Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

On Lanczos-type methods for Wilson fermions[*]

M.H. Gutknecht

Research Report No. 2000-03
April 2000

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

[*] To appear in "Numerical Challenges in Lattice Quantum Chromodynamics", Proceedings of the Interdisciplinary Workshop on Numerical Challenges in Lattice QCD, Wuppertal, August 22-24, 1999 (Eds. A. Frommer, Th. Lippert, B. Medeke, K. Schilling); to appear as Lecture Notes in Computational Science and Engineering (LNCSE), Springer-Verlag, Heidelberg, 2000

On Lanczos-type methods for Wilson fermions*

M.H. Gutknecht

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Research Report No. 2000-03 April 2000

Abstract

Numerical simulations of lattice gauge theories with fermions rely heavily on the iterative solution of huge sparse linear systems of equations. Due to short recurrences, which mean small memory requirement, Lanczos-type methods (including suitable versions of the conjugate gradient method when applicable) are best suited for this type of problem. The Wilson formulation of the lattice Dirac operator leads to a matrix with special symmetry properties that makes the application of the classical biconjugate gradient (BiCG) particularly attractive, but other methods, for example BiCGStab and BiCGStab2 have also been widely used. We discuss some of the pros and cons of these methods. In particular, we review the specific simplification of BiCG, clarify some details, and discuss general results on the roundoff behavior.

Keywords: system of linear equations, iterative method, biconjugate gradient method, Lanczos-type method, simplified Lanczos method, roundoff errors, finite precision arithmetic, Wilson fermions, Dirac operator, lattice QCD, quantum chromodynamics

Subject Classification: 65F10, 65F35

*To appear in "Numerical Challenges in Lattice Quantum Chromodynamics", Proceedings of the Interdisciplinary Workshop on Numerical Challenges in Lattice QCD, Wuppertal, August 22-24, 1999 (Eds. A. Frommer, Th. Lippert, B. Medeke, K. Schilling); to appear as Lecture Notes in Computational Science and Engineering (LNCSE), Springer-Verlag, Heidelberg, 2000

1 The symmetry properties of the Wilson fermion matrix

In the Wilson formulation of the lattice Dirac operator, where the Green's function of a single quark with bare mass m is computed by a model based on simple nearest neighbor coupling on a regular 4-dimensional space-time grid with periodic boundary conditions, the resulting linear system $\mathbf{W}\mathbf{x} = \mathbf{b}$ (which in lattice QCD is often written as $M\psi = \phi$) has a coefficient matrix of the form

$$\mathbf{W} = \mathbf{I} - \kappa \mathbf{B}\,, \tag{1}$$

where $\kappa \in \mathbb{R}$ is the so-called hopping parameter and $\mathbf{B}$ is a matrix of order $12 \times l_1 \times l_2 \times l_3 \times l_4$, with l_μ denoting the number of lattice points in dimension μ. Nowadays, typically $l_\mu = 16$, 32, or 64 for all μ, so that the order of the matrix ranges between $12 \times 16^4 = 786,432$ and $12 \times 64^4 \approx 2 \times 10^8$. The matrix $\mathbf{B}$ is well-known to have useful features [1, 16, 17]: first, $\mathbf{B}$ is formally $\mathbf{\Gamma}_5$-Hermitian or $\mathbf{\Gamma}_5$-selfadjoint, in the sense that[1]

$$\mathbf{B}^\star = \mathbf{\Gamma}_5 \mathbf{B} \mathbf{\Gamma}_5\,, \qquad \text{where} \quad \mathbf{\Gamma}_5 = \mathbf{\Gamma}_5^\star = \mathbf{\Gamma}_5^{-1} \tag{2}$$

is a real diagonal matrix with elements ±1, which takes the form

$$\mathbf{\Gamma}_5 :\equiv \mathrm{diag} \begin{bmatrix} 1 & 1 & \cdots & 1 & -1 & -1 & \cdots & -1 \end{bmatrix}$$

if equations and unknowns are ordered appropriately; second, since the underlying discretization is restricted to nearest-neighbor coupling, $\mathbf{B}$ is at the same time "odd/even symmetric" in the sense that

$$\mathbf{\Sigma}\mathbf{B} = -\mathbf{B}\mathbf{\Sigma}\,, \tag{3}$$

where $\mathbf{\Sigma}$ is a diagonal matrix with $+1$'s and -1's. For example, in the two-dimensional case, a diagonal entry of $\mathbf{\Sigma}$ is $+1$ if for the corresponding grid point (i, j) the difference $i - j$ is even. This actually means that $\mathbf{B}$ is a so-called checker board matrix ("Schachbrett-Matrix" in German, see, *e.g.*, , [50]) with the property that $(\mathbf{B})_{k,l} = 0$ if $k - l$ is even. By suitable simultaneous row and column permutations that correspond to a red-black or even-odd reordering $\mathbf{B}$ can be brought into the form

$$\widetilde{\mathbf{B}} :\equiv \begin{bmatrix} \mathbf{O} & \widetilde{\mathbf{B}}_1 \\ \widetilde{\mathbf{B}}_2 & \mathbf{O} \end{bmatrix}\,, \tag{4}$$

which exhibits that $\widetilde{\mathbf{B}}$ is weakly 2-cyclic [47]. In general, $\mathbf{B}$ is a block checkerboard matrix, which can also be brought into the form (4).

The first symmetry, (2), implies that the spectrum of $\mathbf{B}$ is symmetric about the real axis, and the second, (3), entails that the spectrum is also symmetric about the origin, whence the spectrum is actually symmetric about both axes.

For the Wilson fermion matrix $\mathbf{W}$ we have due to (2)

$$\mathbf{W}^\star = \mathbf{\Gamma}_5 \mathbf{W} \mathbf{\Gamma}_5\,, \tag{5}$$

[1] The star denotes the adjoint or conjugate transpose of a matrix.

and the spectrum is symmetric about the real axis and about the point 1. On the other hand, (4) implies that $\mathbf{W}$ has Young's Property A, which makes the linear system suitable for the SOR method [47], in particular since the spectrum is well captured by an ellipse whose larger axis covers only part of the interval $(0, 2)$, as long as κ remains below a critical value. SOR can be expected to converge about twice as fast as the complex Chebyshev iteration [34, 48], which is also an option, but does not take advantage of the (generalized) odd-even structure (4); see [22] for an analogous, but nonlinear problem from another application. SOR and the Chebyshev method require some preliminary knowledge about the spectrum (which may be obtained from a previous application of the biconjugate gradient method), but require no inner products, which is an important advantage on parallel computers.

The systems of the form $\mathbf{W}\mathbf{x} = \mathbf{b}$ that need to be solved are sometimes formally preconditioned with the matrix

$$\mathbf{\Sigma}\mathbf{W}\mathbf{\Sigma} = \mathbf{I} + \kappa\mathbf{B}\,,$$

so that the system matrix becomes

$$(\mathbf{\Sigma}\mathbf{W})^2 = \mathbf{\Sigma}\mathbf{W}\mathbf{\Sigma}\mathbf{W} = \mathbf{I} - \kappa^2\mathbf{B}^2 \tag{6}$$

and is seen to commute with $\mathbf{\Sigma}$, hence, is a block checker board matrix of the other type (with non-zero block diagonal). This implies that the system decouples into two systems of half the size. Also this matrix and, hence, its two diagonal blocks of roughly half the size are $\mathbf{\Gamma}_5$-adjoint. However, neither $\mathbf{W}$ nor $(\mathbf{\Sigma}\mathbf{W})^2$ have a real spectrum.

In contrast, preconditioning by $\mathbf{\Gamma}_5$ yields a linear system with the matrix $\mathbf{\Gamma}_5\mathbf{W}$, for which in view of (2) and $\kappa \in \mathbb{R}$

$$(\mathbf{\Gamma}_5\mathbf{W})^\star = (\mathbf{\Gamma}_5 - \kappa\mathbf{\Gamma}_5\mathbf{B})^\star = \mathbf{\Gamma}_5^\star - \kappa\mathbf{B}^\star\mathbf{\Gamma}_5^\star = \mathbf{\Gamma}_5 - \kappa\mathbf{\Gamma}_5\mathbf{B} = \mathbf{\Gamma}_5\mathbf{W}\,, \tag{7}$$

which shows that $\mathbf{\Gamma}_5\mathbf{W}$ is Hermitian and, thus, has real spectrum.

Since the iterative solution of $\mathbf{W}\mathbf{x} = \mathbf{b}$ is so time and memory consuming, it is crucial to use algorithms that are particularly suitable for this special system and capitalize upon some of the special properties mentioned. We have referred to SOR in connection with the Property A and the special form of the spectrum of $\mathbf{W}$. Relation (7) suggests to apply a solver for Hermitian indefinite systems, such as MinRes, to $\mathbf{\Gamma}_5\mathbf{W}\mathbf{x} = \mathbf{\Gamma}_5\mathbf{b}$. (However, we need to mention that a recent analysis of Sleijpen, van der Vorst, and Modersitzki [43] shows that the limiting accuracy of MinRes is far below that of most other methods.) Boriçi [1] and Frommer et al. [17] have made use of relation (2) for simplifying the biconjugate gradient (BiCG) method, and we will discuss some not so well known details below. Experiments with these and other methods have been documented in many papers, see, for example, [1, 2, 3, 7, 14, 15, 16].

2 The biconjugate gradient method and some related methods

The first Lanczos-type method, introduced in 1952 by Lanczos [33] as the "complete algorithm for minimized iterations", is essentially what we now call the (standard) BiOMin form [27] of the the biconjugate gradient (BiCG) method [8]. It is fully analogous to the classical Hestenes-Stiefel version of the conjugate gradient (CG) method for Hermitian positive definite systems [30], which is also referred to as OMin algorithm for CG. Unlike CG, which is restricted to Hermitian positive definite systems, BiCG is applicable to general nonsingular square systems $\mathbf{A}\mathbf{x} = \mathbf{b}$. However, in contrast to CG, BiCG may break down due to division by zero. If it does not, then, in exact arithmetic, BiCG would converge in at most N steps, if N denotes the order of the system. In finite precision arithmetic, BiCG is strongly influenced by roundoff errors and thus is not guaranteed to converge. But, in practice, when applied to very large systems, we anyway need methods that converge in much fewer than N steps.

Like the classical OMin version of CG, BiOMin is based on a pair of coupled recurrences for the residual and the direction polynomials. These recurrences are used to build up biorthogonal (or, dual) bases for a pair of nested sequences of dual Krylov spaces,

$$\begin{aligned} \mathcal{K}_n &:\equiv \mathcal{K}_n(\mathbf{A}, \mathbf{y}_0) :\equiv \operatorname{span}\,(\mathbf{y}_0, \mathbf{A}\mathbf{y}_0, \ldots, \mathbf{A}^{n-1}\mathbf{y}_0)\,, && (8) \\ \widetilde{\mathcal{K}}_n &:\equiv \mathcal{K}_n(\mathbf{A}^\star, \widetilde{\mathbf{y}}_0) :\equiv \operatorname{span}\,(\widetilde{\mathbf{y}}_0, \mathbf{A}^\star\widetilde{\mathbf{y}}_0, \ldots, (\mathbf{A}^\star)^{n-1}\widetilde{\mathbf{y}}_0)\,, && (9) \end{aligned}$$

$n = 1, 2, \ldots$. The bases consist of the biorthogonal *Lanczos vectors* $\widetilde{\mathbf{y}}_m \in \widetilde{\mathcal{K}}_m$, $\mathbf{y}_n \in \mathcal{K}_n$ satisfying

$$\langle \widetilde{\mathbf{y}}_m, \mathbf{y}_n \rangle = \begin{cases} 0\,, & m \neq n, \\ \delta_n\,, & m = n. \end{cases} \tag{10}$$

At the same time another pair of bases is generated, consisting of the biconjugate *direction vectors* $\widetilde{\mathbf{v}}_m \in \widetilde{\mathcal{K}}_m$, $\mathbf{v}_n \in \mathcal{K}_n$ satisfying

$$\langle \widetilde{\mathbf{v}}_m, \mathbf{A}\mathbf{v}_n \rangle = \begin{cases} 0\,, & m \neq n, \\ \delta'_n\,, & m = n. \end{cases} \tag{11}$$

Additionally, approximations $\mathbf{x}_n \in \mathbf{x}_0 + \mathcal{K}_n$ of the solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$ are computed, and in BiCG, which is a Petrov–Galerkin method, these satisfy $\mathbf{b} - \mathbf{A}\mathbf{x}_n \perp \widetilde{\mathcal{K}}_n$. In view of (10), the (right-hand side) Lanczos vectors $\mathbf{y}_n$ also satisfy $\mathbf{y}_n \perp \widetilde{\mathcal{K}}_n$, and, in fact, they are normally scaled so that they coincide with the residuals, that is, $\mathbf{y}_n = \mathbf{r}_n :\equiv \mathbf{b} - \mathbf{A}\mathbf{x}_n$. Here is a summary of the resulting standard BiCG algorithm[2].

Algorithm 1 (BiOMin form of the BiCG method) *For solving* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *choose an initial approximation* $\mathbf{x}_0$*, set* $\mathbf{v}_0 := \mathbf{y}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$*, and choose* $\widetilde{\mathbf{v}}_0 := \widetilde{\mathbf{y}}_0$ *such that*

[2] The overbar denotes complex conjugation. Complex quantities can be avoided when all data $(\mathbf{A}, \mathbf{b}, \mathbf{x}_0)$ are real. We define the (complex) Euclidean inner product by $\langle \mathbf{z}, \mathbf{y} \rangle :\equiv \mathbf{z}^\star\mathbf{y} = \sum \overline{\zeta_k}\, \eta_k$.

$\delta_0 := \langle \widetilde{\mathbf{y}}_0, \mathbf{y}_0 \rangle \neq 0$ and $\delta'_0 := \langle \widetilde{\mathbf{y}}_0, \mathbf{A}\mathbf{v}_0 \rangle \neq 0$. Then, for $n = 0, 1, \dots$ compute

$$
\begin{aligned}
\omega_n &:= \delta_n / \delta'_n \,, && \text{(12a)}\\
\mathbf{y}_{n+1} &:= \mathbf{y}_n - \mathbf{A}\mathbf{v}_n \omega_n \,, && \text{(12b)}\\
\widetilde{\mathbf{y}}_{n+1} &:= \widetilde{\mathbf{y}}_n - \mathbf{A}^\star \widetilde{\mathbf{v}}_n \overline{\omega_n} \,, && \text{(12c)}\\
\mathbf{x}_{n+1} &:= \mathbf{x}_n + \mathbf{v}_n \omega_n \,, && \text{(12d)}\\
\delta_{n+1} &:= \langle \widetilde{\mathbf{y}}_{n+1}, \mathbf{y}_{n+1} \rangle, && \text{(12e)}\\
\psi_n &:= -\delta_{n+1}/\delta_n \,, && \text{(12f)}\\
\mathbf{v}_{n+1} &:= \mathbf{y}_{n+1} - \mathbf{v}_n \psi_n \,, && \text{(12g)}\\
\widetilde{\mathbf{v}}_{n+1} &:= \widetilde{\mathbf{y}}_{n+1} - \widetilde{\mathbf{v}}_n \overline{\psi_n} \,, && \text{(12h)}\\
\delta'_{n+1} &:= \langle \widetilde{\mathbf{v}}_{n+1}, \mathbf{A}\mathbf{v}_{n+1} \rangle. && \text{(12i)}
\end{aligned}
$$

If $\mathbf{y}_{n+1} \approx \mathbf{o}$, the process terminates and $\mathbf{x}_{n+1}$ is the solution; if $\delta_{n+1} \approx 0$ (and hence $\psi_n \approx 0$) or $\delta'_{n+1} \approx 0$, but $\mathbf{y}_{n+1} \not\approx \mathbf{o}$, the algorithm breaks down ("Lanczos and pivot breakdowns", respectively).

The recurrence coefficients ω_n and ψ_{n-1} are chosen so that the conditions (10) and (11) are satisfied for $m = n - 1$. The most important feature of BiCG is that, in exact arithmetic, the other of these conditions are then satisfied automatically: the corresponding orthogonality is inherited — at least in exact arithmetic.

By eliminating the direction vectors from the recurrences of Algorithm 1 we obtain the BiORes form of the BiCG method, where the Lanczos vectors are generated by three-term recurrences; see, *e.g.*, [27] for this connection, which is based on an LU decomposition of a tridiagonal matrix:

Algorithm 2 (BiORes form of the BiCG method) *To solve $\mathbf{A}\mathbf{x} = \mathbf{b}$, choose an initial approximation $\mathbf{x}_0$, set $\mathbf{y}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$, and choose $\widetilde{\mathbf{y}}_0$ such that $\delta_0 := \langle \widetilde{\mathbf{y}}_0, \mathbf{y}_0 \rangle \neq 0$. Set $\beta_{-1} := 0$. Then, for $n = 0, 1, \dots$ compute*

$$
\begin{aligned}
\delta_n^{\mathbf{A}} &:= \langle \widetilde{\mathbf{y}}_n, \mathbf{A}\mathbf{y}_n \rangle \,, && \text{(13a)}\\
\alpha_n &:= \delta_n^{\mathbf{A}} / \delta_n \,, && \text{(13b)}\\
\beta_{n-1} &:= \gamma_{n-1}\delta_n/\delta_{n-1} \qquad (\textit{if} \quad n > 0), && \text{(13c)}\\
\gamma_n &:= -\alpha_n - \beta_{n-1} \,, && \text{(13d)}\\
\mathbf{y}_{n+1} &:= (\mathbf{A}\mathbf{y}_n - \mathbf{y}_n\alpha_n - \mathbf{y}_{n-1}\beta_{n-1})/\gamma_n \,, && \text{(13e)}\\
\widetilde{\mathbf{y}}_{n+1} &:= (\mathbf{A}^\star\widetilde{\mathbf{y}}_n - \widetilde{\mathbf{y}}_n\overline{\alpha_n} - \widetilde{\mathbf{y}}_{n-1}\overline{\beta_{n-1}})/\overline{\gamma_n} \,, && \text{(13f)}\\
\delta_{n+1} &:= \langle \widetilde{\mathbf{y}}_{n+1}, \mathbf{y}_{n+1} \rangle \,, && \text{(13g)}\\
\mathbf{x}_{n+1} &:= -(\mathbf{y}_n + \mathbf{x}_n\alpha_n + \mathbf{x}_{n-1}\beta_{n-1})/\gamma_n \,. && \text{(13h)}
\end{aligned}
$$

If $\gamma_n \approx 0$, the algorithm breaks down ("pivot breakdown"). If $\mathbf{y}_{n+1} \approx \mathbf{o}$, it terminates and $\mathbf{x}_{n+1}$ is the solution. If $\mathbf{y}_{n+1} \not\approx \mathbf{o}$, but $\delta_{n+1} \approx 0$, it also breaks down ("Lanczos breakdown").

A serious shortcoming of BiCG and related, so-called Lanczos-type methods for nonsymmetric systems is the possibility of breakdowns. These were probably the main reason why for decades numerical analysts were very reluctant to apply or even promote this method. Finally, look-ahead steps were introduced to circumnavigate such breakdowns [38, 24, 26, 11]. It was also noticed that in practice breakdowns and near-breakdowns with serious effects are quite rare. Ever since, BiCG and other Lanczos-type methods have become more and more popular, although look-ahead is rarely implemented.

Another disadvantage is that in contrast to most other Krylov space methods BiCG requires two matrix-vector products (MVs) per step, but only increases the search space $\mathcal{K}_n$ by one dimension. This disadvantages of BiCG was overcome by Sonneveld [44] with the introduction of the *conjugate gradient squared* (CGS) *method*, which should rather be called BiCGS and will be referred to here as (Bi)CGS. Sonneveld's clever idea was to derive recurrences that produce approximations $\mathbf{x}_n \in \mathbf{x}_0 + \mathcal{K}_{2n}$ whose residuals $\mathbf{r}_n \in \mathcal{K}_{2n+1}$ correspond to the squares p_n^2 of the residual polynomials p of BiCG (which are often referred to as the Lanczos polynomials). If we denote by $\widehat{p}_n$ the polynomials that are associated with the direction vectors $\mathbf{v}_n$ of BiCG, then these recurrences involve, in addition to the iterates $\mathbf{x}_n$, the vector sequences

$$\begin{aligned} \mathbf{r}_n &:\equiv p_n^2(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_{2n+1}\,, & \mathbf{s}_n &:\equiv p_n(\mathbf{A})\widehat{p}_n(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_{2n+1}\,, \\ \mathbf{s}'_n &:\equiv p_{n+1}(\mathbf{A})\widehat{p}_n(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_{2n+2}\,, & \widehat{\mathbf{r}}_n &:\equiv \widehat{p}_n^2(\mathbf{A})\mathbf{r}_0 \in \mathcal{K}_{2n+1}\,. \end{aligned}$$

They are easily derived from the BiOMin recurrences (12b)–(12c) and (12g)–(12h), and they contain the same coefficients $\omega_n := \delta_n/\delta'_n$ and $\psi_n := -\delta_{n+1}/\delta_n$. Note that

$$\begin{aligned} \delta_n &= \langle \widetilde{\mathbf{y}}_n, \mathbf{y}_n \rangle = \langle \overline{p_n}(\mathbf{A}^\star)\widetilde{\mathbf{y}}_0, p_n(\mathbf{A})\mathbf{r}_0 \rangle = \langle \widetilde{\mathbf{y}}_0, p_n^2(\mathbf{A})\mathbf{r}_0 \rangle = \langle \widetilde{\mathbf{y}}_0, \mathbf{r}_n \rangle, \\ \delta'_n &= \langle \widetilde{\mathbf{v}}_n, \mathbf{A}\mathbf{v}_n \rangle = \langle \overline{\widehat{p}_n}(\mathbf{A}^\star)\widetilde{\mathbf{v}}_0, \mathbf{A}\widehat{p}_n(\mathbf{A})\mathbf{r}_0 \rangle = \langle \widetilde{\mathbf{y}}_0, \widehat{p}_n^2(\mathbf{A})\mathbf{A}\mathbf{r}_0 \rangle = \langle \widetilde{\mathbf{y}}_0, \mathbf{A}\widehat{\mathbf{r}}_n \rangle. \end{aligned} \tag{14}$$

A typical behavior of BiCG is that the residual norms $\|\mathbf{y}_n\|$ fluctuate strongly, in particular when the problem solved is ill conditioned and, consequently, the convergence is rather slow. In (Bi)CGS this erratic convergence behavior is even more pronounced. One way to counteract it is by replacing the residual polynomials p_n^2 of (Bi)CGS by a more general product $p_n t_n$, where t_n belongs to another polynomial sequence satisfying a short recurrence. This leads to *Lanczos-type product methods* (*LTPMs*). The first algorithm of this class was BiCGStab, due to van der Vorst [45], where t_n is built up from linear factors: $t_{n+1}(\zeta) = (1 - \chi_{n+1}\zeta)t_n(\zeta)$, and where the sequences

$$\mathbf{r}_n :\equiv p_n(\mathbf{A})t_n(\mathbf{A})\mathbf{r}_0\,, \quad \widehat{\mathbf{r}}_n :\equiv \widehat{p}_n(\mathbf{A})t_n(\mathbf{A})\mathbf{r}_0\,, \quad \mathbf{w}_n :\equiv p_{n+1}(\mathbf{A})t_n(\mathbf{A})\mathbf{r}_0\,,$$

are constructed in addition to the iterates $\mathbf{x}_n$. The coefficient χ_{n+1} is chosen such that

$$||\mathbf{r}_{n+1}|| = \min_{\chi} \; ||\mathbf{w}_n - \mathbf{A}\mathbf{w}_n\chi||\,.$$

For BiCGStab the residual norm history is typically much smoother than for (Bi)CGS, but a disadvantage of this method is that the zeros $1/\chi_n$ of the second set $\{t_n\}$ of polynomials are necessarily all real when a real-valued problem is solved in real arithmetic, even when the spectrum of the matrix is truly complex. Moreover, they remain fixed for all subsequent polynomials of the set. The first disadvantage is avoided if the linear factors are replaced by quadratic factors that are attached every other step; they allow a two-dimensional residual minimization in every other step, as suggested in BiCGStab2 [25]. The second disadvantage is overcome if the second set is chosen to satisfy a three-term recurrence or a pair of coupled two-term recurrences (which can be used for a two-dimensional residual minimization in every step), as suggested by Zhang in his GPBI-CG algorithm [49] (an equivalent form of which is called BiCG×MR2 in [27]).

3 Simplifications due to symmetries

When $\mathbf{A}$ is Hermitian, the choice $\widetilde{\mathbf{y}}_0 := \mathbf{y}_0$ will produce in the BiORes algorithm identical left and right vectors, $\widetilde{\mathbf{y}}_n = \mathbf{y}_n$ $(\forall n)$, and in the BiOMin algorithm also $\widetilde{\mathbf{v}}_n = \mathbf{v}_n$ $(\forall n)$, so that there is no need to compute the left sequences $\{\widetilde{\mathbf{y}}_n\}$ and $\{\widetilde{\mathbf{v}}_n\}$. This is easily seen by induction and by noting that the numbers α_n, β_n, γ_n, ψ_n, and ω_n are real, even when $\mathbf{A}$ or $\mathbf{y}_0$ are complex. The resulting simplified algorithms are then exactly the OMin and the ORes algorithms, respectively, for CG; they may be applied also to indefinite Hermitian systems, but then they can break down too.

One may raise the question whether there are other situations where BiCG simplifies in the sense that only one MV is required per step. In 1953, Rutishauser [39] and later Fletcher [8], both assuming real data, pointed out that in the three-term Lanczos process (and thus also in BiORes, which just makes use of the special normalization $\gamma_n := -\alpha_n - \beta_{n-1}$), the knowledge of a matrix $\mathbf{S}$ satisfying

$$\mathbf{A}^\top = \mathbf{S}\mathbf{A}\mathbf{S}^{-1} \tag{15}$$

allows us to make such a reduction: choosing $\widetilde{\mathbf{y}}_0 := \mathbf{S}\mathbf{y}_0$ yields $\widetilde{\mathbf{y}}_n := \mathbf{S}\mathbf{y}_n$ $(n > 0)$. In [23] we mentioned this again and made the simple observation that the complex case is covered too when we choose $\widetilde{\mathbf{y}}_0 := \overline{\mathbf{S}\mathbf{y}_0}$, which then yields $\widetilde{\mathbf{y}}_n := \overline{\mathbf{S}\mathbf{y}_n}$ $(n > 0)$, as is readily verified. Thus we can delete (13f) if we replace (13a) and (13g) by

$$\begin{aligned} \delta_n^{\mathbf{A}} &:= \langle \overline{\mathbf{y}_n}, \mathbf{A}\mathbf{y}_n \rangle_{\mathbf{S}^\top}\,, & (16)\\ \delta_{n+1} &:= \langle \overline{\mathbf{y}_{n+1}}, \mathbf{y}_{n+1} \rangle_{\mathbf{S}^\top}\,, & (17) \end{aligned}$$

respectively, where

$$\langle \mathbf{z}, \mathbf{y} \rangle_{\mathbf{S}^\top} :\equiv \langle \mathbf{z}, \mathbf{S}^\top \mathbf{y} \rangle = \mathbf{z}^\star \mathbf{S}^\top \mathbf{y}\,. \tag{18}$$

The BiOMin algorithm simplifies in a fully analogous way: we just need additionally

$$\delta'_{n+1} := \langle \overline{\mathbf{v}_{n+1}}, \mathbf{A}\mathbf{v}_{n+1} \rangle_{\mathbf{S}^\top} \tag{19}$$

in order to delete (12c) and (12h). Rutishauser also pointed out that a matrix $\mathbf{S}$ satisfying (15) always exists, as every matrix is known to be similar to its transposed, but the usual proof for this makes use of the Jordan canonical form, see, *e.g.*, [32, p. 134]. Of course, the simplification is only useful, if $\mathbf{S}$ is known and if the matrix-vector products $\mathbf{S}\mathbf{y}_n$ are cheaper than $\mathbf{A}^\star\widetilde{\mathbf{y}}_n$.

Note that (15) does not include the simple Hermitian case $\mathbf{A}^\star = \mathbf{A}$, but it covers the complex symmetric case (where $\mathbf{S}= \mathbf{I}$), which was treated in detail by Freund [9].

In [10] Freund looked for further situations where the Lanczos process simplifies, and in particular for classes of matrices where the matrix $\mathbf{S}$ is known due to the special structure of $\mathbf{A}$. For example, for a Toeplitz matrix (15) holds with $\mathbf{S}$ the antidiagonal unit matrix $\mathbf{J}$. More generally, a matrix satisfying (15) with $\mathbf{S} = \mathbf{J}$ is symmetric about the antidiagonal and is called persymmetric. In [10] Freund treated the cases

$$\mathbf{A}^\top = \mathbf{S}\mathbf{A}\mathbf{S}^{-1}\,, \qquad \mathbf{S} = \mathbf{S}^\top\,, \tag{20}$$

and

$$\mathbf{A}^\star = \mathbf{S}\mathbf{A}\mathbf{S}^{-1}\,, \qquad \mathbf{S} = \mathbf{S}^\star\,. \tag{21}$$

Clearly, (20) is a special case of (15) (the extra condition $\mathbf{S} = \mathbf{S}^\top$ is not needed for the simplification), but for complex matrices (21) is different. In [13], Freund and Nachtigal then referred to the two cases (15) and

$$\mathbf{A}^\star = \mathbf{S}\mathbf{A}\mathbf{S}^{-1}\,, \tag{22}$$

that is, they dropped the symmetry assumption for $\mathbf{S}$ in (21), which, however, seems to be wrong[3] In fact, the recipe is to choose[4] $\widetilde{\mathbf{y}}_0 := \mathbf{S}\mathbf{y}_0$ in BiCG and to aim for $\widetilde{\mathbf{y}}_n = \mathbf{S}\mathbf{y}_n$ $(n > 0)$. Inserting this and (22) in (13f) leads after premultiplication with $\mathbf{S}^{-1}$ to

$$\mathbf{y}_{n+1} := (\mathbf{A}\mathbf{y}_n - \mathbf{y}_n\overline{\alpha_n} - \mathbf{y}_{n-1}\overline{\beta_{n-1}})/\overline{\gamma_n}\,, \tag{23}$$

which differs from (13e) only in the complex conjugated coefficients. By making additionally use of $\mathbf{S}^\star = \mathbf{S}$, we see that (21) implies that

$$(\mathbf{S}\mathbf{A})^\star = \mathbf{S}\mathbf{A}\,, \tag{24}$$

that is $\mathbf{SA}$ is Hermitian. Consequently,

$$\delta_n \;:\equiv\; \langle \widetilde{\mathbf{y}}_n, \mathbf{y}_n\rangle = \langle \mathbf{y}_n, \mathbf{S}\mathbf{y}_n\rangle \in \mathbb{R}\,, \tag{25}$$

$$\delta_n^{\mathbf{A}} \;:\equiv\; \langle \widetilde{\mathbf{y}}_n, \mathbf{A}\mathbf{y}_n\rangle = \langle \mathbf{y}_n, \mathbf{S}\mathbf{A}\mathbf{y}_n\rangle \in \mathbb{R}\,, \tag{26}$$

so that $\alpha_n \in \mathbb{R}$, $\beta_{n-1} \in \mathbb{R}$, and $\gamma_n \in \mathbb{R}$. Therefore, under the assumption (21) the BiORes algorithm can indeed be simplified, and the same is true for BiOMin since, when $\widetilde{\mathbf{v}}_n := \mathbf{S}\mathbf{v}_n$ also

$$\delta_n' :\equiv \langle \widetilde{\mathbf{v}}_n, \mathbf{v}_n\rangle = \langle \mathbf{v}_n, \mathbf{S}\mathbf{A}\mathbf{v}_n\rangle \in \mathbb{R}\,, \tag{27}$$

so that $\psi_n \in \mathbb{R}$ and $\omega_n \in \mathbb{R}$. This can all be recast in a proof by induction showing that choosing $\widetilde{\mathbf{y}}_0 := \mathbf{S}\mathbf{y}_0 := \mathbf{S}\mathbf{y}_0$ yields $\widetilde{\mathbf{y}}_n = \mathbf{S}\mathbf{y}_n$ for $n > 0$ in BiORes, and likewise, additionally choosing $\widetilde{\mathbf{v}}_0 := \mathbf{S}\mathbf{v}_0 := \mathbf{S}\mathbf{y}_0$ in BiOMin implies $\widetilde{\mathbf{v}}_n = \mathbf{S}\mathbf{v}_n$ for $n > 0$. In

[3] We must admit that we made the same mistake in a remark in §6.1 of [27], where we moreover claimed incorrectly that (21) implies that the spectrum of $\mathbf{A}$ is real.

[4] Note that Freund uses in the complex Lanczos process a formal, bilinear inner product $\mathbf{w}^\top\mathbf{y}$ instead of the usual sesquilinear inner product $\langle \widetilde{\mathbf{y}}, \mathbf{y}\rangle = \widetilde{\mathbf{y}}^\star\mathbf{y}$; therefore, up to a scalar factor, his left Lanczos vectors $\mathbf{w}_n$ and ours are related by $\widetilde{\mathbf{y}}_n = \overline{\mathbf{w}_n}$.

summary, for simplifying BiORes and BiOMin when (21) holds, we redefine δ_n, $\delta_n^{\mathbf{A}}$, and δ'_n as given in (25), (26), and (27) in order to delete (13f) in BiORes and (12c), (12h) in BiOMin, as has been proposed by Boriçi [1] and Frommer et al. [17] for the Wilson fermion computations.

Without the condition, $\mathbf{S} = \mathbf{S}^\star$, that is, assuming (22) alone, does not seem to lead to such a simplification, even if we turn to the most general versions of the Lanczos process [27] where γ_n and $\overline{\gamma_n}$ can be chosen freely (the latter need not be the complex conjugate of the former).

Software for the so simplified BiCG algorithms (and of the related QMR algorithm that is not discussed here) is available from

`http://www.math.uni-wuppertal.de/org/SciComp/Projects/QCD.html`

4 Finite precision effects

Roundoff errors can have strong effects on Lanczos-type methods (including CG). This is first of all due to the fact that the methods rely essentially on a variation of the Gram-Schmidt process, which is known to be prone to roundoff effects. Second, particularly in the nonsymmetric case, the computed recurrence coefficients may turn out to have large relative error. Third, the residuals are normally updated using recurrences and, hence, may differ considerably from the true residuals of the approximations $\mathbf{x}_n$. We will now discuss these three types of finite precision effects.

4.1 Loss of orthogonality and loss of linear independence

Recall that, for example, in CG and BiCG a Gram-Schmidt process is applied to make the residual $\mathbf{y}_{n+1}$ orthogonal to the earlier ones or the earlier left Lanczos vectors, respectively. The Gram-Schmidt process makes vectors shorter due to the subtraction of certain projections, and thus tiny errors in the coefficients or in the computation of the linear combinations may ultimately cause large relative errors, and, in particular, a *loss of orthogonality*: $\mathbf{y}_{n+1}$ will not be exactly orthogonal to $\widetilde{\mathbf{y}}_0, \ldots, \widetilde{\mathbf{y}}_n$. This loss of orthogonality is often severe enough to lead to a *loss of linear independence.*

Special is here that it suffices to enforce the orthogonality to $\widetilde{\mathbf{y}}_n$ and $\widetilde{\mathbf{y}}_{n-1}$ (in the case of BiCG), because the orthogonality to $\widetilde{\mathbf{y}}_0, \ldots, \widetilde{\mathbf{y}}_{n-2}$ is inherited in exact arithmetic. This has the advantage that there are only two subtractions, hence the resulting vector $\mathbf{y}_{n+1}\gamma_n$ will not be so much shorter than the one we started with, $\mathbf{A}\mathbf{y}_n$, but the drawback is that the loss of orthogonality may be worse since previous errors are inherited too. Hence, often

$$\frac{\langle \widetilde{\mathbf{y}}_m, \mathbf{y}_n \rangle}{\|\widetilde{\mathbf{y}}_m\| \, \|\mathbf{y}_n\|} \not\approx 0 \quad \text{if} \quad |m-n| \quad \text{large.}$$

For example, the fraction can be easily on the order of 10^{-1}.

This loss of orthogonality is particularly annoying when the tridiagonal matrix $\mathbf{T}_n$ generated in the Lanczos process is used to find approximate eigenvalues of $\mathbf{A}$, because it will cause $\mathbf{T}_n$ to have multiple copies of some of these eigenvalues. Full reorthogonalization, that is, repeating the Gram-Schmidt process with respect to the full set of Lanczos vectors (instead of the last two) would help, but the cost forbids this, since it would be necessary to store all Lanczos vectors. Two strategies have been developed to cope with this difficulty: either the so-called ghost eigenvalues are identified and removed as proposed by Cullum and Willoughby [4], or their creation is avoided by reducing the roundoff errors of the Lanczos vectors, as suggested by Parlett and his group [36, 37, 6, 5]. However, it does not suffice to reduce the roundoff in the three-term Gram-Schmidt process (where $\mathbf{y}_{n+1}$ is made orthogonal to $\widetilde{\mathbf{y}}_n$ and $\widetilde{\mathbf{y}}_{n-1}$) by applying modified Gram-Schmidt or repeated classical Gram-Schmidt (there is little benefit because there are only three terms). Additionally, $\mathbf{y}_{n+1}$ needs to be reorthogonalized with respect to a selection of earlier Lanczos vectors. In the symmetric case, where this technique was explored first, this *selective reorthogonalizition* can be justified by Paige's roundoff analysis for the symmetric Lanczos process. The nonsymmetric case was later treated by Day [6, 5], who systematically explored measures for *maintaining duality*, that is, biorthogonality.

Since selective reorthogonalizition increases the program complexity and the memory requirements, computational physicists tend to prefer the Cullum and Willoughby filtering.

We should mention, however, that even if $\mathbf{T}_n$ is affected by large roundoff errors occuring in the Lanczos process, the implications are not completely devastating: groups of ghost eigenvalues somehow maintain the projection properties of the operator.

For solving linear systems it seems not really worth-while to apply all these tricks. Moreover, it seems to be impossible to adapt them to LTPMs, which are now considered to be the most effective solvers. In fact, when linear systems are solved, the loss of linear independence caused by a loss of orthogonality will just entail a *slowdown of the convergence*. In the symmetric case, this mechanism is well understood due to the work of Greenbaum and Strakoš [18, 21]: *in finite precision arithmetic, the Lanczos process behaves like one for a bigger problem in exact arithmetic.*

4.2 Inaccurate recurrence coefficients and near-breakdowns

To some extent, inaccurate recurrence coefficients in CG and BiCG are clearly linked to the loss of orthogonality just discussed. One effect adds to the other: inaccurate Lanczos vectors lead to inaccurate coefficients, and vice versa. However, in the non-Hermitian case, there is the additional danger that the inner products δ_n and δ'_n may be very small even if the vectors they are formed from are not short. Since an inner product of nearly orthogonal vectors is inherently prone to large relative roundoff error, these cases are dangerous: the quantities computed from δ_n and δ'_n will also have large error. This is then called a *near-breakdown*, since, when one of these inner products is needed and turns out to be exactly 0, then the corresponding algorithm breaks down due to a division by zero (*exact breakdown*). We refer to the case $\delta_n \approx 0$ as *Lanczos breakdown*, and to $\delta'_n \approx 0$ as *pivot breakdown*. In the Hermitian indefinite case (where $\widetilde{\mathbf{y}}_n = \mathbf{y}_n$) Lanczos breakdowns cannot occur, but pivot breakdowns still can. In the BiORes version of BiCG and in the ORes version of CG (when applied to a Hermitian indefinite system), the pivot breakdown reappears as $\gamma_n \approx 0$. Only in the case of a Hermitian positive definite (Hpd) system, CG cannot break down.

Specifically, in BiORes the recurrence coefficients α_n and/or β_{n-1} are inaccurate if any of the inner products

$$\delta_n :\equiv \langle \widetilde{\mathbf{y}}_n, \mathbf{y}_n \rangle \,, \qquad \delta_{n-1} :\equiv \langle \widetilde{\mathbf{y}}_{n-1}, \mathbf{y}_{n-1} \rangle \,, \qquad \delta_n^{\mathbf{A}} :\equiv \langle \widetilde{\mathbf{y}}_n, \mathbf{A}\mathbf{y}_n \rangle$$

has large relative error. Moreover, $\gamma_n :\equiv -\alpha_n - \beta_{n-1}$ may be inaccurate if $|\gamma_n| \ll |\alpha_n|$.

Similarly, in BiOMin, ψ_{n-1} and/or ω_n are inaccurate if any of the inner products

$$\delta_n :\equiv \langle \widetilde{\mathbf{y}}_n, \mathbf{y}_n \rangle \,, \qquad \delta_{n-1} :\equiv \langle \widetilde{\mathbf{y}}_{n-1}, \mathbf{y}_{n-1} \rangle \,, \qquad \delta'_n :\equiv \langle \widetilde{\mathbf{v}}_n, \mathbf{A}\mathbf{v}_n \rangle$$

has large relative error. And likewise, the same types of inaccuracy occur in (Bi)CGS if any of the inner products

$$\delta_n :\equiv \langle \widetilde{\mathbf{y}}_0, \mathbf{r}_n \rangle \,, \qquad \delta_{n-1} :\equiv \langle \widetilde{\mathbf{y}}_0, \mathbf{r}_{n-1} \rangle \,, \qquad \delta'_n :\equiv \langle \widetilde{\mathbf{y}}_0, \mathbf{A}\widehat{\mathbf{r}}_n \rangle$$

has large relative error; see (14).

Except for $\delta_n^{\mathbf{A}} \approx 0$, all of these cases cause near-breakdowns. In particular, a pivot near-breakdown ($\gamma_n \approx 0$ in BiORes or $\delta'_n \approx 0$ in BiOMin) causes not only large local errors in coefficients and vectors, but also very large vectors $\mathbf{x}_n$ and $\mathbf{r}_n$.

Fortunately, in linear system solvers, inaccurate recurrence coefficients only seem to have a strong effect on the convergence when the coefficients are very inaccurate, as it may happen when a near-breakdown occurs. As we mentioned before, for computing approximate eigenvalues the situation is different.

The bad effects of a breakdown or near-breakdown can be avoided by switching to look-ahead steps when necessary, a technique that was developed in [38] for eigenvalue computations, in [24, 26, 11, 12] for various versions of BiCG, and in [28] for LTPMs. Additional contributions and alternative approaches are referred to in [27] and [28], where also a simplification due to Hochbruck [31] is covered. In particular, divisions by near-zeros can be avoided by look-ahead, but there are still some open questions regarding the best strategy for its application.

Further possibilities to improve the accuracy of the recurrence coefficients include, in addition to those for reducing the loss of orthogonality mentioned in the previous subsection:

(i) The application of multiple precision arithmetic to compute the above inner products and the sum $\gamma_n :\equiv -\alpha_n - \beta_{n-1}$, an option one tries to avoid.

(ii) Alternative choices for the left vectors in BiCG. It is well-known (see Algorithm 3 in Saad [40] and Section 6.2 in [27]) that in BiCG the left vectors $\widetilde{\mathbf{y}}_n$ need not be chosen as Lanczos vectors, but could come from another nested basis for the dual space. In fact, LPTMs capitalize exactly upon this freedom. However, experiments done independently by Miroslav Rozložník (private communication) and the author have not turned out a convincing choice different from the standard one. Indeed the theory provides little hope for success in this way.

(iii) Replacing (Bi)CGS by an LTPM with suitably chosen second set of polynomials t_n. However, again there is limited hope for a strong improvement.

4.3 The gap between updated and true residuals

In most Krylov space methods one has the option to compute the residuals $\mathbf{r}_n :\equiv \mathbf{b} - \mathbf{A}\mathbf{x}_n$ explicitly according to this definition or by updating, that is by using some recursion(s) like the coupled two-term recursions (12b), (12g) of BiOMin and the three-term recursion (13e) of BiORes (recall that in BiCG $\mathbf{r}_n = \mathbf{y}_n$). In some cases, the explicit evaluation costs an extra matrix-vector product (MV), but normally another one can be avoided instead. Nevertheless, the folklore is that updating should be used because explicit computation adds to the roundoff in the process of generating the Krylov space, that is, produces a less accurate basis, and thus often slows down convergence. We therefore assume here that the residuals are computed by updating, and we let $\mathbf{r}_n$ denote the nth residual vector so obtained in finite precision arithmetic. Likewise, $\mathbf{x}_n$ is now the iterate computed in finite precision arithmetic, and $\mathbf{b} - \mathbf{A}\mathbf{x}_n$ is the *true residual* obtained in exact arithmetic from $\mathbf{x}_n$. (Actually, in numerical experiments the true residuals are computed in finite precision arithmetic too, but the error in the evaluation of this expression will normally be considerably smaller than the true residual itself, and this is all that is needed in this context.) Clearly, a *gap*

$$\mathbf{f} :\equiv \mathbf{b} - \mathbf{A}\mathbf{x}_n - \mathbf{r}_n \tag{28}$$

between the true and the updated residuals will occur, and one can expect that it will somehow grow with n. This has been known for a long time, but only recently this gap was analyzed for the two most important cases, namely for two-term update formulas

$$\begin{aligned} \mathbf{r}_{n+1} &:= \mathbf{r}_n - \mathbf{A}\mathbf{v}_n\omega_n\,, \\ \mathbf{x}_{n+1} &:= \mathbf{x}_n + \mathbf{v}_n\omega_n \end{aligned} \tag{29}$$

(which need to be combined with one for the direction vectors, say, $\mathbf{v}_0 := \mathbf{r}_0$, $\mathbf{v}_n := \mathbf{r}_n + \mathbf{v}_{n-1}\psi_{n-1}$ $(n > 0)$, which has no influence on the gap) like in BiOMin and in the classical OMin version of CG, and for a pair of three-term recurrences

$$\left.\begin{aligned} \mathbf{r}_{n+1} &:= (\mathbf{A}\mathbf{r}_n - \mathbf{r}_n\alpha_n - \mathbf{r}_{n-1}\beta_{n-1})/\gamma_n\,, \\ \mathbf{x}_{n+1} &:= -(\mathbf{r}_n + \mathbf{x}_n\alpha_n + \mathbf{x}_{n-1}\beta_{n-1})/\gamma_n \end{aligned}\right\} \quad \text{with} \quad \gamma_n := -(\alpha_n + \beta_{n-1}), \tag{30}$$

like in BiORes and the corresponding ORes version of CG. (At the start, $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\mathbf{r}_{-1} := \mathbf{o}$, $\mathbf{x}_{-1} := \mathbf{o}$, $\beta_{-1} := 0$.)

The relevance of this gap is due to the fact that in most methods the updated residuals become ultimately orders of magnitude smaller than the true residuals, which essentially stagnate from a certain moment. Consequently, a large gap means low attainable accuracy: the true residuals will stagnate early.

For the two-term recurrences of the form (29) Greenbaum [19, 20] proved the following result (which improves a similar one of Sleijpen, van der Vorst, and Fokkema [42]):

Theorem 4.1 *Assume iterates and residuals are updated according to* (29). *Then the gap* (28) *between the true and the updated residual is given by*

$$\mathbf{f}_n = \mathbf{f}_0 - \mathbf{l}_0 - \cdots - \mathbf{l}_n\,, \tag{31}$$

where

$$\mathbf{l}_n :\equiv \mathbf{A}\mathbf{h}_n + \mathbf{g}_n \tag{32}$$

is the local error whose components $\mathbf{h}_n$ *and* $\mathbf{g}_n$ *are defined by*

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_n\omega_n + \mathbf{h}_n\,, \qquad \mathbf{r}_{n+1} = \mathbf{r}_n - \mathbf{A}\mathbf{v}_n\omega_n + \mathbf{g}_n\,. \tag{33}$$

In particular,

$$\frac{||\mathbf{f}_n||}{||\mathbf{A}||\,||\mathbf{x}||} \leq (\epsilon + \mathcal{O}(\epsilon^2))\,[n + 2 + (1 + \mu + (n+1)(10 + 2\mu))\Theta_n]\;, \tag{34}$$

where ϵ *denotes the machine-epsilon,* $\mu :\equiv m\sqrt{N}$ *with* m *the maximum number of nonzeros in a row of* $\mathbf{A}$ *and* N *the matrix order, and*

$$\Theta_n :\equiv \max_{k\leq n} \frac{||\mathbf{x}_k||}{||\mathbf{x}||}. \tag{35}$$

In contrast, for a pair of three-term recurrences (30) the following holds [29]:

Theorem 4.2 *Assume iterates and residuals are updated according to* (30)*. Then the gap* (28) *satisfies, up to* $\mathcal{O}(\epsilon^2)$*,*

$$\begin{array}{rcl} \mathbf{f}_{n+1} = \mathbf{f}_0 & - & \mathbf{l}_0 \\ & - & \mathbf{l}_0\dfrac{\beta_0}{\gamma_1} - \mathbf{l}_1 \\ & - & \mathbf{l}_0\dfrac{\beta_0\beta_1}{\gamma_1\gamma_2} - \mathbf{l}_1\dfrac{\beta_1}{\gamma_2} - \mathbf{l}_2 \\ & & \vdots \\ & - & \mathbf{l}_0\dfrac{\beta_0\beta_1\cdots\beta_{n-1}}{\gamma_1\gamma_2\cdots\gamma_n} - \ldots - \mathbf{l}_{n-1}\dfrac{\beta_{n-1}}{\gamma_n} - \mathbf{l}_n\,, \end{array} \tag{36}$$

where

$$\mathbf{l}_n :\equiv (-\mathbf{b}\varepsilon_n + \mathbf{A}\mathbf{h}_n + \mathbf{g}_n)/\gamma_n \tag{37}$$

is the local error whose components $\mathbf{h}_n$*,* $\mathbf{g}_n$*, and* ε_n *are defined by*

$$\begin{array}{rcl} \mathbf{r}_{n+1} & = & (\mathbf{A}\mathbf{r}_n - \mathbf{r}_n\alpha_n - \mathbf{r}_{n-1}\beta_{n-1} + \mathbf{g}_n)/\gamma_n\,, \\ \mathbf{x}_{n+1} & = & -(\mathbf{r}_n + \mathbf{x}_n\alpha_n + \mathbf{x}_{n-1}\beta_{n-1} + \mathbf{h}_n)/\gamma_n\,, \\ \gamma_n & = & -(\alpha_n + \beta_{n-1} + \varepsilon_n). \end{array} \tag{38}$$

It is rather easy to derive from the definitions of the local errors, that is, from (32), (33) and (37), (38), respectively, bounds for these local errors. They show that typically the local errors in algorithms based on three-term updates are larger than those arising in two-term updates. In the former case, (34) and (35) show that the size of the gap mainly depends on the norm of the largest iterate. In the latter case, the largest residual norm also has a direct influence, and the constants in the estimate are larger.

However, the main difference between Theorems 4.1 and 4.2 lies in the explicit formulas (31) and (36) for the gaps: while in the two-term case the gap $\mathbf{f}_n$ is just a sum of local errors $\mathbf{l}_j$, in the three-term case the (normally larger) local errors are multiplied (and thus amplified) by potentially very large factors. So, the gap is typically much bigger in the latter case. This leads to an explanation of the fact that the attainable accuracy, that is, the level on which the true residuals stagnate, is much worse for a three-term based algorithm than for one using two-term updates. This fact can be easily verified numerically in examples where the residual norm fluctuates heavily, a quite common behavior when $\mathbf{A}$ is ill-conditioned. This behavior is more likely in BiCG and other Lanczos-type methods for non-Hermitian systems than in CG, but it can also occur in CG, and, actually, even small CG examples can be constructed to illustrate it; see [29].

These investigations are easily adapted to other methods, including, *e.g.*, (Bi)CGS, where also Theorem 4.1 applies. Consequently, for (Bi)CGS the gap is not as bad as one might expect from the very erratic convergence behavior, because the local errors (which may be large due to high peaks in the residual norm history) are not amplified by large factors. However, the peaks may be so high that neither the updated nor the true residual converge.

A fairly general remedy against the growth of the gap between true and updated residuals — and thus against the corresponding loss of attainable accuracy — is based on an idea of Neumaier [35]: *occasional synchronization of true and updated residual combined with a shift of origin.* Neumaier's proposal is just a variation of using true instead of recursive residuals. He suggested to compute in (Bi)CGS the true residual at every step where the residual norm is reduced and, at the same time, to replace the current system by one for the remaining correction $\delta\mathbf{x}$ in $\mathbf{x}$, so that the current residual becomes the new right-hand side. One can think of this as a repeated shift of the origin or an *implicit iterative refinement.* At the beginning we let

$$\mathbf{b}' := \mathbf{b} - \mathbf{A}\mathbf{x}_0\,, \qquad \mathbf{x}' := \mathbf{x}_0\,, \qquad \mathbf{x}_0 := \mathbf{o}\,,$$

so that $\mathbf{b} - \mathbf{A}\mathbf{x} = \mathbf{b}' - \mathbf{A}\,\delta\mathbf{x}$, where $\delta\mathbf{x} := \mathbf{x} - \mathbf{x}'$. We then apply our algorithm of choice to $\mathbf{A}\,\delta\mathbf{x} = \mathbf{b}'$. At step n, if the *update condition*

$$||\mathbf{r}_n|| < ||\mathbf{b}'||\,\gamma' \qquad \text{(where} \quad \gamma' \in (0, 1] \quad \text{is given)} \tag{39}$$

is satisfied, we include the reassignments

$$\mathbf{b}' := \mathbf{b}' - \mathbf{A}\mathbf{x}_n\,, \qquad \mathbf{x}' := \mathbf{x}' + \mathbf{x}_n\,, \qquad \mathbf{x}_n := \mathbf{o}\,. \tag{40}$$

Note that at every step, we then have

$$\mathbf{r}_n = \mathbf{b}' - \mathbf{A}\mathbf{x}_n = \mathbf{b} - \mathbf{A}(\mathbf{x}' + \mathbf{x}_n)\,.$$

Neumaier actually computed the true residual at every step and chose $\gamma' = 1$, which means that the update is performed at every step where the residual decreases, hence, nearly always. Sleijpen and van der Vorst [41] followed up on this idea and suggested several alternatives to the update condition (39), so that fewer shifts and true residuals are used. Recently, van der Vorst and Ye [46] came up with yet another improvement of this strategy. It pushes the level of stagnation of the true residual down to the level that has to be expected in view of the roundoff bounds for the evaluation of the residual at the rounded exact solution in finite precision arithmetic. In general, each update (40) requires an extra matrix-vector product. However, Neumaier [35] found a way to use it in (BI)CGS for replacing one of the two other such products, and Sleijpen and van der Vorst [41] achieved the same for BICGSTAB.

Acknowledgment. The author would like to express sincere thanks to Andreas Frommer and Miroslav Rozložník for their helpful comments.

References

[1] A. Boriçi. *Krylov Subspace Methods in Lattice QCD.* Diss. ETH, Swiss Federal Institute of Technology (ETH) Zurich, 1996.

[2] A. Boriçi and P. de Forcrand. Fast Krylov space methods for calculation of quark propagators. IPS Research Report 94-03, ETH Zurich, 1994.

[3] G. Cella, A. Hoferichter, V. K. Mitrjushkin, M. Müller-Preuss, and A. Vincere. Efficiency of different matrix inversion methods applied to Wilson fermions. Technical Report HU Berlin–EP-96/17, IFUP–TH 29/96, SWAT/96/108, 1996.

[4] J. K. Cullum and R. A. Willoughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations (2 Vols.).* Birkhäuser, Boston-Basel-Stuttgart, 1985.

[5] D. Day. An efficient implementation of the non-symmetric Lanczos algorithm. *SIAM J. Matrix Anal. Appl.*, 18:566–589, 1997.

[6] D. M. Day III. *Semi-duality in the two-sided Lanczos algorithm.* PhD thesis, University of California at Berkeley, 1993.

[7] P. Fiebach, R. W. Freund, and A. Frommer. Variants of the block-QMR method and applications in quantum chromodynamics. In A. Sydow, editor, *15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics, Vol. 3, Computational Physics, Chemistry and Biology*, pages 491–496. Wissenschaft und Technik Verlag, 1997.

[8] R. Fletcher. Conjugate gradient methods for indefinite systems. In G. A. Watson, editor, *Numerical Analysis, Dundee, 1975*, volume 506 of *Lecture Notes in Mathematics*, pages 73–89. Springer, Berlin, 1976.

[9] R. W. Freund. Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices. *SIAM J. Sci. Statist. Comput.*, 13:425–448, 1992.

[10] R. W. Freund. Lanczos-type algorithms for structured non-Hermitian eigenvalue problems. In J. D. Brown, M. T. Chu, D. C. Ellison, and R. J. Plemmons, editors, *Proceedings of the Cornelius Lanczos International Centenary Conference*, pages 243–245. SIAM, Philadelphia, PA, 1994.

[11] 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.*, 14:137–158, 1993.

[12] R. W. Freund and N. M. Nachtigal. An implementation of the QMR method based on coupled two-term recurrences. *SIAM J. Sci. Comput.*, 15:313–337, 1994.

[13] R. W. Freund and N. M. Nachtigal. Software for simplified Lanczos and QMR algorithms. *Applied Numerical Mathematics*, 19:319–341, 1995.

[14] A. Frommer. Linear system solvers — recent developments and implications for lattice computations. *Nuclear Physics* **B** *(Proc. Suppl.)*, 53:120–126, 1996.

[15] A. Frommer, V. Hannemann, B. Nöckel, T. Lippert, and K. Schilling. Accelerating Wilson fermion matrix inversions by means of the stabilized biconjugate gradient algorithm. *Int. J. Modern Physics C*, 5:1073–1088, 1994.

[16] A. Frommer and B. Medeke. Exploiting structure in Krylov subspace methods for the Wilson fermion matrix. In A. Sydow, editor, *15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics, Vol. 3, Computational Physics, Chemistry and Biology*, pages 485–490. Wissenschaft und Technik Verlag, 1997.

[17] A. Frommer, B. Nöckel, S. Güsken, T. Lippert, and K. Schilling. Many masses on one stroke: economic computation of quark propagators. *Int. J. Modern Physics C*, 6:627–638, 1995.

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

[19] A. Greenbaum. Accuracy of computed solutions from conjugate-gradient-like methods. In M. Natori and T. Nodera, editors, *Advances in Numerical Methods for Large Sparse Sets of Linear Systems*, number 10 in Parallel Processing for Scientific Computing, pages 126–138. Keio University, Yokahama, Japan, 1994.

[20] A. Greenbaum. Estimating the attainable accuracy of recursively computed residual methods. *SIAM J. Matrix Anal. Appl.*, 18(3):535–551, 1997.

[21] A. Greenbaum and Z. Strakoš. Predicting the behavior of finite precision Lanczos and conjugate gradient computations. *SIAM J. Matrix Anal. Appl.*, 13(1):121–137, 1992.

[22] M. H. Gutknecht. Solving Theodorsen's integral equation for conformal maps with the fast fourier transform and various nonlinear iterative method. *Numer. Math.*, 36:405–429, 1981.

[23] M. H. Gutknecht. The unsymmetric Lanczos algorithms and their relations to Padé approximation, continued fractions, and the qd algorithm. in Preliminary Proceedings of the Copper Mountain Conference on Iterative Methods,

April 1990. `http://www.sam.math.ethz.ch/~mhg/pub/CopperMtn90.ps.Z` and `CopperMtn90-7.ps.Z`.

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

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

[26] M. H. Gutknecht. A completed theory of the unsymmetric Lanczos process and related algorithms, Part II. *SIAM J. Matrix Anal. Appl.*, 15:15–58, 1994.

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

[28] M. H. Gutknecht and K. J. Ressel. Look-ahead procedures for Lanczos-type product methods based on three-term recurrences. Tech. Report TR-96-19, Swiss Center for Scientific Computing, June 1996.

[29] M. H. Gutknecht and Z. Strakoš. Accuracy of two three-term and three two-term recurrences for Krylov space solvers. *SIAM J. Matrix Anal. Appl.* To appear.

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

[31] M. Hochbruck. The Padé table and its relation to certain numerical algorithms. Habilitationsschrift, Universität Tübingen, Germany, 1996.

[32] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, New York, 1985.

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

[34] T. A. Manteuffel. The Tchebyshev iteration for nonsymmetric linear systems. *Numer. Math.*, 28:307–327, 1977.

[35] A. Neumaier. Iterative regularization for large-scale ill-conditioned linear systems. Talk at Oberwolfach, April 1994.

[36] B. N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, N.J., 1980.

[37] B. N. Parlett and D. S. Scott. The Lanczos algorithm with selective reorthogonalization. *Math. Comp.*, 33:217–238, 1979.

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

[39] H. Rutishauser. Beiträge zur Kenntnis des Biorthogonalisierungs-Algorithmus von Lanczos. *Z. Angew. Math. Phys.*, 4:35–56, 1953.

[40] Y. Saad. The Lanczos biorthogonalization algorithm and other oblique projection methods for solving large unsymmetric systems. *SIAM J. Numer. Anal.*, 2:485–506, 1982.

[41] G. L. G. Sleijpen and H. A. van der Vorst. Reliable updated residuals in hybrid Bi-CG methods. *Computing*, 56:141–163, 1996.

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

[43] G. L. G. Sleijpen, H. A. van der Vorst, and J. Modersitzki. Effects of rounding errors in determining approximate solutions in Krylov solvers for symmetric linear systems. Preprint 1006, Department of Mathematics, Universiteit Utrecht, March 1997.

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

[45] 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.*, 13:631–644, 1992.

[46] H. A. van der Vorst and Q. Ye. Residual replacement strategies for Krylov subspace iterative methods for the convergence of true residuals. Preprint, 1999.

[47] R. S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, N.J., 1962. Rev. 2nd ed., Springer-Verlag, 1999.

[48] H. E. Wrigley. Accelerating the Jacobi method for solving simultaneous equations by Chebyshev extrapolation when the eigenvalues of the iteration matrix are complex. *Comput. J.*, 6:169–176, 1963.

[49] S.-L. Zhang. GPBI-CG: generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 18(2):537–551, 1997.

[50] H. Zurmühl. *Matrizen und ihre technischen Anwendungen, 4. Aufl.* Springer-Verlag, Berlin, 1994.

Research Reports

No.	Authors	Title
00-03	M.H. Gutknecht	On Lanczos-type methods for Wilson fermions
00-02	R. Sperb, R. Strebel	An alternative to Ewald sums. Part 3: Implementation and results
00-01	T. Werder, K. Gerdes, D. Schötzau, C. Schwab	*hp* Discontinuous Galerkin Time Stepping for Parabolic Problems
99-26	J. Waldvogel	Jost Bürgi and the Discovery of the Logarithms
99-25	H. Brunner, Q. Hu, Q. Lin	Geometric meshes in collocation methods for Volterra integral equations with proportional time delays
99-24	D. Schötzau, Schwab	An *hp* a-priori error analysis of the DG time-stepping method for initial value problems
99-23	R. Sperb	Optimal sub- or supersolutions in reaction-diffusion problems
99-22	M.H. Gutknecht, M. Rozložník	Residual smoothing techniques: do they improve the limiting accuracy of iterative solvers?
99-21	M.H. Gutknecht, Z. Strakoš	Accuracy of Two Three-term and Three Two-term Recurrences for Krylov Space Solvers
99-20	M.H. Gutknecht, K.J. Ressel	Look-Ahead Procedures for Lanczos-Type Product Methods Based on Three-Term Lanczos Recurrences
99-19	M. Grote	Nonreflecting Boundary Conditions For Elastodynamic Scattering
99-18	J. Pitkäranta, A.-M. Matache, C. Schwab	Fourier mode analysis of layers in shallow shell deformations
99-17	K. Gerdes, J.M. Melenk, D. Schötzau, C. Schwab	The *hp*-Version of the Streamline Diffusion Finite Element Method in Two Space Dimensions
99-16	R. Klees, M. van Gelderen, C. Lage, C. Schwab	Fast numerical solution of the linearized Molodensky problem
99-15	J.M. Melenk, K. Gerdes, C. Schwab	Fully Discrete *hp*-Finite Elements: Fast Quadrature
99-14	E. Süli, P. Houston, C. Schwab	*hp*-Finite Element Methods for Hyperbolic Problems
99-13	E. Süli, C. Schwab, P. Houston	*hp*-DGFEM for Partial Differential Equations with Nonnegative Characteristic Form
99-12	K. Nipp	Numerical integration of differential algebraic systems and invariant manifolds
99-11	C. Lage, C. Schwab	Advanced boundary element algorithms