Peter Deuflhard

A Study of Lanczos–Type Iterations for Symmetric Indefinite Linear Systems

Preprint SC 93–6 (March 1993)
Abstract

The Lanczos iteration for symmetric indefinite linear systems seems to be well-known for quite a while. However, in order to modify it with the aim of improved performance, the present paper studies certain aspects in terms of an adjoint scalar three-term recurrence. Thus, at least a different view is opened. Moreover, an alternative 3n-implementation in terms of the Euclidean orthogonal basis has been found that easily permits generalizations. The study is understood as a start-off for further numerical investigations and experiments.
0. Introduction

This paper deals with the numerical solution of linear \( n \)-systems

\[
Ax = b, \quad x, b \in \mathbb{R}
\]

where \( A \) is a symmetric \((n, n)\)-matrix, in general indefinite. For sufficiently large systems, an iterative method will be the method of choice. Since \( A \) is symmetric, an orthogonal basis of \( \mathbb{R}^n \) can be constructed via three-term recurrences — an observation first exploited by C. Lanczos [7]. As a consequence, compact algorithms with fixed array storage \( q \cdot n, q = O(1) \), can be designed. However, utmost care must be taken to achieve numerical stability. Even the algorithm originally proposed by Lanczos turned out to be unstable. The most successful candidate among the stable algorithms is the conjugate gradient method, which, for positive definite \( A \), minimizes the energy norm of the error in each step — a concept, however, which does not carry over to the general indefinite case. An extension to the indefinite case, aiming at minimization of the Euclidean norm of the error, has been proposed by Fridman [5]. Unfortunately, this algorithm is also numerically unstable. A numerically stable alternative using Givens rotations has been designed by Paige and Saunders [8] in a classical numerical linear algebra framework.

The purpose of the present paper is to revisit the topic mainly from the side of its recursive structure without fixing the choice of norm too early. Thus the iterative features of a whole class of algorithms can be viewed in a more general framework — such as compact storage realization, iterative convergence and numerically stable implementation. In Section 1, a few simple technical results are exposed, which may nevertheless not be folklore. On this basis, construction principles for the design of algorithms are discussed — such as approximate Krylov subspace error minimization, error reduction or residual orthogonality. Compact implementations and their numerical stability properties are discussed in Section 3 — together with a \( 3n \)-implementation that permits blending of the Lanczos iteration with other possible iterations.

1. Basic Recursive Structure

Given an initial guess \( x_0 \) of the unknown solution \( x \), let

\[
\Delta x := x - x_0
\]

denote the correction to be computed. Given additionally the initial residual

\[
r_0 := b - Ax_0,
\]
then the symmetric indefinite linear system to be solved arises as
\[ A\Delta x = r_0 , \ A^T = A . \]  
(1.1)

We assume that \( A \) is nonsingular (and will test for this hypothesis in the course of computation by means of the algorithms to be studied). For a given inner product \( \langle \cdot, \cdot \rangle \), which will be fixed later, let \( \{v_j\} \) define an orthogonal basis such that
\[ \langle v_i, v_j \rangle = \sigma_j \delta_{ij} . \]  
(1.2)

We dispose about normalization such that
\[ v_j = p_j(A)r_0 = A^{j-1}r_0 + \ldots \]
in terms of a sequence of polynomials \( \{p_j\} \) with leading coefficient 1. Then the sequence \( \{v_j\} \) can be computed recursively via Schmidt orthogonalization
\[ v_1 = r_0 , \ v_2 = Av_1 - \alpha_1 v_1 \]
\[ v_{k+1} = Av_k - \alpha_k v_k - \gamma_k^2 v_{k-1} \quad k = 2, \ldots, n - 1 \]  
(1.3)

where
\[ \alpha_k = \langle v_k, Av_k \rangle / \sigma_k , \ \gamma_k^2 = \sigma_k / \sigma_{k-1} . \]
The \( \{v_1, \ldots, v_k\} \) span a Krylov subspace
\[ \mathcal{K}_k(A; r_0) := \{ r_0, Ar_0, \ldots, A^{k-1}r_0 \} . \]

Let
\[ \overline{v}_j := v_j / \sqrt{\sigma_j} \quad j = 1, \ldots, n \]
denote the associated orthonormal basis. In order to introduce matrix notation, let \( \overline{V}_k := [\overline{v}_1, \ldots, \overline{v}_k] \) denote orthogonal \((n, k)\)-matrices such that
\[ \overline{V}_k^T \overline{V}_k = I_k \quad k = 1, \ldots, n \]
and
\[ \overline{V}_n^T \overline{V}_n = I_n . \]

In this notation, (1.3) can be written as
\[ A\overline{V}_k = \overline{V}_k \overline{T}_k + \gamma_{k+1} \overline{v}_{k+1} e_k^T \]  
(1.4)
in terms of the symmetric tridiagonal \((k, k)\)-matrices
\[ \overline{T}_k = \overline{V}_k^T A \overline{V}_k , \quad \overline{T}_k = \begin{bmatrix} \alpha_1 & \gamma_2 & 0 & \cdots & 0 \\ \gamma_2 & \ddots & \ddots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \gamma_k \\ 0 & \cdots & \cdots & \gamma_k & \alpha_k \end{bmatrix} \]  
(1.5)
With the decomposition (1.4) for \( k = n \), the original system (1.1) to be solved now arises as
\[
T_n \xi^n = \|r_0\| e_1, \quad \xi^n := \nabla^T_n \Delta x .
\] (1.6)

In componentwise notation
\[
\xi_j := \langle v_j, \Delta x \rangle \quad j = 1, \ldots, n
\] (1.7)
we thus obtain the solution in the form
\[
\Delta x = \sum_{j=1}^n \xi_j v_j .
\] (1.8)

As for the formal solution of (1.6), we apply \textit{rational Cholesky decomposition}
\[
T_k = B_k D_k B_k^T \quad k = 1, \ldots, n
\] (1.9)
with
\[
B_k = \begin{bmatrix}
1 & & \\
& b_2 & \ddots \\
& \ddots & \ddots \\
& & & b_k \\
& & & & 1
\end{bmatrix}, \quad D_k = \text{diag}(d_1, \ldots, d_k).
\]

Comparing entries of \( B_k \), \( D_k \) and \( T_k \) then yields the forward recursions
\[
d_1 = \alpha_1, \quad d_{j+1} = \alpha_{j+1} - \gamma^2_{j+1}/d_j \quad j = 1, \ldots, n - 1
\] (1.10)
and
\[
b_{j+1} = \gamma_{j+1}/d_j \quad j = 1, \ldots, n - 1 .
\]

By Sylvester’s theorem of inertia [6], the sign distribution of the sequence \( \{d_j\} \) equals the one of the eigenvalues of \( A \). In particular, since \( A \) is assumed to be nonsingular, none of the \( d_1, \ldots, d_n \) will vanish. On the other hand, if any of the \( d_j \) vanishes in the course of computation, then \( A \) must be singular. For this reason, utmost care must be taken, whenever these \( d_j \) are used in actual computation, to treat the “nearly singular” case.

Upon rechanging normalization from \( v_j \) to \( v_j \), the coefficients \( \xi_j \) will change to
\[
\xi_j = \frac{\langle v_j, \Delta x \rangle}{\langle v_j, v_j \rangle} = \frac{\langle v_j, \Delta x \rangle}{\sigma_j} \quad j = 1, \ldots, n
\]
in terms of the expansion
\[
\Delta x = \sum_{j=1}^n \xi_j v_j .
\] (1.11)
Accordingly, the tridiagonal system arises as

\[ T_n \xi^n = e_1 \quad (1.12) \]

with

\[
T_n := \begin{bmatrix}
\alpha_1 & \gamma_2 & & & & \\
1 & \ddots & \ddots & & & \\
& \ddots & \ddots & \ddots & & \\
& & \ddots & \ddots & \ddots & \\
& & & 1 & \alpha_n & \\
\end{bmatrix}.
\]

In order to construct an iterative method, define

\[ \Delta x_k := x_k - x_0, \ r_k := b - Ax_k, \ k = 0, \ldots, n, \quad (1.13) \]

which implies the iterative updates

\[ x_k = x_{k-1} + \xi_k v_k, \ r_k = r_{k-1} - \xi_k A v_k \quad k = 1, \ldots, n. \quad (1.14) \]

Note that

\[ \Delta x_n = \Delta x, \ r_n = 0. \quad (1.15) \]

Obviously, once the coefficients \( \{\xi_j\} \) can be computed recursively, the updates (1.14) can be performed easily.

**Lemma 1** Let \( r_0 \neq 0 \). In the notation just introduced, the coefficients \( \{\xi_j\} \) satisfy the (adjoint) three-term recurrence

\[ \xi_{j-1} + \alpha_j \xi_j + \gamma_{j+1}^2 \xi_{j+1} = 0 \quad (1.16) \]

with \( \xi_0 = -1. \) At most one component \( \xi_q \) of \( \{\xi_j\} \) can vanish. Moreover, the iterative residuals can be represented as

\[ r_k = \xi_{k+1} \gamma_{k+1}^2 v_k - \xi_k v_{k+1} \quad k = 0, \ldots, n - 1. \quad (1.17) \]

**Proof.** In order to show (1.16) above, just recall (1.12)

\[ T_n \xi^n = e_1 = (1, 0, \ldots, 0)^T. \]

Herein, rows 2 up to \((n - 1)\) anyway represent (1.16). Row 1 is included by defining \( \xi_0 = -1. \) Row \( n \) is included, since \( \sigma_{n+1} = \langle v_{n+1}, v_{n+1} \rangle = 0 \), which implies \( \gamma_{n+1}^2 = \sigma_{n+1}/\sigma_n = 0 \), so that the last row degenerates.

With the homogeneous three-term recurrence (1.16) valid, no more than one single component \( \xi_q \) is allowed to vanish — otherwise all components must vanish, which is a contradiction to the condition \( r_0 \neq 0. \)
Finally, the representation for the residuals is easily obtained:

\[ r_k = r_0 - A \Delta x_k = v_1 - \sum_{j=1}^{k-1} \xi_j A v_j \]
\[ = - \sum_{j=1}^{k-1} (\xi_{j-1} + \xi_j \alpha_j + \xi_{j+1} \gamma_{j+1}^2) v_j - (\xi_{k+1} + \xi_k \alpha_k) v_k - \xi_k v_{k+1} \]
\[ = \xi_{k+1} \gamma_{k+1}^2 v_k - \xi_k v_{k+1} . \]

With \( \xi_0 = -1 \) and the recurrence (1.16) given, a forward recursion for \( \Delta x_k, k = 1, 2, \ldots \) can be principally constructed, once the coefficient

\[ \xi_1 = \frac{\langle r_0, \Delta x \rangle}{\langle r_0, r_0 \rangle} = \frac{\langle v_1, \Delta x \rangle}{\sigma_1} \] (1.18)

is computationally available. For the Euclidean inner product, (1.18) is just the definition, which cannot be evaluated, unless the solution \( \Delta x \) is already known. If, however, \( A \) is not only symmetric but also positive definite, then \( \langle \cdot, \cdot \rangle \) can be specified as the well–known energy product

\[ \langle u, v \rangle_A := \langle u, Av \rangle . \] (1.19)

As a consequence, one then obtains

\[ \xi_1 = \frac{\langle r_0, \Delta x \rangle_A}{\langle r_0, r_0 \rangle_A} = \frac{\langle v_1, v_1 \rangle}{\langle v_1, Av_1 \rangle} = \frac{\sigma_1}{\langle v_1, v_2 + \alpha_1 v_1 \rangle} = \frac{1}{\alpha_1} . \] (1.20)

This property leads to the conjugate gradient method.

Assume, however, that the inner product is specified such that \( \xi_1 \) cannot be directly evaluated. Then the dependence of the iterative solutions \( \Delta x_k \) on \( \xi_1 \) is of interest.

**Lemma 2** Let \( \{ \eta_j \}, \{ \zeta_j \} \) denote two solutions of the adjoint three–term recurrence (1.16) with starting values

\[ \eta_0 = -1 \, , \, \eta_1 = 0 \, \text{ and } \, \zeta_0 = 0 \, , \zeta_1 = 1 . \] (1.21)

Then the Krylov subspace solution \( \Delta x_k \) can be represented as

\[ \Delta x_k = \sum_{j=1}^{k} \xi_j v_j \, , \, \xi_j = \eta_j + \xi_1 \cdot \zeta_j \] (1.22)

or, equivalently

\[ \Delta x_k = \Delta y_k + \xi_1 \Delta z_k \]

5
in term of
\[ \Delta y_k = \sum_{j=1}^{k} \eta_j v_j, \quad \Delta z_k = \sum_{j=1}^{k} \zeta_j v_j. \]
The sequences \( \{\eta_j\}, \{\zeta_j\} \) or, equivalently, the corrections \( \Delta y_k, \Delta z_k, k = 1, \ldots, n \) are linearly independent. Moreover, one has
\[ \eta_j \neq 0, \quad \zeta_j \neq 0 \quad j = 2, \ldots, n \] (1.23)
In particular, the \( \zeta \)-sequence can be obtained from
\[ \zeta_{k+1} = -\frac{\sigma_k}{\sigma_{k+1}} d_k \zeta_k, \quad k = 1, \ldots, n-1, \] (1.24)
with\( \{d_k\} \) as defined in (1.10).

**Proof.** The adjoint three-term recurrence for the coefficients \( \xi_j \) has two linearly independent solutions. With \( \xi_0 = -1 \) given and \( \xi_1 \) to be used as a parameter, the selection of the solutions \( \{\eta_j\}, \{\zeta_j\} \) by \( \eta_0 = -1, \eta_1 = 0 \) and \( \zeta_0 = 0, \zeta_1 = 1 \) is natural. Both sequences already have one vanishing element, which implies that a further element cannot vanish, unless the whole sequence vanishes — which would be a contradiction. As for the linear independence, let
\[ D(k, k+1) := \eta_k \zeta_{k+1} - \zeta_k \eta_{k+1} \] (1.25)
denote the associated Casorati determinant — compare e.g. [2] or [3]. Then insertion of (1.16) yields
\[ \gamma_{k+1}^\sigma D(k, k+1) = -\eta_k (\zeta_{k-1} + \alpha_k \zeta_k) + \zeta_k (\eta_{k-1} + \alpha_k \eta_k) = D(k-1, k). \]
Hence, with \( \gamma_{k+1}^\sigma = \sigma_{k+1}/\sigma_k \), the invariance relation
\[ \sigma_{k+1} D(k, k+1) = \ldots = \sigma_1 D(0, 1) = -\sigma_1 \] (1.26)
is directly obtained. As a consequence
\[ D(k, k+1) \neq 0 \quad k = 0, \ldots, n-1. \]
which implies linear independence of the sequences \( \{\eta_j\}, \{\zeta_j\} \) and the corrections \( \Delta y_k, \Delta z_k \) for all \( k = 1, \ldots, n \). Finally, with (1.16) for \( \{\zeta_j\} \), i.e.
\[ \zeta_{j-1} + \alpha_j \zeta_j + \gamma_j^\sigma \zeta_{j+1} = 0 \quad j = 1, \ldots, n-1 \]
and \( \zeta_j \neq 0 \) for \( j = 1, \ldots, n \), we obtain
\[ \frac{\zeta_{j-1}}{\zeta_j} + \alpha_j + \frac{\sigma_{j+1} \zeta_{j+1}}{\sigma_j \zeta_j} = 0, \quad j = 1, \ldots, n-1. \]
Upon comparing this continued fraction representation with the rational recursion (1.10), one immediately observes equality of these recursions for the choice

\[ d_j = -\frac{\sigma_j + 1}{\sigma_j} \]

and, in addition, consistence of the starting values, since

\[ \alpha_1 = d_1 = -\frac{\gamma_2 \zeta_2}{\zeta_0} = \frac{\zeta_0 + \alpha_1 \zeta_0}{\zeta_0} \].

\[ \blacksquare \]

2. Algorithm Design Principles

The problem stated in the preceding section is the efficient recursive computation of the Krylov subspace solutions

\[ T_k \xi^k = e_1, \quad \Delta x_k = V_k \xi^k \] (2.1)

for increasing index \( k \) with the tridiagonal matrices

\[ T_k = \begin{bmatrix}
\alpha_1 & \gamma_2^0 \\
1 & \ddots & \ddots \\
& \ddots & \ddots & \gamma_k^0 \\
& & 1 & \alpha_k
\end{bmatrix} \]

wherein

\[ \sigma_j := \langle v_j, v_j \rangle, \quad \alpha_j := \langle v_j, Av_j \rangle / \sigma_j. \]

For \( A \) symmetric and positive definite, which implies the same for \( T_k \), the corrections \( \Delta x_k \) are just those obtained from the conjugate gradient method. Following [8], this can be seen (in our notation) starting from (1.6) and (1.26):

\[ \Delta x_k = \nabla_k \bar{\xi}^k = \nabla_k T_k^{-1} \| r_0 \| e_1 = \nabla_k B_k^{-T} D_k^{-1} B_k^{-T} \| r_0 \| e_1. \]

At this point, a splitting into pure forward recursions is done by introducing a new basis \( \{ \overline{p}_j \} \) by

\[ \nabla_k B_k^{-T} =: \overline{p}_k = [\overline{p}_1, \ldots, \overline{p}_k]. \]

With \( b_j = \gamma_j / d_{j-1}, \ j = 2, \ldots, n \), this can be written as

\[ \overline{v}_1 = \overline{p}_1 = r_0 / \| r_0 \|, \quad \overline{v}_j = \frac{\gamma_j}{d_{j-1}} \overline{p}_{j-1} + \overline{p}_j, \quad j = 2, \ldots, n \] (2.2)
For this basis, we obtain
\[ P_k^T A P_k = B_k^{-1} V_k^T A V_k B_k^{-T} = D_k , \]
which means that the \( \{ p_j \} \) are \( A \)-orthogonal and
\[ d_j = \langle p_j, A p_j \rangle > 0 , \quad j = 1, \ldots, n . \]
(2.4)
So the energy product \( \langle \cdot, \cdot \rangle_A = \langle \cdot, A \cdot \rangle \) is the appropriate inner product. Then the associated coefficients, say \( \xi_k = (\xi_1, \ldots, \xi_k) \), can be obtained from
\[ B_k D_k \xi_k = \| r_0 \| e_1 \]
by pure forward recursion.
For \( A \) symmetric indefinite, however, we still have \( d_j \neq 0 \) for all \( j \), but an energy product can no longer be defined. Rather, the Euclidean inner product or other suitable choices need to be investigated. For this reason, various principles for constructing such algorithms are now discussed in the light of the results of Section 1.

### 2.1 Iterative Error Minimization

For a given iterate \( x_k = x_0 + \Delta x_k \), let
\[ \varepsilon_k := \langle x_k - x, x_k - x \rangle^{1/2} = \| x_k - x \| \]
(2.6)
denote the error in the norm induced by the given inner product \( \langle \cdot, \cdot \rangle \). Let this inner product be such that \( \xi_i \) can be actually evaluated. Then (2.1) leads to the well-known Galerkin condition
\[ \langle \Delta x_k, x_k - x \rangle = 0 , \]
(2.7)
which then directly induces the minimization property
\[ \| x_k - x \| = \min_{y \in K_k} \| y - x \| \]
(2.8)
and the associated reduction property
\[ \varepsilon_{k+1}^2 = \varepsilon_k^2 - \| x_{k+1} - x_k \|^2 \leq \varepsilon_k^2 . \]
(2.9)
As a consequence of (2.8), the ultimate solution is obtained at the final step, i.e.
\[ x_n = x . \]
(2.10)
For \( A \) positive definite, this situation can be constructed by choosing the energy product
\[ \langle \cdot, \cdot \rangle := \langle \cdot, A \cdot \rangle \]
(2.11)
as inner product. As a consequence, $\sqrt{\kappa_2(A)}$ with $\kappa_2(A)$ the spectral condition number of $A$ governs the convergence behavior of the iterates. Unfortunately, for $A$ indefinite, (2.11) would no longer define an inner product. Therefore, even though the formal extension of the cg-method can be defined, its convergence properties are no longer satisfactory — and, in addition, its traditional implementation may just fail at intermediate steps. In this situation, FRIDMAN [5] had suggested to modify the Krylov subspaces by choosing

$$v_i = Ar_0$$

which then implies that

$$\xi_i = \langle v_i, \Delta x \rangle / \sigma_i = \langle r_0, A \Delta x \rangle / \sigma_i = 1 .$$

Upon looking back to the proof of Lemma 1, it can be seen that the above choice means a switch to the normal equations

$$A^2 \Delta x = Ar_0$$

and, as a consequence, to Krylov subspaces of the form $K_k(A^2; Ar_0)$. This is not what is wanted here. Hence, other possibilities need to be discussed.

### 2.2 Truncated Coefficient Approximation

The short-hand notation (2.1) hides the fact that the components of the vectors $\xi^k$ change fully with each increase of $k$. In componentwise notation, we have to write

$$\Delta x_k = \sum_{j=1}^k \xi^k_j v_j .$$

At this point, PAIGE and SAUNDERS [8] decided to exploit the tridiagonal matrices $T_k$ by Givens transformations from the right — thus reducing the entries to be changed. Here, however, we will resort to Lemma 2 above. From this, the coefficients $\xi^k_j$ can be seen to have a fairly simple structure.

**Lemma 3** The coefficients $\xi^k_j$ as defined in (2.15) may be expressed by

$$\xi^k_j = \eta_j + \xi^k_1 \zeta_j \quad j = 1, \ldots, n$$

in terms $\{\eta_j\}$, $\{\zeta_j\}$ as defined in Lemma 2 with

$$\xi^k_1 = \frac{\eta_{k+1}}{\zeta_{k+1}} \quad k = 1, \ldots, n , \quad \xi^n_1 \equiv \xi_1 .$$

The associated residuals $r_k$ are mutually orthogonal with respect to $\langle \cdot, \cdot \rangle$ and can be represented as

$$r_k = -\hat{\xi}^k v_{k+1} \quad k = 0, \ldots, n - 1 .$$

with

$$\hat{\xi}^k := \xi^k = -\frac{\sigma_i}{\zeta_{k+1} \sigma_{k+1}} \quad k = 0, \ldots, n - 1 .$$
Proof. Let \( \xi^k := \left( \xi^k_1, \ldots, \xi^k_k \right)^T \) with \( \xi^j_k \) as in (2.15). Then \( \Delta x_k \) can be defined via
\[
T_k \xi^k = e_1. \tag{2.20}
\]
In view of Lemma 2, \( \xi^k \) is split according to
\[
\xi^k_j = \eta^k_j + \xi^k_k \cdot \zeta^k_j, \quad j = 0, \ldots, k, \quad k = 1, \ldots, n,
\]
which is just (2.16). Let \( \eta^k := (\eta_1, \ldots, \eta_k)^T, \xi^k := (\zeta_1, \ldots, \zeta_k) \), then the equivalent equations to (2.20) are easily verified to be
\[
\begin{align*}
T_k \eta^k &= e_1 - \gamma_{k+1}^2 \eta_{k+1} e_k, \tag{2.21} \\
T_k \xi^k &= -\gamma_{k+1} \eta_{k+1} e_k. \tag{2.22}
\end{align*}
\]
Now with (2.16) and the combination (2.20), (2.21), and (2.22) we have
\[
O = T_k (\eta^k + \xi^k \xi^k) - T_k \xi^k = \gamma_{k+1}^2 (\eta_{k+1} + \xi^k_k \zeta_{k+1}) e_k.
\]
From this, (2.17) follows immediately. For \( k = n \), we have \( \xi^*_n = \xi_1 \), since \( \Delta x_n = \Delta x \). Finally, the residual representation (1.17) can be employed to obtain
\[
r_k = r_0 - A \Delta x_k = r_0 - A (\Delta y_k + \xi^k_1 \cdot \Delta z_k)
= \gamma_{k+1}^2 (\eta_{k+1} + \xi^k_k \zeta_{k+1}) v_k - (\eta_k + \xi^k_k \zeta_k) v_{k+1}.
\]
In the notation above this boils down to
\[
r_k = -\hat{\xi}_k v_{k+1}, \quad \hat{\xi}_k = (\eta_k + \xi^k_k \zeta_k).
\]
Insertion of (2.17) and use of the Casorati determinant relation (1.26) then yields
\[
\hat{\xi}_k = \frac{1}{\zeta_{k+1}} (\eta_k \zeta_{k+1} - \zeta_k \eta_{k+1}) = D(k, k+1) / \zeta_{k+1}
= \sigma_1 D(0, 1) / (\zeta_{k+1} \sigma_{k+1}) = -\sigma_1 / (\zeta_{k+1} \sigma_{k+1}).
\]
This completes the proof of the lemma. \( \Box \)

Note that (2.16) is a representation both for the \( cg \)–iterates (including the indefinite case) and the iterates obtained from SYMMLQ due to [8]. The residual orthogonality is well–known but usually proved for the spd–case only. In Section 3.2 below, we will give an efficient algorithm to compute the iterates based on the orthogonal residuals. Unfortunately, these iterates do not reduce any error norm, if \( A \) is indefinite. For this reason, a further design principle is investigated next.
2.3 Iterative Error Reduction

As was shown in Section 2.2 before, the fact that $\xi_k \neq \xi_1$ for $k < n$ leads to the consequence that the error $x_k - x$ is not reduced systematically in the norm $\| \cdot \| = \langle \cdot, \cdot \rangle^{1/2}$. It might therefore be of interest to study a correction of the type (2.15), but with

$$\xi_k := \eta_j + \beta_k \zeta_j \quad j = 1, \ldots, n$$  \hspace{1cm} (2.23)

for arbitrary choice of $\{\beta_k\}$. The question of interest will then be: can the coefficients $\beta_k$ be determined in such a way that at least (2.7) and (2.9) can be saved, if not the full minimization property (2.8)?

**Lemma 4** Consider general iterates $x_k = x_0 + \Delta x_k$ with $\Delta x_k$ defined by (2.15) and (2.23). Let $\| \cdot \|$ be the norm induced by the specified inner product $\langle \cdot, \cdot \rangle$. Then the choice

$$\beta_k = -\frac{\langle \Delta y_k, \Delta z_k \rangle}{\langle \Delta z_k, \Delta z_k \rangle}$$ \hspace{1cm} (2.24)

guarantees that

$$\varepsilon_{k+1}^2 = \varepsilon_k^2 - \|x_{k+1} - x_k\|^2 \leq \varepsilon_k^2.$$

The denominator in (2.24) never vanishes.

**Proof.** Starting from (2.23), we obtain the error expansion

$$x_k - x = \Delta x_k - \Delta x = \sum_{j=1}^{k} (\eta_j + \beta_k \zeta_j)v_j - \sum_{j=1}^{n} (\eta_j + \xi_1 \cdot \zeta_j)v_j$$

$$= \sum_{j=k+1}^{n} (\eta_j + \xi_1 \cdot \zeta_j)v_j + (\beta_k - \xi_1) \sum_{j=1}^{k} \zeta_j v_j$$

Hence, with (2.15) and the orthogonality of the $\{v_j\}$

$$\langle x_k - x, \Delta x_k \rangle = (\beta_k - \xi_1) \sum_{j=1}^{k} \zeta_j (\eta_j + \beta_k \zeta_j) \sigma_j.$$

There are two choices that make this inner product vanish: either $\beta_k = \xi_1$, the case already discussed in Section 2.1, or

$$\sum_{j=1}^{k} \eta_j \zeta_j \sigma_j + \beta_k \cdot \sum_{j=1}^{k} \zeta_j^2 \sigma_j = 0. \hspace{1cm} (2.25)$$

Upon introducing the definitions of $\Delta y_k, \Delta z_k$ from Lemma 2, the result (2.24) is directly confirmed. Finally note that

$$\langle \Delta z_1, \Delta z_1 \rangle = \zeta_1^2 \sigma_1 = \sigma_1 > 0,$$
which implies
\[ \langle \Delta z_k, \Delta z_k \rangle > 0 \quad \forall k. \]

For actual computation, (2.25) will be used to define \( \{ \beta_k \} \) recursively
\[
\begin{align*}
\mu_1 &:= 0, \quad \nu_1 := \sigma_1, \quad \beta_1 = 0, \\
k = 1, 2, \ldots: \quad &\mu_{k+1} = \mu_k + \eta_{k+1} \zeta_{k+1} \sigma_{k+1} \\
&\nu_{k+1} = \nu_k + \zeta_{k+1}^2 \sigma_{k+1} \\
&\beta_{k+1} = -\mu_{k+1} / \nu_{k+1}
\end{align*}
\] (2.26)

Finally, note that this choice of \( \beta_k \) implies the following orthogonal projection property
\[
\Delta x_k = \Delta y_k - \frac{\langle \Delta y_k, \Delta z_k \rangle}{\langle \Delta z_k, \Delta z_k \rangle} \Delta z_k = \left( I - \frac{\Delta z_k \Delta z_k^T}{\Delta z_k^T \Delta z_k} \right) \Delta y_k \quad (2.27)
\]
written, for ease of notation, in terms of the Euclidean inner product.

3. Compact Iteration Schemes

In the positive definite case, an efficient compact iteration scheme with essentially \( 3n \) array storage is the well–known conjugate gradient method. The aim of this section is to construct a comparable scheme for the indefinite case. One trick in the cg–method certainly is the introduction of the residuals
\[ r_k := b - Ax_k = r_0 - A\Delta x_k \] (3.1)
as the actual variables in order to avoid computational use of the ill–conditioned \( n \)–system of three–term recurrences (1.3) for the basis elements \( \{ v_j \} \). This trick is now analyzed for the indefinite case.

3.1 Residual Projection

The result (1.9) of Lemma 1 readily leads to two different ways of representing the coefficients \( \{ \xi_j \} \). First, we have
\[ \langle v_k, r_k \rangle = \xi_{k+1} \cdot \gamma_{k+1}^2 \sigma_k = \xi_{k+1} \sigma_{k+1}, \quad k = 1, 2, \ldots, n - 1 \] (3.2)
Second, we obtain
\[ \langle v_{k+1}, r_k \rangle = -\xi_k \sigma_{k+1}, \quad k = 0, 1 \ldots, n - 1. \]

For \( k = 0 \), the latter representation leads to \( \xi_0 = -1 \), as required. For \( k > 0 \), however, this representation is just an identity, since the actual computation
of \( r_k \) involves \( x_k \), which, in turn, requires the knowledge of \( \xi_k \). Therefore, only (3.2) remains useful for actual computation. Since (3.2) starts with \( \xi_2 \), any approach based on this formula will only be executable, if \( \xi_1 \) is known — in agreement with the structural results of Lemma 1. Suppose therefore that \( \xi_1 \) is available, then (1.9) inspires the following outline of an algorithm:

\[
\begin{align*}
  k &= 0 : & v_1 &:= r_0 , & x_1 &:= x_0 + \xi_1 v_1 , & \sigma_1 &= \langle v_1, v_1 \rangle \\
  k &= 1, \ldots, n - 1 : & r_k &:= r_{k-1} - \xi_k A v_k \\
 & & [\xi_{k+1}] &:= \langle v_k, r_k \rangle \\
 & & & \text{if } \xi_k \neq 0 \text{ then} \\
 & & v_{k+1} &:= \left( \frac{[\xi_{k+1}] v_k}{\sigma_k} - r_k \right) / \xi_k \\
 & & \sigma_{k+1} &:= \langle v_{k+1}, v_{k+1} \rangle , & \xi_{k+1} &:= [\xi_{k+1}] / \sigma_{k+1} \\
\end{align*}
\]

The critical step in this algorithm is certainly (3.3) for \( \xi_k = 0 \). Fortunately, vanishing components of \( \{\xi_j\} \) can only occur once — see Lemma 1. So a type of “look–ahead strategy” can be derived, in principle. For details of such strategies, see e.g. [4] or [1]. Unfortunately, however, “nearly vanishing” components of \( \{\xi_j\} \) cannot generally be excluded — with the consequence that a look–ahead strategy of variable length must be developed for emergency situations. Moreover, upon inserting (3.2) into (3.3), any new basis element is seen to come from a residual projection, since

\[
\begin{align*}
  v_{k+1} &= \left( \frac{\langle v_k, r_k \rangle}{\langle v_k, v_k \rangle} v_k - r_k \right) = - \left( I - \frac{v_k v_k^T}{v_k^T v_k} \right) r_k .
\end{align*}
\]

(Once more, for ease of writing, the Euclidean inner product has been used.) As a consequence, either numerical instability or frequent restart under a condition such as

\[
\sigma_{k+1} < \sigma_1 \cdot \varepsilon \quad \text{(3.5)}
\]

will occur. For this reason, this type of algorithm is not pursued further here.

### 3.2 Residual Orthogonality

Upon recalling the general iteration for

\[
\Delta x_k = \sum_{j=1}^{k} \xi_j^k v_j , \quad \xi_j^k = \eta_j + \beta_k \zeta_j \quad j = 1, \ldots, k ,
\]

13
an alternative simple idea to exploit the residuals as actual variables can be derived. Lemma 3 above shows that for

$$\hat{\beta}_k := \xi_k = -\frac{\eta_{k+1}}{\zeta_{k+1}}$$

(3.7)

the iterates \(\hat{x}_k := x_0 + \Delta x_k\) give rise to orthogonal residuals of the form

$$\hat{r}_k = -\hat{\xi}_k v_{k+1} \quad k = 0, \ldots, n - 1,$$

with

$$\hat{\xi}_k := -\frac{\sigma_{1}}{(\zeta_{k+1}\sigma_{k+1})} \quad k = 0, \ldots, n - 1.$$  

For actual computation of the \(\{\hat{\xi}_k\}\) note that, with (1.24)

$$\frac{\hat{\xi}_k}{\xi_{k-1}} = \frac{\xi_k \sigma_k}{\zeta_{k+1} \sigma_{k+1}} = -\frac{1}{d_k} \quad k = 1, \ldots, n.$$  

(3.8)

This recursion is started with \(\hat{\xi}_0 = -1\), which implies \(\hat{\xi}_1 = 1/\alpha_1\). In a similar way, the \(\{\hat{\beta}_k\}\) can be obtained recursively starting from

$$\hat{\beta}_1 = -\frac{\eta_0 + \alpha_1 \eta_1}{\zeta_0 + \alpha_1 \zeta_1} = \frac{1}{\alpha_1}.$$  

(3.9)

Now, define \(\hat{\tau}_1 := \hat{\beta}_1\) and

$$\hat{\tau}_k := \hat{\beta}_k - \hat{\beta}_{k-1} \quad k = 2, \ldots, n.$$  

(3.10)

Then, with (1.26), we have

$$\hat{\tau}_k = -\frac{\eta_{k+1}}{\zeta_{k+1}} + \frac{\eta_k}{\zeta_k} = \frac{D(k, k+1)}{\zeta_k \zeta_{k+1}} = -\frac{\xi_k}{\zeta_k} \frac{\sigma_k}{\zeta_k \zeta_{k+1} \sigma_{k+1}} = \hat{\xi}_k.$$  

(3.11)

Hence, for \(k = 1, \ldots, n - 1:\)

$$\frac{\hat{\tau}_{k+1}}{\hat{\tau}_k} = \frac{\zeta_k \zeta_{k+1} \sigma_{k+1}}{\zeta_{k+1} \zeta_{k+2} \sigma_{k+2}} \frac{\sigma_{k+1}}{\sigma_k} \frac{\zeta_k \delta_k}{\zeta_{k+1} \sigma_{k+1}} \frac{\zeta_{k+1} \sigma_{k+1}}{\zeta_{k+2} \sigma_{k+2}},$$

which, by insertion of (1.24), yields

$$\hat{\tau}_{k+1} = \frac{\hat{\tau}_k \sigma_{k+1}}{(\sigma_k d_k d_{k+1})} \quad k = 1, \ldots, n - 1.$$  

(3.12)

The orthogonality of the residuals can be exploited to derive an alternative recursive scheme in terms of the basis \(\{v_j\}\). The derivation requires some rather straightforward calculations, which we summarize in the following lemma.
Lemma 5 Let \( \{\hat{x}_k\} \) denote the extended cg–iterates, \( \{\hat{r}_k\} \) the associated residuals, and \( d\hat{x}_k := \hat{x}_k - \hat{x}_{k-1} \), the iterative corrections. Moreover, define the expressions

\[
\hat{\sigma}_k := \langle \hat{r}_{k-1}, \hat{r}_{k-1} \rangle, \quad \hat{d}_k := d_k / \hat{\sigma}_k, \quad \hat{\alpha}_k = \langle \hat{r}_{k-1}, A\hat{r}_{k-1} \rangle / \hat{\sigma}_k. \tag{3.13}
\]

Then the iterates can be computed by the compact scheme \((k = 1, \ldots, n - 1)\) as follows:

\[
\hat{r}_k := \hat{r}_{k-1} - A\hat{x}_k
\]

\[
d\hat{x}_{k+1} := \frac{\hat{d}_k}{d_{k+1}} d\hat{x}_k + \frac{1}{\hat{\sigma}_{k+1} d_{k+1}} \hat{r}_k \tag{3.14}
\]

\[
\hat{x}_{k+1} := \hat{x}_k + d\hat{x}_{k+1},
\]

wherein

\[
\hat{d}_k = \sum_{j=1}^{k} (-1)^{k-j} \hat{\alpha}_j / \hat{\sigma}_j. \tag{3.15}
\]

The iteration is started with

\[
\hat{r}_0 := b - Ax_0, \quad d\hat{x}_1 = \frac{1}{\alpha_1} \hat{r}_0, \quad \hat{x}_1 = x_0 + d\hat{x}_1.
\]

Proof. For general iterates \( \{x_k\} \) we obtain

\[
dx_{k+1} = \Delta x_{k+1} - \Delta x_k = \sum_{j=1}^{k+1} (\eta_j + \beta_{k+1} \zeta_j) v_j - \sum_{j=1}^{k} (\eta_j + \beta_k \zeta_j) v_j
\]

\[
= (\eta_{k+1} + \beta_k \zeta_{k+1}) v_{k+1} + (\beta_{k+1} - \beta_k) \sum_{j=1}^{k} \zeta_j v_j
\]

\[
= (\eta_{k+1} + \beta_k \zeta_{k+1}) v_{k+1} + (\beta_{k+1} - \beta_k) \Delta z_{k+1}. \tag{3.16}
\]

The special iterates \( \{\hat{x}_k\} \) are characterized by \( \beta_k = \hat{\beta}_k \), which yields

\[
d\hat{x}_{k+1} = \hat{r}_{k+1} \Delta z_{k+1}.
\]

From this, we directly calculate

\[
d\hat{x}_{k+1} = \hat{r}_{k+1} (\Delta z_k + \zeta_{k+1} v_{k+1}) = \frac{\hat{r}_{k+1}}{\hat{r}_k} d\hat{x}_k = \frac{\hat{r}_{k+1} \zeta_{k+1} \sigma_{k+1}}{\xi_k} \frac{1}{d_{k+1}} - 1.
\]

Insertion of (3.11), (3.12) and (1.24) yields

\[
\frac{\hat{r}_{k+1} \zeta_{k+1} \sigma_{k+1}}{\xi_k} = \frac{\hat{r}_k}{\xi_k} \frac{\zeta_{k+1} \sigma_{k+1}}{\zeta_k \sigma_k d_k} = \frac{\zeta_{k+1} \sigma_{k+1}}{\zeta_k \sigma_k d_k}. \frac{1}{d_{k+1}} = -\frac{1}{d_{k+1}}.
\]
Once more with (3.12), we thus have

\[ d\hat{x}_{k+1} = \frac{\sigma_{k+1}}{\sigma_k d_k d_{k+1}} d\hat{x}_k + \frac{1}{d_{k+1}} \hat{r}_k. \]

At this point, the continued fraction representation (1.10) can be used, in principle, which reads

\[ d_1 = \alpha_1, \quad d_{k+1} = \alpha_{k+1} - \frac{\sigma_{k+1}}{\sigma_k d_k} \quad k = 1, \ldots, n - 1. \]

Note that due to (2.18)

\[ \tilde{\sigma}_{k+1} = \tilde{\xi}_k^2 \sigma_{k+1}, \quad \tilde{\alpha}_{k+1} = \alpha_{k+1}. \]

Upon introducing \( \hat{d}_k \), we arrive at the linear recursion

\[ \hat{d}_k = \frac{\hat{\alpha}_k}{\hat{\sigma}_k} - \hat{d}_{k-1}, \quad (3.17) \]

which can be solved in closed form to confirm (3.15). Finally, with (3.8) it is seen that

\[ \frac{\sigma_{k+1}}{\sigma_k d_k d_{k+1}} = \frac{\hat{\sigma}_{k+1} \hat{\xi}_{k+1}^2}{\hat{\xi}_k^2 \hat{\sigma}_k \cdot d_k d_{k+1}} = \frac{\hat{\sigma}_{k+1} d_k}{\hat{\sigma}_k d_{k+1}} = \hat{d}_k / \hat{d}_{k+1}. \]

\[ \blacksquare \]

Remark. For the sake of clarity, the connection of the recursive scheme (3.14) with the usual cg–scheme is described. For this purpose, just combine (2.2), which defines the basis \( \{p_j\} \) from the basis \( \{v_j\} \), with the residual representation (1.9). By means of the simple scaling transformation

\[ p_k := -\tilde{\xi}_{k-1} \sqrt{\sigma_k} \tilde{p}_k, \quad (3.18) \]

one arrives at

\[ p_1 = \tilde{r}_0, \quad p_{k+1} = \tilde{r}_k + \frac{\hat{\sigma}_k}{\hat{\sigma}_{k+1}} p_k, \quad k = 1, \ldots, n - 1, \quad (3.19) \]

which is the usual formula — see e.g. [6] or [3]. In the standard cg–scheme, the \( \{d_k\} \) are computed via (2.4), whereas here formula (3.15) is employed.

In order to actually evaluate \( \{\hat{d}_k\} \) via (3.15), define

\[ S_k := \sum_{j=1}^{k} (-1)^{j-1} \hat{\alpha}_j =: S_k^+ - S_k^- \quad (3.20) \]
wherein $S^+_k$, $S^-_k$ represent all positive or all negative terms, respectively, in the sum so that
\[
\hat{d}_k = (-1)^{k-1}(S^+_k - S^-_k), \quad S^+_k, S^-_k > 0.
\] (3.21)

The relative condition number of this summation is then
\[
\kappa_k := \frac{S^+_k + S^-_k}{S^+_k - S^-_k}.
\] (3.22)

Restart of the iteration (3.14) will therefore be necessary whenever the requirement
\[
\varepsilon \kappa_k < 1
\] (3.23)
is violated — for some suitably chosen default value $\varepsilon$.

### 3.3 Blended Iteration Schemes

The above considerations led to an alternative compact scheme for the cg–iteration, especially designed for the symmetric indefinite case. It can, however, also just be used to compute the necessary intermediate terms for any other iteration characterized by
\[
\Delta x_k = \sum_{j=1}^k (\eta_j + \beta_k \zeta_j) v_j
\] with $\beta_k \neq \hat{\beta}_k$. For the associated general iteration recall from (3.16) that
\[
dx_{k+1} = (\eta_{k+1} + \beta_k \zeta_{k+1}) v_{k+1} + \tau_{k+1} \Delta z_{k+1}
\] with $\tau_{k+1} := \beta_{k+1} - \beta_k$. In the frame of Lemma 5 just observe that
\[
v_{k+1} = -\frac{1}{\xi_k} \hat{r}_k, \quad \Delta z_{k+1} = \frac{1}{\tau_{k+1}} \hat{d} \hat{r}_{k+1}.
\] (3.24)

Therefore, we must just replace the $\hat{x}$–update by an $x$–update of the form
\[
x_{k+1} = x_k + \frac{\tau_{k+1}}{\tau_{k+1}} \hat{d} \hat{r}_{k+1} - \frac{(\eta_{k+1} + \beta_k \zeta_{k+1})}{\xi_k} \hat{r}_k.
\] (3.25)

For computational realization, note that (1.24) is equivalent to
\[
\zeta_{k+1} = -\frac{\zeta_k}{(\hat{d}_k \hat{\sigma}_{k+1})}
\] (3.26)
and (2.19) can be used to derive
\[
\hat{\xi}_k = -\frac{\zeta_{k+1} \hat{\sigma}_{k+1}}{\hat{\sigma}_{k+1}} = \frac{\zeta_k}{(\hat{d}_k \hat{\sigma}_1)}.
\] (3.27)
Moreover, $\hat{\tau}_{k+1}$ can be expressed by

$$\hat{\tau}_{k+1} = 1/(\hat{\sigma}_j \hat{d}_{k+1}).$$

(3.28)

Finally, the factor before $\hat{r}_k$ should be evaluated in the form

$$\eta_{k+1} + \beta_k \zeta_{k+1} = (\beta_k - \hat{\beta}_k)/(\xi_k \zeta_{k+1}),$$

which then assures that $x_k = \hat{x}_k$ arises also numerically for $\beta_k = \hat{\beta}_k$ and $\tau_{k+1} = \hat{\tau}_{k+1}$. 
References


