarrow b\hat{\rho}_n^l - Ax_n^l = w_n^l. \tag{2.15}$$

From (2.15) and the recursion for the product vectors w_n^l we can now obtain recursions for the product iterates x_n^l and the scalars $\hat{\rho}_n^l$ in the following way. By (2.10) and (2.15) we have

$$\begin{aligned} b\hat{\rho}_{n+1}^l - Ax_{n+1}^l &= w_{n+1}^l = \left(Aw_n^l - w_n^l\alpha_n - w_{n-1}^l\beta_n\right)/\gamma_n \\ &= b\left[-\hat{\rho}_n^l\alpha_n - \hat{\rho}_{n-1}^l\beta_n\right]/\gamma_n + A\left[w_n^l + x_n^l\alpha_n + x_{n-1}^l\beta_n\right]/\gamma_n. \end{aligned}$$

Thus, it follows that

$$\begin{aligned} x_{n+1}^l &= -\left[w_n^l + x_n^l\alpha_n + x_{n-1}^l\beta_n\right]/\gamma_n, \\ \hat{\rho}_{n+1}^l &= -\left[\hat{\rho}_n^l\alpha_n + \hat{\rho}_{n-1}^l\beta_n\right]/\gamma_n. \end{aligned} \tag{2.16}$$

Similarly, we obtain from (2.12) and (2.15)

$$\begin{aligned} x_n^{l+1} &= -w_n^l\eta_l + x_n^l\xi_l + x_n^{l-1}(1-\xi_l), \\ \hat{\rho}_n^{l+1} &= \hat{\rho}_n^l\xi_l + \hat{\rho}_n^{l-1}(1-\xi_l). \end{aligned} \tag{2.17}$$

The horizontal update for the scalars $\hat{\rho}_n^l$ in (2.17) simplifies to $\hat{\rho}_n^{l+1} = \hat{\rho}_n^l$ if the polynomials $\tau_l(\zeta)$ are consistent, since then $\hat{\rho}_n^l = \hat{\rho}_n^0 = \rho_n(0)$ for all $n, l \geq 0$.

Altogether, generating the polynomials τ_l by a general, consistent three-term recurrence (2.11) leads to the general LTPM BiOxThree, which is described by the following algorithm.

ALGORITHM 2.1 (BiOxThree).

0) Choose $x_0^0, \widetilde{z}_0 \in \mathbb{C}^N$; $\quad w_0^0 = b - Ax_0^0$; $\quad \delta_0 = \langle \widetilde{z}_0, w_0^0 \rangle$; $\quad \hat{\rho}_0^0 = 1$;
1) For $n = 0 : n_{\max}$ Do:
2) If $n = 0$ Then
3) $\beta_n = 0$; $\alpha_n = \langle \widetilde{z}_0, Aw_n^n \rangle / \delta_n$; $w_{n+1}^n = Aw_n^n - w_n^n \alpha_n$; $x_{n-1}^n = 0$; $\hat{\rho}_{n-1}^n = 0$;
4) Else
5) $\beta_n = \langle \widetilde{z}_0, Aw_n^{n-1} \rangle / \delta_{n-1}$; $\quad \alpha_n = (\langle \widetilde{z}_0, Aw_n^n \rangle - \langle \widetilde{z}_0, w_{n-1}^n \rangle \beta_n) / \delta_n$;
6) $w_{n+1}^n = Aw_n^n - w_n^n \alpha_n - w_{n-1}^n \beta_n$;
7) EndIf;
8) $\gamma_n = \|w_{n+1}^n\|$; $\quad w_{n+1}^n = w_{n+1}^n / \gamma_n$; $\quad x_{n+1}^n = -(w_n^n + x_n^n \alpha_n + x_{n-1}^n \beta_n) / \gamma_n$;
9) $\hat{\rho}_{n+1}^n = -(\hat{\rho}_n^n \alpha_n + \hat{\rho}_{n-1}^n \beta_n) / \gamma_n$; $\quad \delta_{n+1} = \langle \widetilde{z}_0, w_{n+1}^{n+1} \rangle$; $\quad$ If $\delta_{n+1} < \varepsilon$ Then Stop;
10) If $n > 0$ Then
11) $w_{n+1}^{n-1} = \left(Aw_n^{n-1} - w_n^{n-1} \alpha_n - w_{n-1}^{n-1} \beta_n\right) / \gamma_n$;
12) $x_{n+1}^{n-1} = -\left(w_n^{n-1} + x_n^{n-1} \alpha_n + x_{n-1}^{n-1} \beta_n\right) / \gamma_n$; $\hat{\rho}_{n+1}^{n-1} = -\left(\hat{\rho}_n^{n-1} \alpha_n + \hat{\rho}_{n-1}^{n-1} \beta_n\right) / \gamma_n$;
13) EndIf;
14) Compute η_n, ξ_n;
15) $w_{n+1}^{n+1} = Aw_{n+1}^n \eta_n + w_{n+1}^n \xi_n + w_{n+1}^{n-1}(1 - \xi_n)$;
16) $x_{n+1}^{n+1} = -w_{n+1}^n \eta_n + x_{n+1}^n \xi_n + x_{n+1}^{n-1}(1 - \xi_n)$; $\quad \hat{\rho}_{n+1}^{n+1} = \hat{\rho}_{n+1}^n$;
17) $w_n^{n+1} = Aw_n^n \eta_n + w_n^n \xi_n + w_n^{n-1}(1 - \xi_n)$;
18) $x_n^{n+1} = -w_n^n \eta_n + x_n^n \xi_n + x_n^{n-1}(1 - \xi_n)$; $\quad \hat{\rho}_n^{n+1} = \hat{\rho}_n^n$;
19) If $\left\|w_{n+1}^{n+1} / \hat{\rho}_{n+1}^{n+1}\right\| \leq$ tol Then Stop;
20) EndFor;

Different choices for η_n and ξ_n, or more general for $\tau_n(\zeta)$, lead to different LTPMs. In the following we briefly discuss some possible choices of these polynomials, ordered by the length of the recurrence by which they are formed.

2.2.1. BiOStab. As in van der Vorst's BiCGStab [31], in BiOStab the polynomials $\tau_l(\zeta)$ are chosen as a product of polynomials of degree 1:

$$\tau_0(\zeta) \equiv 1, \qquad \tau_{l+1}(\zeta) = (1 + \eta_l \zeta)\tau_l(\zeta) \quad \text{for } l \geq 0, \tag{2.18}$$

where η_l is defined by minimizing the norm of $w_{l+1}^{l+1} = w_{l+1}^l + \eta_l A w_{l+1}^l$. Thus

$$\left\|w_{l+1}^l + \eta_l A w_{l+1}^l\right\| = \min_{\eta \in \mathbb{C}} \left\|w_{l+1}^l + \eta A w_{l+1}^l\right\|, \tag{2.19}$$

which leads to

$$\eta_l = -\frac{\langle A w_{l+1}^l, w_{l+1}^l \rangle}{\langle A w_{l+1}^l, A w_{l+1}^l \rangle}. \tag{2.20}$$

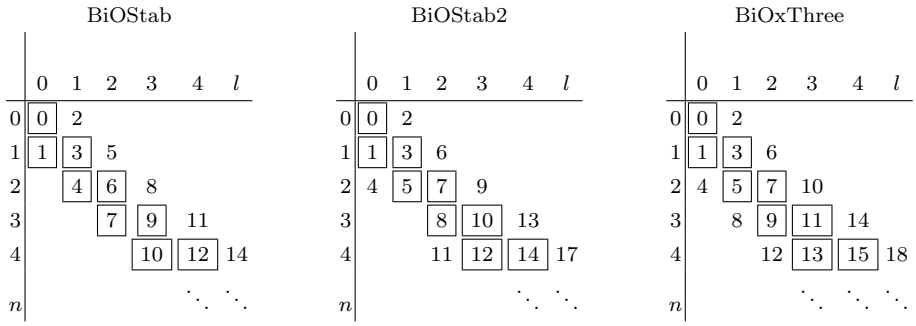

FIG. 2.1. *Movement of BiOStab, BiOStab2, and BiOxThree (an algorithm using the general three-term recurrence* (2.11) *for* τ_l*) in the w-table. The numbers illustrate in which sequence the entries are computed. A framed entry indicates that the corresponding entry as well as its product with the system matrix A is computed.*

To obtain an algorithmic description of BiOStab, we just need to leave out steps 10) to 13) in Algorithm 2.1, compute η_n according to (2.20), and set $\xi_n = 1$ for all $n \geq 0$. The product vectors w_n^l that are computed in BiOStab are shown in Figure 2.1.

2.2.2. BiOStab2. BiOStab2 is the three-term version of BiCGStab2 [17], where the polynomials $\tau_l(\zeta)$ are defined as

$$\begin{array}{lcll} \tau_0(\zeta) & \equiv & 1, & \\ \tau_{l+1}(\zeta) & = & (1+\eta_l\zeta)\,\tau_l(\zeta) & \text{if } l \text{ is even,} \\ \tau_{l+1}(\zeta) & = & (\xi_l+\eta_l\zeta)\,\tau_l(\zeta) + (1-\xi_l)\,\tau_{l-1}(\zeta) & \text{if } l \text{ is odd.} \end{array} \tag{2.21}$$

In the odd steps (where l is even) η_l may be obtained by solving the one-dimensional minimization problem (2.19), so η_l is given by (2.20). If $|\eta_l|$ is small, this choice is bad, since we do not advance in the Krylov space [28]. Then some other value should be chosen. Except for roundoff, the choice has no effect on later steps, because in the even steps (where l is odd) ξ_l and η_l are determined by solving the two-dimensional minimization problem:

$$\begin{aligned} &\left\| w_{l+1}^{l-1} + \left(w_{l+1}^l - w_{l+1}^{l-1}\right)\xi_l + A w_{l+1}^l \eta_l \right\| \\ &\qquad = \min_{\xi,\eta\in\mathbb{C}} \left\| w_{l+1}^{l-1} + \left(w_{l+1}^l - w_{l+1}^{l-1}\right)\xi + A w_{l+1}^l \eta \right\|. \end{aligned} \tag{2.22}$$

Introducing the matrix $B_{l+1} := \left[\ \left(w_{l+1}^l - w_{l+1}^{l-1}\right) \mid A w_{l+1}^l\ \right]$, we can write (2.22) as the least-squares problem

$$\left\| w_{l+1}^{l-1} + B_{l+1} \begin{bmatrix} \xi_l \\ \eta_l \end{bmatrix} \right\| = \min_{\xi,\eta\in\mathbb{C}} \left\| w_{l+1}^{l-1} + B_{l+1} \begin{bmatrix} \xi \\ \eta \end{bmatrix} \right\|.$$

Therefore, ξ_l and η_l can be computed as the solution of the corresponding normal equations:

$$\begin{bmatrix} \xi_l \\ \eta_l \end{bmatrix} = -\left(B_{l+1}^H B_{l+1}\right)^{-1} B_{l+1}^H w_{l+1}^{l-1}. \tag{2.23}$$

For the algorithmic description of BiOStab2, we only need to leave out steps 10) to 13) in Algorithm 2.1 if n is even. In this case we use $\xi_n = 1$, and η_n can be chosen freely, for example, according to (2.20) if $|\eta_l|$ is not too small. For n odd, η_n and ξ_n are computed according to (2.23). The w-table for BiOStab2 is also depicted in Figure 2.1.

2.2.3. BiOxThree (BiOxMR2, BiOxCheb). These LTPMs use polynomials $\tau_l(\zeta)$ that are generated by the consistent three-term recursion (2.11) in every step after the first initialization step. The BiOxMR2 method, which was introduced by the second author in a talk in Oberwolfach in April 1994 and was also independently found by Cao [6] and, in an equivalent form, by Zhang [34], determines the coefficients η_n and ξ_n in every step by the solution of the two-dimensional minimization problem (2.22) except for the very first step, where the one-dimensional minimization in (2.19) is performed. Thus this method consists of a product of the BiO process with one for a two-dimensional minimization of the residual (MR2), which is indicated by the name BiOxMR2.

Moreover, shifted and scaled Chebyshev polynomials can be generated by a recurrence like (2.11). After acquiring some information about the spectrum of the system matrix A, for example, by performing some BiOxMR2 iterations, one can then also use as polynomials $\tau_n(\zeta)$ the scaled and shifted Chebyshev polynomials that correspond to an ellipse surrounding the estimated spectrum [22], [23]. This leads to a combination of the Lanczos process with the Chebyshev method, which we name BiOxCheb [25].

Algorithm 2.1 describes both methods if in step 14) the computation of η_n and ξ_n is adapted as outlined above. The product vectors that need to be computed are again shown in Figure 2.1.

2.2.4. BiOS. In BiOS the Lanczos polynomial is squared by the choice $\tau_l(\zeta) = \rho_l(\zeta)$. The vectors $\widetilde{z}_n = \overline{\tau}_n(A^H)\widetilde{z}_0 = \overline{\rho}_n(A^H)\widetilde{y}_0 = \widetilde{y}_n$ are then exactly the left Lanczos vectors, so that now $y_n \perp \mathcal{L}_n$ as well as $\widetilde{z}_n \perp \mathcal{K}_n$ is fulfilled. For this reason the computation of the coefficients α_n in (2.9) simplifies to

$$\alpha_n = \frac{\langle \widetilde{z}_0, Aw_n^l \rangle}{\langle \widetilde{z}_0, w_n^n \rangle}, \tag{2.24}$$

and the w-table becomes symmetric, i.e., $w_n^l = w_l^n$.

For an algorithmic description of BiOS we have to change step 6) in Algorithm 2.1 to

$$6') \qquad w_{n+1}^n = \left(Aw_n^n - w_n^n\alpha_n - w_n^{n-1}\beta_n\right)/\gamma_n,$$

where we exploit the symmetry of the w-table. Further, we have to replace steps 14) to 18) by

$$14') \qquad w_{n+1}^{n+1} = \left(Aw_{n+1}^n - w_{n+1}^n\alpha_n - w_{n+1}^{n-1}\beta_n\right)/\gamma_n,$$
$$15') \qquad x_{n+1}^{n+1} = -\left(w_{n+1}^n + x_{n+1}^n\alpha_n + x_{n+1}^{n-1}\beta_n\right)/\gamma_n,$$
$$16') \qquad \hat{\rho}_{n+1}^{n+1} = -\left(\hat{\rho}_{n+1}^n\alpha_n + \hat{\rho}_{n+1}^{n-1}\beta_n\right)/\gamma_n.$$

The movement of BiOS in the w-table is illustrated in Figure 2.2.

3. QMR solution approach.

3.1. Key ingredients. The QMR approach [15], [13] relies on a basis for the Krylov space $\mathcal{K}_m$ and its image under an application of the system matrix. Since in any LTPM the product vectors w_n^l on the staircase $\left\{w_n^l : l = n \vee l = n-1\right\}$ of the w-table are computed (compare Figure 2.1 and Figure 2.2), it is natural to use these product vectors as a basis of $\mathcal{K}_m$. Grouping them together as the columns of a matrix

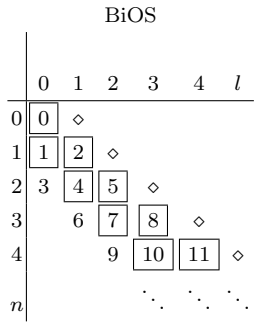

FIG. 2.2. *Movement of BiOS in the w-table. The numbers illustrate in which sequence the entries are computed. A framed entry indicates that the corresponding entry as well as its product with the system matrix A is computed, whereas entries marked by a diamond are needed in the recurrence, but because of the symmetry of the w-table in this case, they do not need to be computed.*

leads to

$$\begin{aligned} W_{2n+1} &:= \left[w_0^0, w_1^0, w_1^1, \ldots, w_n^{n-1}, w_n^n\right] \quad \text{for } n \geq 0, \\ W_{2n} &:= \left[w_0^0, w_1^0, w_1^1, \ldots, w_n^{n-1}\right] \quad \text{for } n \geq 1. \end{aligned} \tag{3.1}$$

Then the columns of W_m are a basis for $\mathcal{K}_m$. It remains to express the product of these basis vectors with the system matrix A as a linear combination of themselves.

Let us first consider the case where the polynomials are generated by the normalized three-term recurrence (2.11), so that the horizontal recurrence for the product vectors is given by (2.12). From (2.10) with $l = n$ we obtain

$$Aw_n^n = w_{n+1}^n \gamma_n + w_n^n \alpha_n + w_{n-1}^n \beta_n. \tag{3.2}$$

Note that the product vector w_{n-1}^n is not located on the staircase, but using the recurrence (2.12) for $n = n-1$ and $l = n-1$ leads to

$$w_{n-1}^n = Aw_{n-1}^{n-1}\eta_{n-1} + w_{n-1}^{n-1}\xi_{n-1} + w_{n-1}^{n-2}(1-\xi_{n-1}). \tag{3.3}$$

Inserting (3.3) into (3.2) results in

$$\begin{aligned} &A\left(w_n^n - w_{n-1}^{n-1}\beta_n\eta_{n-1}\right) \\ &\qquad = w_{n+1}^n\gamma_n + w_n^n\alpha_n + w_{n-1}^{n-1}\beta_n\xi_{n-1} + w_{n-1}^{n-2}\beta_n(1-\xi_{n-1}). \end{aligned} \tag{3.4}$$

On the other hand, from (2.12) for $n = n+1$ and $l = n$ we get

$$Aw_{n+1}^n = \left(w_{n+1}^{n+1} - w_{n+1}^n\xi_n - w_{n+1}^{n-1}(1-\xi_n)\right)/\eta_n. \tag{3.5}$$

Again, w_{n+1}^{n-1} is not located on the diagonal staircase, so we need to use (2.10) for $l = n-1$ to replace it by $(Aw_n^{n-1} - w_n^{n-1}\alpha_n - w_{n-1}^{n-1}\beta_n)/\gamma_n$, which yields

$$\begin{aligned} &A\left(w_{n+1}^n + w_n^{n-1}\frac{(1-\xi_n)}{\eta_n\gamma_n}\right) \\ &\qquad = w_{n+1}^{n+1}\frac{1}{\eta_n} - w_{n+1}^n\frac{\xi_n}{\eta_n} + w_n^{n-1}\frac{\alpha_n(1-\xi_n)}{\eta_n\gamma_n} + w_{n-1}^{n-1}\frac{\beta_n(1-\xi_n)}{\eta_n\gamma_n}. \end{aligned} \tag{3.6}$$

If we arrange the coefficients on the left-hand side of (3.4) and (3.6) in an upper triangular matrix F_m of size $m \times m$ and the coefficients on the right-hand side of (3.4) and (3.6) in a Hessenberg matrix $\tilde{H}_{m+1,m}$ of size $(m+1) \times m$, i.e.,

$$F_m := \begin{bmatrix} 1 & 0 & -\beta_1\eta_0 & & \\ & 1 & 0 & * & \\ & & 1 & 0 & \ddots \\ & & & 1 & \ddots \\ & & & & \ddots \end{bmatrix}, \quad \tilde{H}_{m+1,m} := \begin{bmatrix} \alpha_0 & 0 & * & * & & \\ \gamma_0 & * & 0 & * & * & \\ & * & * & 0 & * & \ddots \\ & & * & * & 0 & \ddots \\ & & & * & * & \ddots \\ & & & & * & \ddots \\ & & & & & \ddots \end{bmatrix}, \tag{3.7}$$

where the asterisks denote nonzero entries given by (3.4) and (3.6), we can write (3.4) and (3.6) in the following compact form:

$$AV_m := AW_mF_m = W_{m+1}\tilde{H}_{m+1,m}, \tag{3.8}$$

where

$$[v_1, \ldots, v_m] := V_m := W_mF_m \tag{3.9}$$

is another set of basis vectors for $\mathcal{K}_m$.

For BiOS, where $\tau_l(\zeta) = \rho_l(\zeta)$, the horizontal recurrence (2.12) is replaced by

$$w_n^{l+1} = \left(Aw_n^l - w_n^l\alpha_l - w_n^{l-1}\beta_l\right)/\gamma_l. \tag{3.10}$$

In this case we obtain the recursions

$$\begin{aligned} &A\left(w_n^n - w_{n-1}^{n-1}\frac{\beta_n}{\gamma_{n-1}}\right) \\ &\qquad = w_{n+1}^n\gamma_n + w_n^n\alpha_n - w_{n-1}^{n-1}\frac{\beta_n\alpha_{n-1}}{\gamma_{n-1}} - w_{n-1}^{n-2}\frac{\beta_n\beta_{n-1}}{\gamma_{n-1}}, \\ &A\left(w_{n+1}^n - w_n^{n-1}\frac{\beta_n}{\gamma_n}\right) \\ &\qquad = w_{n+1}^{n+1}\gamma_n + w_{n+1}^n\alpha_n - w_n^{n-1}\frac{\alpha_n\beta_n}{\gamma_n} - w_{n-1}^{n-1}\frac{\beta_n^2}{\gamma_n}, \end{aligned} \tag{3.11}$$

which can also be written in the compact form (3.8) with different matrices F_m and $\tilde{H}_{m+1,m}$ that have the same nonzero structure as those in (3.7).

It is tempting to multiply (3.8) from the right by the easily computable inverse F^{-1} to get

$$AW_m = W_{m+1}\tilde{H}_{m+1,m}F_m^{-1} =: W_{m+1}\tilde{\tilde{H}}_{m+1,m}. \tag{3.12}$$

However, $\tilde{\tilde{H}}_{m+1,m}$ will be a nearly full Hessenberg matrix. Since we are interested in short-term recurrences for the QMR iterates, we will base the QMR approach on the relation (3.8). Nevertheless, we would like to mention that the relation (3.12) can

be applied to obtain an estimate for the spectrum of the system matrix A from the eigenvalues of $\tilde{\tilde{H}}_{m,m}$, which could be used in BiOxCheb to construct the Manteuffel ellipse. For this purpose even better estimates of the spectrum can be obtained by extending the approach of computing the roots of the B-orthogonal[2] and residual polynomials, as proposed in [1], to LTPMs. For this, also, the matrix $\tilde{\tilde{H}}_{m,m}$ is needed [25].

With the relation (3.8) at hand, the QMR iterates can be derived in a standard way [15]. If $x_0 \in \mathbb{C}^N$ denotes an arbitrary initial approximation of $A^{-1}b$, the iterates in a Krylov subspace method can be written for some $q_m \in \mathcal{K}_m$ and some $u_m \in \mathbb{C}^m$ as

$$x_m = x_0 + q_m = x_0 + V_m u_m, \tag{3.13}$$

since the columns of V_m form a basis of $\mathcal{K}_m$. Under the assumption that the first product vector is defined by $w_0^0 = b - Ax_0 = r_0$, we conclude from (3.13) by using (3.8) that

$$r_m = b - Ax_m = W_{m+1}\Omega_{m+1}^{-1}\left(\omega_0 e_1^{(m+1)} - \Omega_{m+1}\tilde{H}_{m+1,m}u_m\right), \tag{3.14}$$

with $e_1^{(m+1)} := (1, 0, \ldots, 0)^T \in \mathbb{C}^{m+1}$ and a scaling matrix

$$\Omega_{m+1} = \operatorname{diag}(\omega_0, \ldots, \omega_m); \qquad \omega_k := \begin{cases} \left\| w_{k/2}^{k/2} \right\| & \text{if } k \text{ is even,} \\ \left\| w_{(k+1)/2}^{(k-1)/2} \right\| & \text{if } k \text{ is odd,} \end{cases}$$

which is used to normalize the columns of W_{m+1}. Instead of minimizing $\|r_m\|$, which would be expensive, the QMR approach consists of quasi-minimizing the residual by determining u_m as the solution of

$$\left\|\omega_0 e_1^{(m+1)} - H_{m+1,m}u_m\right\| = \min_{u\in\mathbb{C}^m}\left\|\omega_0 e_1^{(m+1)} - H_{m+1,m}u\right\|, \tag{3.15}$$

where we introduced $H_{m+1,m} := \Omega_{m+1}\tilde{H}_{m+1,m}$. The least-squares problem (3.15) is solved in a standard way by using a QR decomposition of $H_{m+1,m}$,

$$H_{m+1,m} = Q_{m+1}^H R_{m+1,m} =: Q_{m+1}^H \begin{bmatrix} R_m \\ 0 \end{bmatrix}. \tag{3.16}$$

This leads to the solution

$$u_m = R_m^{-1}\,[I_m|0]\,Q_{m+1}\omega_0 e_1^{(m+1)}. \tag{3.17}$$

3.2. QMR updates. It now remains to show that the iterates in (3.13) can be updated by a short-term recurrence. Combining (3.13) with (3.17) yields

$$\begin{aligned} x_m &= x_0 + V_m R_m^{-1}\,[I_m|0]\,Q_{m+1}\omega_0 e_1^{(m+1)} \\ &=: x_0 + P_m\,[I_m|0]\,Q_{m+1}\omega_0 e_1^{(m+1)}, \end{aligned} \tag{3.18}$$

[2]For any Hermitian matrix $B \in \mathbb{C}^{N\times N}$, B-orthogonal means orthogonal with respect to the inner product $\langle B\cdot, \cdot\rangle$.

where we introduced the new direction vectors

$$[p_1, \ldots, p_m] := P_m := V_m R_m^{-1}. \tag{3.19}$$

Since $H_{m+1,m}$ is an upper Hessenberg matrix, we note that the QR decomposition (3.16) can be computed by m Givens rotations. In particular, the QR decomposition (3.16) for m can be easily updated from the one for $m-1$ by applying all previous $m-1$ Givens rotations and one additionally calculated Givens rotation G_m to the last column of $H_{m+1,m}$. Since $H_{m+1,m} = \Omega_{m+1}\widetilde{H}_{m+1,m}$ has upper bandwidth 3 (see (3.7)), R_m in (3.16) has at most upper bandwidth 4. Therefore, the last column of R_m is of the form $[0, \ldots, 0, \mu_{m-4}^{(m)}, \mu_{m-3}^{(m)}, \mu_{m-2}^{(m)}, \mu_{m-1}^{(m)}, \mu_m^{(m)}]^T$. Thus, rewriting (3.19) as $P_m R_m = V_m$, it follows that the new direction vector p_m can be updated by the five-term recurrence

$$p_m = \left(v_m - p_{m-1}\mu_{m-1}^{(m)} - \cdots - p_{m-4}\mu_{m-4}^{(m)}\right)/\mu_m^{(m)} \tag{3.20}$$

after v_m is computed from the product vectors by the two-term recurrence (3.9). Since Q_{m+1} is the product of the newly calculated Givens rotation G_m with $\left[\begin{smallmatrix} Q_m & 0 \\ 0 & 1 \end{smallmatrix}\right]$, the first $m-1$ components of the right-hand side $Q_{m+1}\omega_0 e_1^{(m+1)}$ in (3.18) remain unchanged from step $m-1$ to step m. In particular, we have

$$\left[(t^{(m)})^T, \widetilde{\pi}_{m+1}\right]^T := Q_{m+1}\omega_0 e_1^{(m+1)} = G_m\left[(Q_m\omega_0 e_1^{(m)})^T, 0\right]^T = \left[(t^{(m-1)})^T, \pi_m, \widetilde{\pi}_{m+1}\right]^T.$$

Therefore, we can split the update formula (3.18) into an old and a new part:

$$x_m = x_0 + P_m t^{(m)} = x_0 + P_{m-1}t^{(m-1)} + p_m\pi_m = x_{m-1} + p_m\pi_m. \tag{3.21}$$

3.3. QMR smoothing property. We now want to relate the residuals corresponding to the QMR version of an LTPM, denoted by r_k^{QMR}, to the residuals corresponding to the iterates of the same LTPM that are defined by relation (2.15), i.e., $r_k^{\mathrm{LTPM}} = w_n^l/\hat{\rho}_n^l$ with $k = n + l$ and $l \in \{n-1, n\}$. We follow the approach of Zhou and Walker [35]. To simplify our notation, we define

$$\Lambda_{m+1} := \mathrm{diag}\left(\lambda_0, \ldots, \lambda_m\right), \text{ with } \lambda_k := \begin{cases} \hat{\rho}_{k/2}^{k/2} & \text{if } k \text{ is even,} \\ \hat{\rho}_{(k+1)/2}^{(k-1)/2} & \text{if } k \text{ is odd.} \end{cases}$$

Since $r_k^{\mathrm{LTPM}} \in \mathcal{K}_{k+1}$ we observe that $r_{k-1}^{\mathrm{LTPM}} - r_k^{\mathrm{LTPM}} = As_k$ for some $s_k \in \mathcal{K}_k$. This leads to the relation

$$AS_m = W_{m+1}\Lambda_{m+1}^{-1}\hat{H}_{m+1,m}, \tag{3.22}$$

with

$$S_m := [s_1, \ldots, s_m], \quad \hat{H}_{m+1,m} := \begin{bmatrix} 1 & & \\ -1 & 1 & \\ & \ddots & \ddots \end{bmatrix}.$$

Since the columns of S_m also form a basis of $\mathcal{K}_m$, we can now write the QMR iterates as $x_m^{\mathrm{QMR}} = x_0 + S_m\hat{u}$, for some $\hat{u} \in \mathbb{C}^m$. Thus, we have by (3.22)

$$r_m^{\mathrm{QMR}} = r_0 - AS_m\hat{u} = W_{m+1}\Omega_{m+1}^{-1}\left(\Lambda_{m+1}^{-1}\left(\omega_0 e_1^{(m+1)} - \Omega_{m+1}\hat{H}_{m+1,m}\hat{u}\right)\right). \tag{3.23}$$

By comparing this with (3.14) and (3.15) it follows that $\hat{u}$ is the solution of

$$\min_{u\in\mathbb{C}^m} \left\| \Lambda_{m+1}^{-1} \left(\omega_0 e_1^{(m+1)} - \Omega_{m+1} \hat{H}_{m+1,m} u \right) \right\| := \sigma_m^2. \tag{3.24}$$

To solve the least-squares problem (3.24) we perform the change of variables $\widetilde{u} := [\widetilde{u}_0, \ldots, \widetilde{u}_m]^T := e_1^{(m+1)} - \hat{H}_{m+1,m} u$. This leads to

$$\sigma_m^2 = \min_{\sum_{i=0}^{m} \tilde{u}_i = 1} \sum_{i=0}^{m} \left(\frac{\omega_i}{\lambda_i} \right)^2 \tilde{u}_i^2,$$

which has for $i = 0, 1, \ldots, m$ the solution

$$\tilde{u}_i = \frac{\left(\frac{\lambda_i}{\omega_i} \right)^2}{\sum_{i=0}^{m} \left(\frac{\lambda_i}{\omega_i} \right)^2}, \quad \text{and gives} \quad \sigma_m^2 = \frac{1}{\sum_{i=0}^{m} \left(\frac{\lambda_i}{\omega_i} \right)^2}. \tag{3.25}$$

Inserting (3.25) back into (3.23) results in

$$\begin{aligned} r_m^{\mathrm{QMR}} &= W_{m+1} \Omega_{m+1}^{-1} \left(\Lambda_{m+1}^{-1} \Omega_{m+1} \tilde{u} \right) = W_{m+1} \Lambda_{m+1}^{-1} \tilde{u} \\ &= \frac{1}{\sum_{j=0}^{m} \left(\frac{\lambda_j}{\omega_j} \right)^2} \sum_{k=0}^{m} \left(\frac{\lambda_k}{\omega_k} \right)^2 r_k^{\mathrm{LTPM}} = \frac{1}{\sum_{j=0}^{m} \left\| r_j^{\mathrm{LTPM}} \right\|^{-2}} \sum_{k=0}^{m} \frac{1}{\left\| r_k^{\mathrm{LTPM}} \right\|^2} r_k^{\mathrm{LTPM}}, \end{aligned} \tag{3.26}$$

where we used that $\|r_k^{\mathrm{LTPM}}\| = \frac{\omega_k}{\lambda_k}$. It follows that the QMR residual r_m^{QMR} is a convex combination of the LTPM residual $r_0^{\mathrm{LTPM}}, \ldots, r_m^{\mathrm{LTPM}}$. From the choice of the weights we can directly understand the smoothing property, since they become small for large intermediate residuals.

As already pointed out in [35], this result gives an economical way of obtaining the QMR-LTPM iterates from the iterates and residuals of the corresponding LTPM. From (3.26) it follows that

$$\begin{aligned} r_m^{\mathrm{QMR}} &= \frac{\sigma_m^2}{\sigma_{m-1}^2} r_{m-1}^{\mathrm{QMR}} + \frac{\sigma_m^2}{\left\| r_m^{\mathrm{LTPM}} \right\|^2} r_m^{\mathrm{LTPM}}, \\ x_m^{\mathrm{QMR}} &= \frac{\sigma_m^2}{\sigma_{m-1}^2} x_{m-1}^{\mathrm{QMR}} + \frac{\sigma_m^2}{\left\| r_m^{\mathrm{LTPM}} \right\|^2} x_m^{\mathrm{LTPM}}, \end{aligned} \tag{3.27}$$

where σ_m can be updated from the relation $\frac{\sigma_m^2}{\sigma_{m-1}^2} + \frac{\sigma_m^2 \lambda_m^2}{\omega_m^2} = 1$. In the next section we will compare this implementation of the QMR smoothing with the more expensive one of section 3.2, based on Givens rotations.

4. Numerical examples. For solving large real-world problems it is crucial to combine the iterative solver with preconditioning in order to accelerate the convergence, and thus to reduce the number of iterations necessary to reach a prescribed accuracy and to diminish thereby the influence of roundoff errors. On parallel computer architectures a reduction of the necessary iterations means, in addition, a decrease of the cost for global communication, which is required for the computation of inner products. For different preconditioning strategies, which are also suited for parallel computer architectures, we refer the reader to [26].

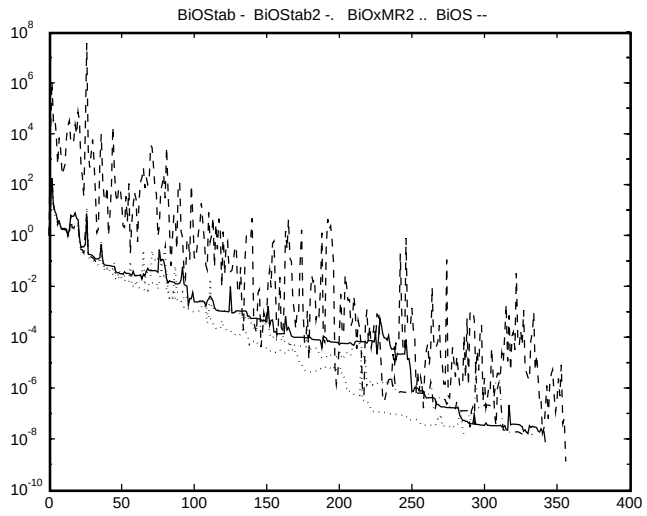

FIG. 4.1. *The recursive residual norm history (i.e.,* $\log(\|r_n\|)$ *vs. iteration number* n*) for the ORSREG*1 *problem solved by different LTPMs.*

Since we aim in the following at a comparison of different iterative methods rather than of different preconditioners, we use in this section a test problem of moderate size, but without preconditioning. The test problem used in all experiments, except for the last one, is the ORSREG1 problem from the OILGEN contribution of the Harwell-Boeing Sparse Matrix Collection, for which the system matrix is real, non-Hermitian, and of order 2205 with 14,133 nonzero entries.

First we compare the LTPMs with each other. In Figure 4.1 we have plotted the recursive residual for all four LTPMs without QMR smoothing. Obviously, the convergence curve for BiOS shows large oscillations, whereas combining the Lanczos process with a local minimization already yields a much smoother convergence. BiOxMR2 is converging slightly faster than BiOStab2, which is slightly faster than BiOStab. To reduce the norm of the recursive residual below 10^{-8}, BiOxMR2 needs about 50 fewer iterations than BiOStab. Because of the oscillating behavior of BiOS, it is hard to compare this LTPM speedwise with the other ones. For example, if we chose as stopping criterion a relative reduction of the recursive residual norm below 10^{-6}, BiOS would win.

Figure 4.2 shows the recursive residual convergence curves for all LTPMs combined now with the QMR smoothing. Here we used the work-saving QMR implementation (3.27) of section 3.3, which we will call in the following Zhou–Walker QMR. The smoothing effect of QMR is clearly demonstrated, especially for BiOS, which is now better compared to the other methods. It may appear redundant to combine the local minimization in BiOxMR2, BiOStab, and BiOStab2 with a global quasiminimization. However, the main purpose of the local minimization is only to extend the Krylov space in a stable way (see the last example below), as long as no look-ahead is needed.

In Figure 4.3 we have plotted the norm of the true residual for all LTPMs without QMR smoothing. All methods start to stagnate, at least after 250 iterations. BiOS starts to stagnate first, with a residual norm of about 10^{-2}, whereas all the other methods reduce the residual to 10^{-6}. This shows clearly the sensitivity of BiOS to roundoff errors. For BiOS, look-ahead is really important [19].

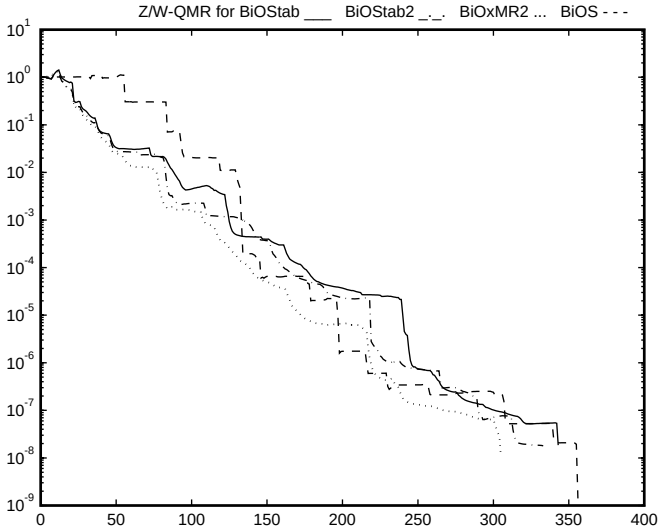

FIG. 4.2. *The recursive residual norm history (i.e.,* $\log(\|r_n\|)$ *vs. iteration number* n*) for the ORSREG*1 *problem solved by different LTPMs combined with Zhou–Walker QMR smoothing.*

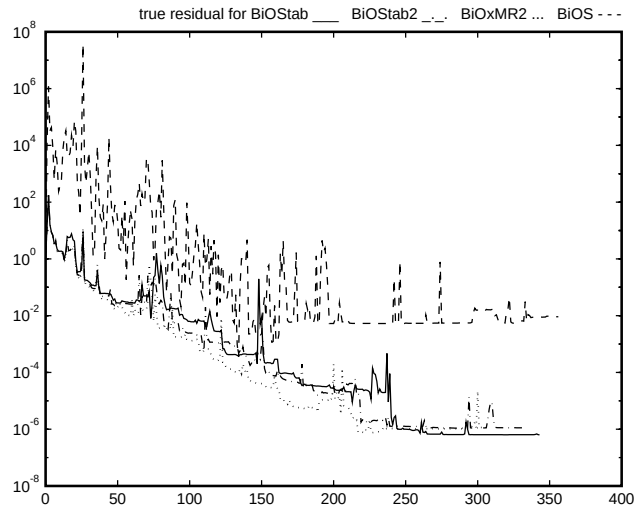

FIG. 4.3. *The true residual norm history (i.e.,* $\log(\|r_n\|)$ *vs. iteration number* n*) for the ORSREG*1 *problem solved by different LTPMs.*

The results for the QMR approach outlined in section 3.2 and in the following Freund–Nachtigal QMR are shown in Figure 4.4. In this QMR approach neither the residual vectors nor their norm is explicitly generated. Instead of using the standard bound for the residual involving the scalars s_k from the Givens rotations [15, eq. (4.11)], we computed in this case the true residual. Except for BiOS, the norm of the true residual can now be reduced to 10^{-8} with this approach.

The direct implementation of the Zhou–Walker QMR smoothing according to (3.27) is based on the recursive residual. As soon as the recursive residual drifts apart from the true residual, the same also happens for the Zhou–Walker QMR residual.

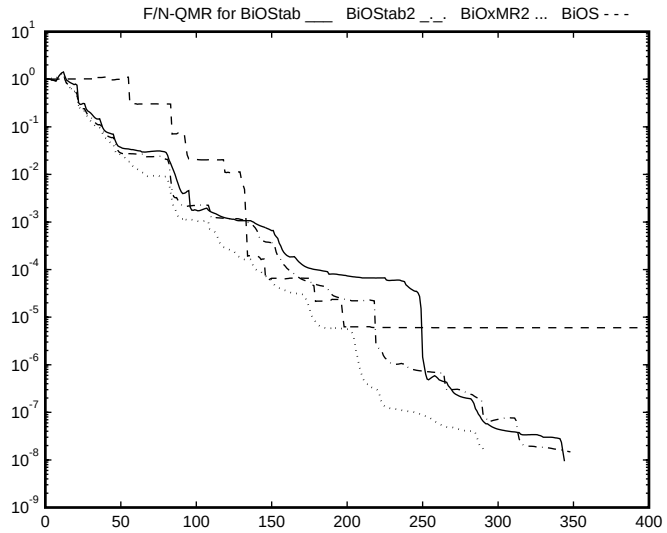

FIG. 4.4. *The true residual norm history (i.e.,* $\log(\|r_n\|)$ *vs. iteration number* n*) for the ORSREG*1 *problem solved by different LTPMs combined with Freund–Nachtigal QMR smoothing.*

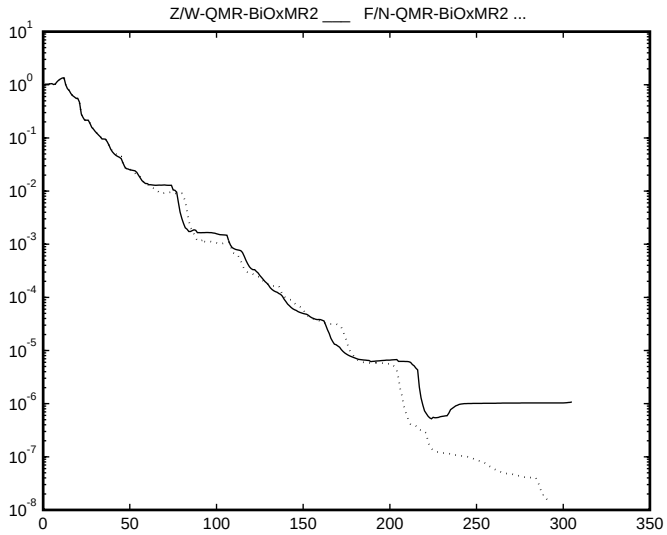

FIG. 4.5. *The true residual norm history (i.e.,* $\log(\|r_n\|)$ *vs. iteration number* n*) for the ORSREG*1 *problem solved by BiOxMR*2 *with Zhou–Walker QMR and Freund–Nachtigal QMR.*

This is also demonstrated in Figure 4.5, where we compare for BiOxMR2 the norm of the true residual of the Freund–Nachtigal QMR with the one from the Zhou–Walker QMR. The more stable implementation of the Zhou–Walker QMR, as proposed in [35], would require here a further multiplication with the system matrix, since $A(x_k - x_{k-1})$ is not directly available from the algorithm because of the use of the Lanczos three-term recurrences.

As indicated above, a main purpose of the local minimization in an LTPM is to extend the Krylov space in a reliable way. However, if η_l computed by the local minimization becomes small, this is no longer true. This is more likely to happen in

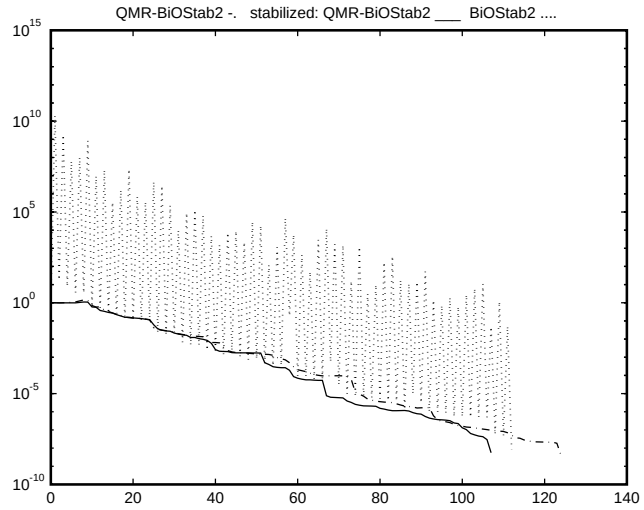

FIG. 4.6. *The recursive residual norm history (i.e.,* $\log(\|r_n\|)$ *vs. iteration number* n*) for the FS*7601 *problem solved by F/N-QMR-BiOStab*2 *(dash-dotted), and by the stabilized versions where* η_l *is replaced by* 1 *if small: BiOStab*2 *(dotted), F/N-QMR-BiOStab*2 *(solid).*

BiOStab and in the even steps of BiOStab2, where a one-dimensional minimization is performed. A simple remedy is to substitute η_l by 1. For the FS7601 problem from the Harwell-Boeing collection, for example, in nearly every even step of BiOStab2, the computed η_l from the one-dimensional minimization is less than $\varepsilon^{1/4}$. Replacing η_l by 1 in this case leads to high oscillations in the residual norm, as shown in Figure 4.6. Of course, these oscillations disappear when QMR is applied, and so this stabilization even reduces the necessary number of iterations.

5. Conclusions. Starting from the one-sided Lanczos bi-orthogonalization process, we systematically derived LTPMs that are based on three-term recurrences. Introducing the notation of product vectors w_n^l and arranging them in a w-table helps to visualize these different methods. Every LTPM computes certain elements of the w-table by moving from the upper left corner downwards to the right. The vertical movement is based on the Lanczos process, whereas for the horizontal movement, the recurrence relation for $\tau_l(\zeta)$ is used.

Squaring the Lanczos polynomial, as in BiOS, leads to the known amplification of the very irregular convergence behavior of the Lanczos process. On the other hand, combining the Lanczos process with a local minimization already yields a much smoother convergence. In general, the convergence speed of these methods is comparable. However, the less smooth converging methods are more sensitive to roundoff errors, which may cause slower convergence, and in particular, the true residual, to deviate earlier from the recursive one. This is especially true for BiOS.

Furthermore, we combined these LTPMs with a QMR smoothing technique that can be implemented in two different ways. The approach due to Zhou and Walker requires less work; however, if the recursive residual is drifting apart from the true residual, then the same will happen for the smoothed residuals. Because of our use of three-term recurrences, the difference of two iterates multiplied by the system matrix is not available in the algorithm, so the more stable version of this QMR smoothing, as proposed in [35], would require an additional matrix–vector multiplication. With the

implementation of the QMR smoothing similar to Freund and Nachtigal [15], these problems do not appear. Except for BiOS, it is in most cases possible to reduce even the true residual below the required tolerance. This also shows how susceptible BiOS is to roundoff errors.

To stabilize the vertical movement, look-ahead procedures for the Lanczos process can be applied [19]. The w-table is then a very important tool for minimizing the overhead of matrix–vector multiplications for these procedures.

To stabilize the horizontal movement, i.e., to avoid a small η_l, a higher-dimensional local minimization could be applied, as in BiCGStab(ℓ). However, for $\ell > 2$, this becomes more expensive. Here the use of the quasi-minimization technique introduces some flexibility in the underlying algorithm. Then we need not constrain the basic product algorithm to generate a residual polynomial with a small norm, because presumably the quasi-minimization step will handle that. We can rather concentrate on extending the Krylov space in a stable way, for example, by just replacing η_l by 1 if it becomes small.

Of course, this raises the question of what the best choice for the polynomial $\tau_l(\zeta)$ would be. This leads to optimal iteration polynomials of hybrid methods, like Chebychev, Faber, or L^2-optimal polynomials. However, this choice would require a good estimate for the spectrum of the system matrix. Further work in this direction will be published elsewhere. We are also in the process of developing look-ahead procedures for LTPMs that are based on the more stable coupled two-term recurrences.

REFERENCES

[1] T. Barth and T. A. Manteuffel, *Estimating the Spectrum of a Matrix Using the Roots of the Polynomials Associated with the QMR Iterations*, Tech. report, Center for Computational Mathematics, University of Colorado at Denver, Denver, CO, 1993.

[2] T. Barth and T. A. Manteuffel, *Variable metric conjugate gradient methods*, in Advances in Numerical Methods for Large Sparse Sets of Linear Equations, N. Natori and T. Nodera, eds., No. 10 in Matrix Analysis and Parallel Computing, PCG '94, Keio University, Yokahama, Japan, 1994, pp. 165–188.

[3] C. Brezinski and M. Redivo Zaglia, *Treatment of near-breakdown in the CGS algorithms*, Numer. Algorithms, 7 (1994), pp. 33–73.

[4] C. Brezinski and M. Redivo Zaglia, *Look-ahead in BI-CGSTAB and other product-type methods for linear systems*, BIT, 35 (1995), pp. 169–201.

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

[6] Z.-H. Cao, *On the QMR approach for iterative methods including coupled three-term recurrences for solving nonsymmetric linear systems*, manuscript, 1995.

[7] 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), pp. 338–347.

[8] J. Cullum, *Peaks, plateaus, numerical instabilities in a Galerkin/minimal residual pair of methods for solving $Ax = b$*, Appl. Numer. Math., 19 (1995), pp. 255–278.

[9] J. Cullum and A. Greenbaum, *Relations between Galerkin and norm-minimizing iterative methods for solving linear systems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 223–247.

[10] V. Faber and T. A. Manteuffel, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.

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

[12] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Generalized conjugate gradient squared*, J. Comput. Appl. Math., 71 (1996), pp. 125–146.

[13] R. W. Freund, *Transpose-free quasi-minimal residual methods for non-Hermitian linear systems*, SIAM J. Sci. Comput., 14 (1993), pp. 470–482.

[14] 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), pp. 137–158.

[15] 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.
[16] M. H. Gutknecht, *Changing the norm in conjugate gradient type algorithms*, SIAM J. Numer. Anal., 30 (1993), pp. 40–56.
[17] M. H. Gutknecht, *Variants of* BICGSTAB *for matrices with complex spectrum*, SIAM J. Sci. Comput., 14 (1993), pp. 1020–1033.
[18] M. H. Gutknecht, *A completed theory of the unsymmetric Lanczos process and related algorithms, Parts* I *and* II, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 594–639 and 15 (1994), pp. 15–58.
[19] M. H. Gutknecht and K. J. Ressel, *Look-ahead procedures for Lanczos-type product methods based on three-term recurrences*, SIAM J. Matrix Anal. Appl., submitted.
[20] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards, 45 (1950), pp. 255–281.
[21] C. Lanczos, *Solution of systems of linear equations by minimized iterations*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 33–53.
[22] T. A. Manteuffel, *The Tchebychev iteration for nonsymmetric linear systems*, Numer. Math., 28 (1977), pp. 307–327.
[23] T. A. Manteuffel, *Adaptive procedure for estimating parameters for the nonsymmetric Tchebychev iteration*, Numer. Math., 31 (1978), pp. 183–208.
[24] 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.
[25] K. J. Ressel, *Hybrid Lanczos-type product methods*, in preparation.
[26] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, MA, 1996.
[27] W. Schönauer, *Scientific Computing on Vector Computers*, Elsevier, Amsterdam, 1987.
[28] G. L. G. Sleijpen and D. R. Fokkema, *BiCGstab(l) for linear equations involving matrices with complex spectrum*, Electron. Trans. Numer. Anal., 1 (1993), pp. 11–32.
[29] P. Sonneveld, *CGS, a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 36–52.
[30] C. H. Tong, *A family of quasi-minimal residual methods for nonsymmetric linear systems*, SIAM J. Sci. Comput., 15 (1994), pp. 89–105.
[31] 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.
[32] R. Weiss, *Convergence Behavior of Generalized Conjugate Gradient Methods*, Ph.D. thesis, University of Karlsruhe, Germany, 1990.
[33] R. Weiss, *Properties of generalized conjugate gradient methods*, J. Numer. Linear Algebra Appl., 1 (1994), pp. 45–63.
[34] S.-L. Zhang, *GPBi-CG: Generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems*, SIAM J. Sci. Comput., 18 (1997), pp. 537–551.
[35] L. Zhou and H. F. Walker, *Residual smoothing techniques for iterative methods*, SIAM J. Sci. Comput., 15 (1994), pp. 297–312.