The Effects of Loss of Orthogonality on Large Scale Numerical Computations

Christopher C. Paige

McGill University, Montreal, Quebec H3A 0E9, Canada
Paige@cs.mcgill.ca
http://www.cs.mcgill.ca/~chris/welcome

Abstract. Many useful large sparse matrix algorithms are based on orthogonality, but for efficiency this orthogonality is often obtained via short term recurrences. This can lead to both loss of orthogonality and loss of linear independence of computed vectors, yet with well designed algorithms high accuracy can still be obtained. Here we discuss a nice theoretical indicator of loss of orthogonality and linear independence and show how it leads to a related higher dimensional orthogonality that can be used to analyze and prove the effectiveness of such algorithms. We illustrate advantages and shortcomings of such algorithms with Cornelius Lanczos' Hermitian matrix tridiagonalization process. The paper is reasonably expository, keeping simple by avoiding some detailed analyses.

Keywords: Orthogonality · Large sparse matrices · Lanczos process

1 Introduction

We deal with the loss of orthogonality caused by finite precision calculations. For the computations and diagrams here we use IEEE Standard double precision floating point arithmetic (with unit roundoff $\epsilon = 2^{-53} \approx 10^{-16}$).

This loss of orthogonality can come in different forms, for example it was shown by Paige et al. [13] that the Modified Gram Schmidt (MGS) based Generalized Minimum Residual (GMRES) method of Saad and Schultz [18] will produce a backward stable solution (i.e., a solution that is exact for a slightly perturbed problem, see e.g., [5, Sect. 3.4.10]) to an unsymmetric linear system $Ax = b$, $A \in \mathbb{R}^{n\times n}$ nonsingular, in n steps. If the MGS process is used continuously for the full n steps, the sequence of supposedly orthogonal vectors that MGS-GMRES produces loses orthogonality, usually severely, but linear independence is not lost. However the k-th step involves orthogonalizing a vector against the previous k vectors, and so for large problems MGS-GMRES is usually too expensive to be used this way for a full n steps.

The author's work was supported by NSERC of Canada grant OGP0009236.

O. Gervasi et al. (Eds.): ICCSA 2018, LNCS 10962, pp. 429–439, 2018.
https://doi.org/10.1007/978-3-319-95168-3_29

Fortunately there is a class of short recurrence orthogonalization algorithms based on the Hermitian matrix tridiagonalization process of Lanzos [7], "the Lanczos process". But the sequence of supposedly orthogonal vectors $v_1, v_2, \ldots, v_k$ that the Lanczos process produces might not only lose orthogonality, it can lose linear independence as well. Surprisingly, if implemented correctly, see e.g., [5, Sect. 10.1.2], the process still provides accurate results, although sometimes taking many more than the expected n steps. Thus such processes tend to be used iteratively, stopping when some desired criterion is established.

The loss of linear independence makes the analysis of such algorithms both different and difficult. However it is still important to show if such short recurrence algorithms necessarily supply good accuracy, and if in fact they supply backward stable solutions. After giving an approach to measuring loss of orthogonality, and using this to provide a related true orthogonality in a higher dimension, we illustrate an approach to analyzing such short recurrence algorithms by combining these ideas to analyze the Lanczos process.

2 Notation and Terminology

We use "$\triangleq$" for "is defined to be", and "$\equiv$" for "is equivalent to". Let I_n denote the $n \times n$ unit matrix, with j-th column e_j. We say $Q_1 \in \mathbb{C}^{n\times k}$ has orthonormal columns if $Q_1^H Q_1 = I$ and write $Q_1 \in \mathbb{U}^{n\times k}$. We denote the Frobenius norm by $\|B\|_F \triangleq \sqrt{\text{trace}(B^H B)}$, the Euclidean norm by $\|v\|_2 \triangleq \sqrt{v^H v}$ and the spectral norm by $\|B\|_2 \triangleq \sigma_{\max}(B)$, the maximum singular value of B.

We often index matrices by dimensional subscripts as in V_k when the $(k{+}1)$-st matrix can be obtained from the k-th by adding a column, or a column and a row. This holds for $V_k \in \mathbb{C}^{n\times k}$ and $S_k \in \mathbb{C}^{k\times k}$. Otherwise we use superscripts, as in $Q^{(k)}$, and then subscripts will denote partitioning, as in $Q^{(k)} \equiv [Q_1^{(k)} | \, Q_2^{(k)}]$. We often omit the particular superscript $\cdot^{(k)}$ when the meaning is clear (but do not omit any other superscripts, e.g., we do not omit $\cdot^{(k+1)}$).

We sometimes use "$\approx$" to mean "equal to within $O(\epsilon)\|A\|_2$", where, together with the computer floating-point precision ϵ, $O(\epsilon)$ may be polynomially dependent on the number of steps k, the dimension n of A, and the maximum number of nonzeros in a row of A, see [11, Sect. 3.2].

3 Measuring Loss of Orthogonality

If we are interested in the loss of orthogonality of k vectors in $\mathbb{C}^n$ we are not concerned with their lengths, so from now on we assume we are dealing with

$$V_k = [v_1, v_2, \ldots, v_k] \in \mathbb{C}^{n\times k}, \quad \|v_j\|_2 = 1, \quad j{=}1{:}k. \tag{1}$$

If these vectors are orthogonal then $V_k^H V_k = I_k$, so a possible measure of loss of orthogonality is $\|V_k^H V_k - I_k\|_2$. Alternatively, let U_k be the strictly upper triangular part of $V_k^H V_k$, so $V_k^H V_k = U_k^H + I_k + U_k$, then we could use $\|U_k\|_2$ as a measure. Unfortunately $\|U_k\|_2$ can become unbounded as k increases.

A breakthrough in handling loss of orthogonality in numerical algorithms arose with the realization in [10] that an idea on loss of orthogonality in modified Gram-Schmidt (MGS) outlined by Björck and Paige in [1] could be extended to apply to *any* sequence of unit-length vectors v_j. This was based on the matrix S_k defined in [10, Theorem 2.1], for which we will show in (4) with U_k above

$$S_k \triangleq (I_k + U_k)^{-1}U_k \in \mathbb{C}^{k\times k},$$
$$\|S_k\|_2 \le 1; \quad V_k^H V_k = I \Leftrightarrow \|S_k\|_2 = 0; \quad V_k^H V_k \ \text{ singular} \Leftrightarrow \|S_k\|_2 = 1.$$

The number of unit singular values of S_k equals the rank deficiency of V_k. With these properties, S_k provides an ideal description of the loss of orthogonality of V_k in (1). In Fig. 1 we give an example showing how $\|U_k\|_2$ keeps increasing with k, while $\|S_k\|_2$ reaches and stays at 1, indicating loss of linear independence.

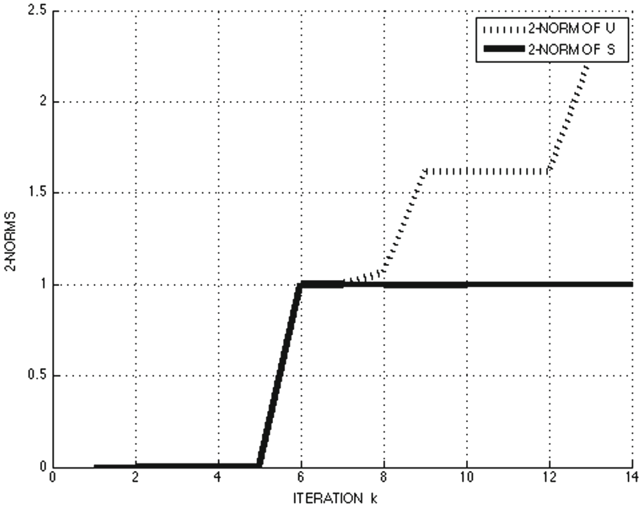

Fig. 1. Loss of orthogonality & linear independence, via $\|U_k\|_2$ and $\|S_k\|_2$.

4 Obtaining a Unitary Matrix from Unit-Length Vectors

We can use the strictly upper triangular matrix S_k for more than indicating loss of orthogonality, it allows us to develop an $(n+k)\times(n+k)$ unitary matrix $Q^{(k)}$ from any $n \times k$ matrix V_k with unit-length columns. This can be used for analyzing the behavior of orthogonalization algorithms, as we show later.

Theorem 1 ([10, Theorem 2.1]). *For integers $n \ge 1$ and $k \ge 1$, and $V_j \triangleq [v_1, \ldots, v_j] \in \mathbb{C}^{n\times j}$ with $\|v_j\|_2 = 1$, $j = 1:k+1$, define the matrix S_k based on U_k, the strictly upper triangular part of $V_k^H V_k = I + U_k + U_k^H$,*

$$S_k \triangleq (I_k + U_k)^{-1}U_k \equiv \ U_k(I_k + U_k)^{-1} \in \mathbb{C}^{k\times k}. \tag{2}$$

Clearly S_k is strictly upper triangular and $I_k - S_k$ is always nonsingular. Then

$$U_k S_k = S_k U_k, \quad U_k = (I_k - S_k)^{-1} S_k \equiv S_k (I_k - S_k)^{-1}, \quad (I_k - S_k)^{-1} = I_k + U_k, \quad (3)$$

$$\|S_k\|_2 \leq 1; \quad V_k^H V_k = I \Leftrightarrow \|S_k\|_2 = 0; \quad V_k^H V_k \ \textit{singular} \Leftrightarrow \|S_k\|_2 = 1. \quad (4)$$

Importantly, S_k is the unique *strictly upper triangular $k \times k$ matrix such that*

$$Q^{(k)} \equiv \left[\begin{array}{c|c} Q_{11}^{(k)} & Q_{12}^{(k)} \\ \hline Q_{21}^{(k)} & Q_{22}^{(k)} \end{array}\right] \triangleq \left[\begin{array}{c|c} S_k & (I_k - S_k) V_k^H \\ \hline V_k (I_k - S_k) & I_n - V_k (I_k - S_k) V_k^H \end{array}\right] \in \mathbb{U}^{(n+k)\times(n+k)}. \quad (5)$$

We also write $Q^{(k)} \equiv \left[\underset{k}{Q_1^{(k)}} \middle| \underset{n}{Q_2^{(k)}}\right]$. Define $\begin{bmatrix} s_k \\ 0 \end{bmatrix} \triangleq S_k e_k$, then with (3) we have

$$S_k e_k = (I_k - S_k) U_k e_k = \begin{bmatrix} (I_{k-1} - S_{k-1}) V_{k-1}^H v_k \\ 0 \end{bmatrix}, \quad s_{k+1} = (I_k - S_k) V_k^H v_{k+1}, \quad (6)$$

$$Q_1^{(k+1)} = \left[\begin{array}{c} S_{k+1} \\ \hline V_{k+1}(I_{k+1} - S_{k+1}) \end{array}\right] = \left[\begin{array}{c|c} S_k & s_{k+1} \\ 0 & 0 \\ \hline V_k (I_k - S_k) & v_{k+1} - V_k s_{k+1} \end{array}\right]. \quad (7)$$

Proof. This was proven in [10], but we give a simpler proof that $Q_1^{(k)H} Q_1^{(k)} = I$ here. We drop the sub- and superscripts k for clarity.

First in (3) $US = SU$ follows from (2), as does $(I + U)S = U$, then

$$\begin{aligned}
&(I + U)(I - S) = (I + U)I - (I + U)S = (I + U) - U = I, \\
&U(I - S) = I - (I - S) = S, \qquad U = S(I - S)^{-1}, \\
&Q_1 \triangleq \begin{bmatrix} S \\ V(I - S) \end{bmatrix} = \begin{bmatrix} S(I - S)^{-1} \\ V \end{bmatrix} (I - S) = \begin{bmatrix} U \\ V \end{bmatrix} (I - S), \\
&Q_1^H Q_1 = (I - S)^H [U^H U + V^H V](I - S) = (I - S)^H [U^H U + I + U^H + U](I - S) \\
&\qquad\quad = (I - S)^H [(I + U)^H (I + U)](I - S) = I.
\end{aligned}$$

This approach can easily be extended to prove that $Q^{(k)} \in \mathbb{U}^{(n+k)\times(n+k)}$ in (5). The proofs that $\|S_k\|_2 \leq 1$, and that the number of unit singular values of S_k denote the rank deficiency of V_k, follow immediately from the CS-Decomposition (CSD) (see [2,19] and e.g., [5]) applied to $Q_1^{(k)} \in \mathbb{U}^{(k+n)\times k}$ in (5).

5 The Lanczos Tridiagonalization Process

Given $A = A^H \in \mathbb{C}^{n\times n}$ and a vector $v_1 \in \mathbb{C}^n$ of unit-length, i.e., $v_1^H v_1 = 1$, a reasonable implementation of the Hermitian matrix tridiagonalization process of Lanczos, see [7], [9, (2.1)–(2.8)], and, e.g., [5, Sects. 10.1–10.3], gives the equivalent of a 3-term recurrence: $v_2 \beta_2 = A v_1 - v_1 \alpha_1$, then for $k = 1, 2, \ldots$

$$v_{k+1} \beta_{k+1} = A v_k - v_k \alpha_k - v_{k-1} \beta_k,$$

where $\alpha_k = v_k^H A v_k$ and $\beta_{k+1} > 0$ is chosen to ensure that $\|v_{k+1}\|_2 = 1$.

If we define $V_k \triangleq [v_1, \ldots, v_k] \in \mathbb{C}^{n\times k}$ then in theory this gives after k steps

$$AV_k = V_k T_k + v_{k+1}\beta_{k+1}e_k^T = V_{k+1}T_{k+1,k}, \quad V_k^H V_k = I_k, \quad T_{k+1,k} = \left[\frac{T_k}{\beta_{k+1}e_k^T}\right], \quad (8)$$

where the real symmetric tridiagonal matrix T_k has diagonal elements $\alpha_1, \ldots, \alpha_k$ and positive next-to-diagonal elements $\beta_2, \ldots, \beta_k$, and, again in theory, the process will necessarily stop in $\ell \le n$ steps, with $\beta_{\ell+1} = 0$.

Lanczos originally presented his tridiagonalization process in [7] for solving the eigenproblem of $A = A^H$, but mentioned it would be useful for solving linear systems $Ax = b$, and in [8] he adapted it for this purpose when A is symmetric positive definite. This was equivalent to taking $\beta_1 = \|b\|_2$, $v_1 = b/\beta_1$, and at the k-th step of the Lanczos process (8) computing the approximation $x_k = V_k z_k$ where $T_k z_k = e_1\beta_1$. We call this "Lanczos-CG". In theory this gives the solution x no later than the n-th step, and is mathematically equivalent to Hestenes and Stiefel's method of conjugate gradients (CG) in [6], see, e.g., [5, Sect. 11.3].

6 Analysis of the Finite Precision Lanczos Process

To understand the effect of rounding errors, the finite precision Lanczos process can be analyzed by using Theorem 1 to move to a higher dimension.

Theorem 2 ([11, Corollary 3.2]). *After k finite precision steps of a well implemented Lanczos process with $A = A^H$ leading to $T_{k+1,k}$ as in (8), let $V_{k+1} = [v_1, v_2, \ldots, v_{k+1}]$ be the matrix of computed Lanczos vectors normalized to have unit length. Then with the backward error term $H^{(k)} = H^{(k)H}$ where $\|H^{(k)}\|_2 \le O(\epsilon)\|A\|_2$, if S_k and s_{k+1} are as in Theorem 1, we have*

$$\left(\begin{bmatrix} T_k & 0 \\ 0 & A \end{bmatrix} + H^{(k)}\right) Q_1^{(k)} = Q_1^{(k)} T_k + q^{(k+1)}\beta_{k+1}e_k^T = \left[Q_1^{(k)} \,\middle|\, q^{(k+1)}\right] T_{k+1,k}, \quad (9)$$

$$\left[Q_1^{(k)} \,\middle|\, q^{(k+1)}\right] \triangleq \left[\begin{array}{c|c} S_k & s_{k+1} \\ V_k(I - S_k) & v_{k+1} - V_k s_{k+1} \end{array}\right] \in \mathbb{U}^{(n+k)\times(k+1)}. \quad (10)$$

Here $q^{(k+1)}$ is the last column of $Q_1^{(k+1)}$ with its zero $(k+1)$-st element removed, see (7). Bounds for $H^{(k)}$ are discussed in [11, Sect. 3].

This shows that the computed $T_{k+1,k}$ is the *exact* result of k steps of an exact Lanczos process with exact orthogonality arising from a strange but fully defined Hermitian matrix with an $O(\epsilon)\|A\|_2$ Hermitian backward error $H^{(k)}$. This surprising higher dimensional result allows us to understand the behavior of the finite precision Lanczos process, as we will now illustrate.

7 Accuracy of the Finite Precision Lanczos Process

Here we restrict the analysis to the most usual case of $A = A^H$ with no multiple eigenvalues, and v_1 having a component of every eigenvector of A. This allows a much simpler presentation than that in [12].

The behavior of the finite precision Lanczos process depends heavily on the $Q^{(k)}$ in (5) used in Theorem 2. If there are no rounding errors $S_k = 0$ and $Q_{22}^{(n)} = 0$:

$$Q^{(k)} \equiv \left[\begin{array}{c|c} Q_{11}^{(k)} & Q_{12}^{(k)} \\ \hline Q_{21}^{(k)} & Q_{22}^{(k)} \end{array}\right] \triangleq \left[\begin{array}{c|c} 0 & V_k^H \\ \hline V_k & I_n - V_k V_k^H \end{array}\right] \in \mathbb{U}^{(n+k)\times(n+k)}, \quad Q^{(n)} = \left[\begin{array}{c|c} 0 & V_n^H \\ \hline V_n & 0 \end{array}\right].$$

The practical behavior also depends on $\|Q_{22}^{(k)}\|_F$ decreasing. From (5) and (6),

$$Q_{22}^{(k)} v_{k+1} = [I_n - V_k(I_k - S_k)V_k^H] v_{k+1} = v_{k+1} - V_k s_{k+1}. \tag{11}$$

Define the orthogonal projectors $\mathcal{P}_j \triangleq I_n - v_j v_j^H$. Since S_k is strictly upper triangular, $S_1 = 0$ and $Q_{22}^{(1)} = \mathcal{P}_1$. Then from (5) and (11) $Q_{22}^{(k)} = \mathcal{P}_1 \cdots \mathcal{P}_k$, since

$$\begin{aligned} Q_{22}^{(k+1)} &= I_n - \begin{bmatrix} V_k & v_{k+1} \end{bmatrix} \begin{bmatrix} I_k - S_k & -s_{k+1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_k^H \\ v_{k+1}^H \end{bmatrix} \qquad (12) \\ &= I_n - V_k(I_k - S_k)V_k^H - (v_{k+1} - V_k s_{k+1})v_{k+1}^H \\ &= Q_{22}^{(k)} - Q_{22}^{(k)} v_{k+1} v_{k+1}^H = Q_{22}^{(k)}(I_n - v_{k+1}v_{k+1}^H) = Q_{22}^{(k)}\mathcal{P}_{k+1} = \mathcal{P}_1 \cdots \mathcal{P}_{k+1}, \\ \|Q_{22}^{(k+1)}\|_F^2 &= \mathrm{trace}[Q_{22}^{(k+1)} Q_{22}^{(k+1)H}] = \mathrm{trace}[Q_{22}^{(k)}(I_n - v_{k+1}v_{k+1}^H)Q_{22}^{(k)H}] \\ &= \|Q_{22}^{(k)}\|_F^2 - \|Q_{22}^{(k)} v_{k+1}\|_2^2 = \|Q_{22}^{(k)}\|_F^2 - \|v_{k+1} - V_k s_{k+1}\|_2^2, \qquad (13) \end{aligned}$$

showing just how $\|Q_{22}^{(k)}\|_F$ decreases. It was shown in [9] that orthogonality is not lost in the Lanczos process until the first eigenvector of A has converged, so until that point $\|Q_{22}^{(k)}\|_F^2$ decreases by about 1 each step.

It can be seen from (13) that $\|Q_{22}^{(k)}\|_F$ is non-increasing, and in [12] it is shown that $\|Q_{22}^{(k)}\|_F^2 \searrow 0$ with the above conditions, but here we avoid that analysis and just give results when $Q_{22}^{(k)} = 0$ for some k. In (5) $Q_{22}^{(k)} = 0$ implies that $Q_{12}^{(k)} \in \mathbb{U}^{k\times n}$ and $Q_{21}^{(k)H} \in \mathbb{U}^{k\times n}$, so that from the CS-Decomposition of $Q^{(k)}$ the singular values of $S_k \equiv Q_{11}^{(k)}$ include exactly n zeros and $k-n$ ones, see also [17], therefore there exist $W \triangleq [W_1, W_2] \in \mathbb{U}^{k\times k}$ and $P \triangleq [P_1, P_2] \in \mathbb{U}^{k\times k}$ such that the singular value decomposition (SVD) of S_k is

$$S_k = W_1 P_1^H, \quad S_k P_2 = 0, \quad W_2^H S_k = 0; \qquad W_2, P_2 \in \mathbb{U}^{k\times n}. \tag{14}$$

Now since $Q_1^{(k)} P \in \mathbb{U}^{(k+n)\times k}$ we have in (10), with $(I - S_k)^H V_k^H = Q_{21}^{(k)H} \in \mathbb{U}^{k\times n}$,

$$Q_1^{(k)} P = \begin{bmatrix} S_k P \\ V_k(I - S_k)P \end{bmatrix} = \left[\begin{array}{c|c} W_1 & 0 \\ V_k(P_1 - W_1) & V_k P_2 \end{array}\right] = \left[\begin{array}{c|c} W_1 & 0 \\ 0 & V_k P_2 \end{array}\right], \quad V_k P_2 \in \mathbb{U}^{n\times n}. \tag{15}$$

From (13) and the orthonormality in (10) together with (14), we have

$$Q_{22}^{(k)}=0 \Rightarrow \{v_{k+1}=V_k s_{k+1}, \quad S_k^H s_{k+1}=0, \quad \|s_{k+1}\|_2=1, \quad s_{k+1}=W_2 W_2^H s_{k+1}\}.$$

Multiplying (9) on the right by P and using (15), where $\left[\begin{smallmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{smallmatrix}\right] \triangleq H^{(k)}$,

$$\begin{aligned}
&\begin{bmatrix} T_k+H_{11} & H_{12} \\ H_{21} & A+H_{22} \end{bmatrix}\left[\begin{array}{c|c} W_1 & 0 \\ 0 & V_k P_2 \end{array}\right] = \left[\begin{array}{c|c} W_1 & 0 \\ 0 & V_k P_2 \end{array}\right] P^H T_k P + \begin{bmatrix} W_2 W_2^H s_{k+1} \\ 0 \end{bmatrix} \beta_{k+1} e_k^T P, \\
&H_{21} W_1 = V_k P_2 P_2^H T_k P_1, \quad P_2^H T_k P_1 = (V_k P_2)^H H_{21} W_1 \approx 0, \\
&T_k P_2 = P_2(P_2^H T_k P_2) + P_1(P_1^H T_k P_2) = P_2(P_2^H T_k P_2) + P_1(H_{21} W_1)^H V_k P_2, \\
&(T_k - P_1 W_1^H H_{21}^H V_k) P_2 = P_2(P_2^H T_k P_2), \quad P^H T_k P \approx \begin{bmatrix} P_1^H T_k P_1 & 0 \\ 0 & P_2^H T_k P_2 \end{bmatrix}, \qquad (16) \\
&(A + H_{22}) V_k P_2 = V_k P_2 (P_2^H T_k P_2), \quad (V_k P_2) \in \mathbb{U}^{n \times n}.
\end{aligned}$$

But P_2 in (14) is arbitrary up to multiplication on the right by a unitary matrix, so we can transform to give $\widetilde{P}_2^H T_k \widetilde{P}_2 = \tilde{\Lambda} = \mathrm{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n)$, giving

$$(T_k - P_1 W_1^H H_{21}^H V_k) \widetilde{P}_2 = \widetilde{P}_2 \tilde{\Lambda}, \qquad (17)$$

$$(A + H_{22})(V_k \widetilde{P}_2) = (V_k \widetilde{P}_2) \tilde{\Lambda}, \quad V_k \widetilde{P}_2 \in \mathbb{U}^{n \times n}. \qquad (18)$$

Therefore $\{\tilde{\Lambda}, (V_k \widetilde{P}_2)\}$ is a backward stable eigensystem for A, i.e., is an eigensystem for a slightly perturbed A, where $\{\tilde{\Lambda}, \widetilde{P}_2\}$ is a backward stable partial eigensystem for T_k. A similar approach can be used to show that solutions of equations methods based on the Lanczos process for sufficiently nonsingular Hermitan A make available backward stable solutions $\tilde{x}$ for $Ax = b$ via, see [12],

$$(T_k - P_1 W_1^H H_{12} V_k) \tilde{y} = e_1 \beta_1, \quad (A + H_{22}) V_k \tilde{y} = b, \quad \tilde{x} \triangleq V_k \tilde{y}.$$

So when $Q_{22}^{(k)} = 0$, the Lanczos process makes available backward stable solutions for the eigenproblem and solution of equations with Hermitian A.

Some important points need to be emphasized.

- We wrote "makes available backward stable solutions" because this paper and [12] only analyze the Lanczos process, and do not include the extra steps required to solve the eigenproblem or solution of equations with T_k, etc.
- When $A = A^H$ has no multiple eigenvalues we often do not need to continue until $Q_{22}^{(k)} = 0$ to machine precision, because we often obtain the solutions we want before this. An advantage of the present analysis is that, since we would eventually obtain complete backward stable solutions, we now know that the process never loses accuracy (apart from the possible build up of the $O(\epsilon)$ mentioned in Sect. 2, although that seems small in practice).

- When A has multiple eigenvalues we do not expect to obtain $Q_{22}^{(k)} = 0$. We still obtain accurate eigenvalues and solutions of equations, but eigenvalue multiplicities are not clear, similar to the exact Lanczos process, see [12].
- It is possible to choose matrices that make the process slow, especially if A has several very close but unequal eigenvalues, see [12]. This results in one or more eigenpairs of A converging again and again before the desired solutions are obtained. These repeats create the $m_k \times m_k$ matrix $P_1^H T_k P_1$ in (16), and the resulting m_k extra steps cause the slow down. The converged eigenvalues of $P_1^H T_k P_1$ are copies of previously obtained eigenvalues of A.
- Therefore in using the Lanczos process and related algorithms it is important to understand the problem, and where possible attempt to limit such slowdown, usually by some pre-conditioning of the matrix A.
- The Golub-Kahan bidiagonalization (GKB) of a rectangular matrix, see [4], [5, Sect. 10.4.1], has many important uses, see for example, [3,14–16], and can be modelled as a Lanczos process with a specially structured $A = A^H$, see for example [5, Sect. 10.4.3]. Presumably the present analysis can be extended to analyze the GKB.

8 Computational Results for Solutions of Equations

It is easiest to understand plots of solutions of equations, so we just give these.

In Fig. 2 we give an example of Lanczos-CG applied to a 1281 × 1281 finite element matrix (`A = gallery('wathen',20,20);` in MATLAB™) with x elements chosen uniformly random in $[-1, 1]$, and taking $b = Ax$. Here A has 2-norm condition number $\chi_2(A) = 1.8964 \times 10^3$, $\|x\|_2 = 20.4013$, $\|b\|_2 = 1.7280 \times 10^3$. The A-norm of the error $[(x-x_k)^T A(x-x_k)]^{1/2}$ and the residual norm $\|b-Ax_k\|_2$ reach their ultimate precision in about 320 steps, far less than the full 1281 steps. We plot $\|Q_{22}^{(k)}\|_2$ via $Q_{22}^{(k)} = \mathcal{P}_1 \cdots \mathcal{P}_k$, see (12). It does not decrease even well after $k = 1281$. Nevertheless, backward stable results are still available.

For Fig. 3 we chose `A = gallery('prolate',20,.25);` in MATLAB™, a 20×20 symmetric positive definite ill-conditioned Toeplitz matrix that gives CG and the Lanczos process trouble, delaying convergence by many steps by finding extreme eigenvalues again and again. We produced x and b in a similar manner to Fig. 2. Here A is very badly conditioned with $\chi_2(A) = 1.2688 \times 10^{14}$, $\|x\|_2 = 3.1443$, $\|b\|_2 = 2.2557$. The A-norm of the error and the residual norm (which we usually measure) take many more than the ideal 20 steps, not reaching their ultimate precision until about $k = 110$ (where $\|Q_{22}^{(k)}\|_2$ is far from negligible), while $\|Q_{22}^{(k)}\|_2$ decreases irregularly but fairly consistently after $k = 65$.

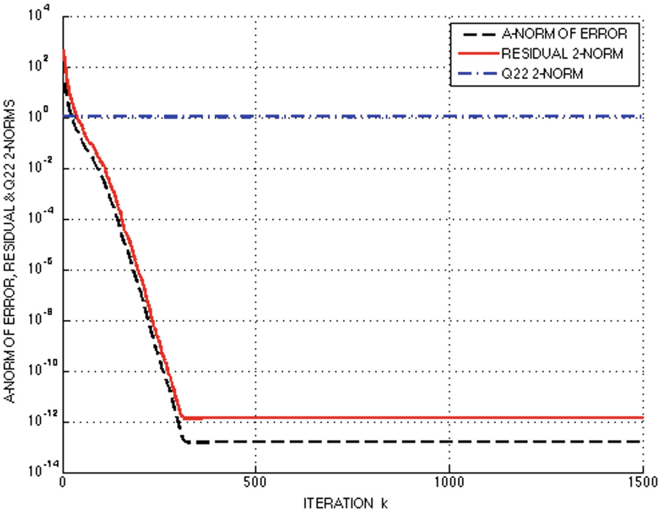

Fig. 2. Solving $Ax = b$ via Lanczos-CG on a 1281 × 1281 finite element matrix.

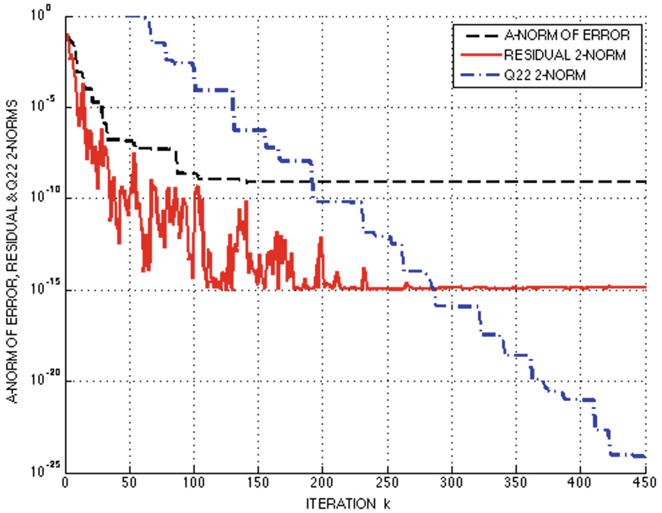

Fig. 3. Solving $Ax = b$ via Lanczos-CG on a 20 × 20 prolate matrix.

9 Conclusions

For a matrix $A = A^H$ the tridiagonalization process of Cornelius Lanczos was initially dismissed because of loss of orthogonality, but is now the basis for some of our most useful algorithms for large sparse matrix computations. The process is useful for the eigenproblem and solution of linear systems. It can lose orthogonality and even linear independence of the computed vectors, so that the computed tridiagonal matrix can develop indefinitely. But when correctly implemented as a 3-term recurrence, see e.g., [5, Sect. 10.1.2], or better still as two 2-term recurrences, see [9], [12, (1.1)], it makes available backward stable solutions to both the eigenproblem and solution of equations. It can provide answers in far fewer than the expected number of iterations, see Fig. 2, a 1281 × 1281 solution of equations problem, or far more, see Fig. 3, a 20×20 solution of equations problem. The Lanczos process is a powerful tool, but many large sparse solution of equations problems require careful preconditioning to speed up the computational solution.

It was shown in [9] that the Lanczos process does not start to lose orthogonality until the first eigenpair has converged, and so the process can be very useful when only a few eigenvalues are required. But if many are required then the possible loss of linear independence of the computed vectors can slow the process significantly. Nevertheless it was shown in (17) and (18) that when A has no multiple eigenvalues and $Q_{22}^{(k)} = 0$, then a set of n backward stable eigenvalues of A are available from n backward stable eigenvalues of T_k, so that there is never any significant loss of accuracy. Similar available accuracy was shown in [12] when A has multiple eigenvalues, but there $\|Q_{22}^{(k)}\|_F$ does not necessarily decrease beyond a certain point. The intermediate case where A has several very close but unequal eigenvalues can slow the process down greatly.

References

1. Björck, Å., Paige, C.C.: Loss and recapture of orthogonality in the modified Gram-Schmidt algorithm. SIAM J. Matrix Anal. Appl. **13**, 176–190 (1992). https://doi.org/10.1137/0613015
2. Davis, C., Kahan, W.M.: Some new bounds on perturbations of subspaces. Bull. Am. Math. Soc. **75**(4), 863–868 (1969). https://doi.org/10.1090/S0002-9904-1969-12330-X
3. Fong, D., Saunders, M.A.: An iterative algorithm for sparse least-squares problems. SIAM J. Sci. Comput. **33**(5), 2950–2971 (2011). https://doi.org/10.1137/10079687X
4. Golub, G.H., Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix. SIAM J. Numer. Anal. **2**, 205–224 (1965). https://doi.org/10.1137/0702016
5. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th ed., The Johns Hopkins University Press, Baltimore (2013). (December 2012) ISBN 9781421407944
6. Hestenes, M., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**, 409–436 (1952). https://doi.org/10.6028/jres.049.044
7. Lanczos, C.: An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. J. Res. Natl. Bur. Stand. **45**, 255–282 (1950). https://doi.org/10.6028/jres.045.026
8. Lanczos, C.: Solution of systems of linear equations by minimized iterations. J. Res. Natl. Bur. Stand. **49**, 33–53 (1952). https://doi.org/10.6028/jres.049.006
9. Paige, C.C.: Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem. Linear Algebra Appl. **34**, 235–258 (1980). https://doi.org/10.1016/0024-3795(80)90167-6

10. Paige, C.C.: A useful form of unitary matrix obtained from any sequence of unit 2-norm n-vectors. SIAM J. Matrix Anal. Appl. **31**, 565–583 (2009). https://doi.org/10.1137/080725167
11. Paige, C.C.: An augmented stability result for the Lanczos Hermitian matrix tridiagonalization process. SIAM J. Matrix Anal. Appl. **31**, 2347–2359 (2010). https://doi.org/10.1137/090761343
12. Paige, C.C.: Accuracy of the Lanczos process for the eigenproblem and solution of equations. SIAM J. Matrix Anal. Appl. (submitted)
13. Paige, C.C., Rozložník, M., Strakoš, Z.: Modified gram-schmidt (MGS), least squares, and backward stability of MGS-GMRES. SIAM J. Matrix Anal. Appl. **28**, 264–284 (2006). https://doi.org/10.1137/050630416
14. Paige, C.C., Saunders, M.A.: LSQR: an algorithm for sparse linear equations and sparse least squares. ACM Trans. Math. Softw. **8**, 43–71 (1982). https://doi.org/10.1145/355984.355989
15. Paige, C.C., Strakoš, Z.: Scaled total least squares fundamentals. Numer. Math. **91**, 117–146 (2002). https://doi.org/10.1007/s002110100314
16. Paige, C.C., Strakoš, Z.: Core problems in linear algebraic systems. SIAM J. Matrix Anal. Appl. **27**, 861–875 (2006). https://doi.org/10.1137/040616991
17. Paige, C.C., Wülling, W.: Properties of a unitary matrix obtained from a sequence of normalized vectors. SIAM J. Matrix Anal. Appl. **35**, 526–545 (2014). https://doi.org/10.1137/120897687
18. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**, 856–869 (1986). https://doi.org/10.1137/0907058
19. Stewart, G.W.: On the perturbation of pseudo-inverses, projections, and linear least squares problems. SIAM Rev. **19**, 634–662 (1977). https://doi.org/10.1137/1019104