RESTARTING THE NON-SYMMETRIC LANCZOS ALGORITHM VIA THE IMPLICITLY SHIFTED LR ALGORITHM*

P. S. NEGI† AND C. ARRATIA‡

Abstract. The shifted QR iteration is used as a restart procedure for the Arnoldi method for the calculation of a few eigenvalues of a large matrix. We show that the underlying idea can be utilized in much the same manner via the shifted LR iteration to create a restart procedure for the non-symmetric Lanczos algorithm for eigenvalue calcuations. Additionally, we show that the (shifted) LR iteration can be performed implicitly in a manner similar to the Francis' algorithm, resulting in a bulge-chase type procedure which does not require the explicit construction of the full lower and upper triangular matrices.

Key words. LR algorithm, unsymmetric Lanczos, implicit restart

MSC codes. 68Q25, 68R10, 68U05

1. Introduction.

The Arnoldi iteration [1] is a popular Krylov space method for calculating a few eigenvalues of a large matrix. The method relies on the generation of a sequence of Krylov vectors which determine the subspace within which approximations of the eigenvalue-eigenvector pairs are obtained. Depending on the accuracy and number of eigenpair approximations needed, the Krylov space size can become exceedingly large so that the quality of the results may be limited by the available memory. Sorensen [9] introduced an elegant procedure for restarting the Arnoldi factorization based on polynomial filters, which are applied through the implicitly shifted QR iterations on the reduced Hessenberg matrix obtained through the Arnoldi method. In particular, the use of exact shifts was shown to be successful in the convergence process of the eigenspace[9] of the specified eigenvalues. The method has subsequently found widespread application through the ARPACK library [4]. The use of QR iterations ensures that the reduced matrix preserves its Hessenberg structure through the transforms that make up the restart process. If the underlying matrix is symmetric, the Arnoldi iteration reduces to the Lanczos algorithm and, the Hessenberg matrix reduces to a symmetric tridiagonal matrix. The QR iteration preserves the symmetric tridiagonal structure as well and, as pointed out by Sorensen [9], the implicit restart process applies equally well for the Lanczos method for symmetric matrices.

One would like to extend this procedure to the case of the non-symmetric Lanczos method. However, the reduced matrix that one obtains is a non-symmetric tridiagonal matrix, with the tridiagonal structure being the result of the recurrence relations of the Lanczos algorithm [8]. Since the QR iterations do not preserve the banded structure of non-symmetric matrices, a straightforward application of the restart procedure propounded by Sorensen will lead to a loss of this tridiagonal structure of the reduced matrix (the Hessenberg structure will still be preserved). This loss of structure can be circumvented if one looks to the predecessor of the QR algorithm namely, the LR algorithm proposed by Rutihauser [6, 7, 5], which has the attractive property of preserving the band structure of a matrix. This property was already pointed out by Rutihauser in [6] where the banded matrices were referred to as striped matrices.

*Submitted to the editors DATE.

Funding: This work was funded by the Fog Research Institute under contract no. FRI-454.

†Imagination Corp., Chicago, IL (prabal.negi@su.se, http://www.imag.com/~ddoe/).

‡Department of Applied Mathematics, Fictional University, Boise, ID (cristobal.arratia@su.se).

As we will show in the next section, shifted LR transforms is the appropriate generalization of the restart procedure to the case of non-symmetric Lanczos iteration. The process would necessarily require refining both the right as well as the left Krylov spaces simultaneously.

The rest of the paper is organized as follows. In the next section we start with the introduction of the non-symmetric Lanczos iteration and then develop the restart procedure. We also show that the LR algorithm can be implemented as a bulge-chase method, similar to the Francis' algorithm. In section 3 we apply the restart process to the Grcar matrix, and make some concluding remarks in section 4.

2. Non-symmetric Lanczos.

Lanczos first introduced his algorithm in [3] as a method for tridiagonalizing a matrix, but also realized that the method could be used iteratively to find eigenvalues. For an arbitry matrix A, the method generates a pair of Krylov subspaces $\{v_1, \ldots, v_j\}$ and $\{w_1, \ldots, w_j\}$, through repeated action of A and A^H respectively. We refer to these as the right and left Krylov spaces respectively and they satisfy the biorthogonality relation $w_i^H v_j = \delta_{ij}$. The two subspaces are generated through the following recurrence relations:

$$\delta_{j+1} v_{j+1} = A v_j - \alpha_j v_j - \beta_j v_{j-1} \tag{2.1a}$$

$$\beta_{j+1} w_{j+1} = A^H w_j - \alpha_j w_j - \delta_j w_{j-1} \tag{2.1b}$$

which in matrix form can be written as

$$A V_m = V_m T_m + v_{m+1} e_m^T \tag{2.2a}$$

$$A^H W_m = W_m T_m^H + w_{m+1} e_m^T \tag{2.2b}$$

3. Main results.

We interleave text filler with some example theorems and theorem-like items.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Here we state our main result as Theorem 3.1; the proof is deferred to **??**.

THEOREM 3.1 (LDL^T Factorization [2]). *If $A \in \mathbb{R}^{n\times n}$ is symmetric and the principal submatrix $A(1:k, 1:k)$ is nonsingular for $k = 1 : n-1$, then there exists a unit lower triangular matrix L and a diagonal matrix*

$$D = \operatorname{diag}(d_1, \ldots, d_n)$$

such that $A = LDL^T$. The factorization is unique.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

THEOREM 3.2 (Mean Value Theorem). *Suppose f is a function that is continuous on the closed interval $[a, b]$. and differentiable on the open interval (a, b). Then there exists a number c such that $a < c < b$ and*

$$f'(c) = \frac{f(b) - f(a)}{b - a}.$$

In other words,

$$f(b) - f(a) = f'(c)(b - a).$$

Observe that Theorems 3.1 and 3.2 and Corollary 3.3 correctly mix references to multiple labels.

COROLLARY 3.3. *Let $f(x)$ be continuous and differentiable everywhere. If $f(x)$ has at least two roots, then $f'(x)$ must have at least one root.*

Proof. Let a and b be two distinct roots of f. By Theorem 3.2, there exists a number c such that

$$f'(c) = \frac{f(b) - f(a)}{b - a} = \frac{0 - 0}{b - a} = 0. \qquad \square$$

Note that it may require two LaTeX compilations for the proof marks to show.

Display matrices can be rendered using environments from `amsmath`:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad C = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{3.1}$$

Equation (3.1) shows some example matrices.

We calculate the Fréchet derivative of F as follows:

$$\begin{align} F'(U, V)(H, K) &= \langle R(U, V), H\Sigma V^T + U\Sigma K^T - P(H\Sigma V^T + U\Sigma K^T)\rangle \tag{3.2a} \\ &= \langle R(U, V), H\Sigma V^T + U\Sigma K^T\rangle \notag \\ &= \langle R(U, V)V\Sigma^T, H\rangle + \langle \Sigma^T U^T R(U, V), K^T\rangle. \tag{3.2b} \end{align}$$

Equation (3.2a) is the first line, and (3.2b) is the last line.

4. Algorithm. Sed gravida lectus ut purus. Morbi laoreet magna. Pellentesque eu wisi. Proin turpis. Integer sollicitudin augue nec dui. Fusce lectus. Vivamus faucibus nulla nec lacus. Integer diam. Pellentesque sodales, enim feugiat cursus volutpat, sem mauris dignissim mauris, quis consequat sem est fermentum ligula. Nullam justo lectus, condimentum sit amet, posuere a, fringilla mollis, felis. Morbi nulla nibh, pellentesque at, nonummy eu, sollicitudin nec, ipsum. Cras neque. Nunc augue. Nullam vitae quam id quam pulvinar blandit. Nunc sit amet orci. Aliquam erat elit, pharetra nec, aliquet a, gravida in, mi. Quisque urna enim, viverra quis, suscipit quis, tincidunt ut, sapien. Cras placerat consequat sem. Curabitur ac diam. Curabitur diam tortor, mollis et, viverra ac, tempus vel, metus.

Our analysis leads to the algorithm in Algorithm 4.1.

Curabitur ac lorem. Vivamus non justo in dui mattis posuere. Etiam accumsan ligula id pede. Maecenas tincidunt diam nec velit. Praesent convallis sapien ac est. Aliquam ullamcorper euismod nulla. Integer mollis enim vel tortor. Nulla sodales placerat nunc. Sed tempus rutrum wisi. Duis accumsan gravida purus. Nunc nunc. Etiam facilisis dui eu sem. Vestibulum semper. Praesent eu eros. Vestibulum tellus nisl, dapibus id, vestibulum sit amet, placerat ac, mauris. Maecenas et elit ut erat placerat dictum. Nam feugiat, turpis et sodales volutpat, wisi quam rhoncus neque, vitae aliquam ipsum sapien vel enim. Maecenas suscipit cursus mi.

Algorithm 4.1 Build tree

Define $P := T := \{\{1\}, \ldots, \{d\}\}$
while $\#P > 1$ **do**
  Choose $C' \in \mathcal{C}_p(P)$ with $C' := \operatorname{argmin}_{C \in \mathcal{C}_p(P)} \varrho(C)$
  Find an optimal partition tree $T_{C'}$
  Update $P := (P \backslash C') \cup \{\bigcup_{t \in C'} t\}$
  Update $T := T \cup \{\bigcup_{t \in \tau} t : \tau \in T_{C'} \backslash \mathcal{L}(T_{C'})\}$
end while
return T

5. Experimental results. Quisque facilisis auctor sapien. Pellentesque gravida hendrerit lectus. Mauris rutrum sodales sapien. Fusce hendrerit sem vel lorem. Integer pellentesque massa vel augue. Integer elit tortor, feugiat quis, sagittis et, ornare non, lacus. Vestibulum posuere pellentesque eros. Quisque venenatis ipsum dictum nulla. Aliquam quis quam non metus eleifend interdum. Nam eget sapien ac mauris malesuada adipiscing. Etiam eleifend neque sed quam. Nulla facilisi. Proin a ligula. Sed id dui eu nibh egestas tincidunt. Suspendisse arcu.

Figure 1 shows some example results. Additional results are available in the supplement in Table 1.

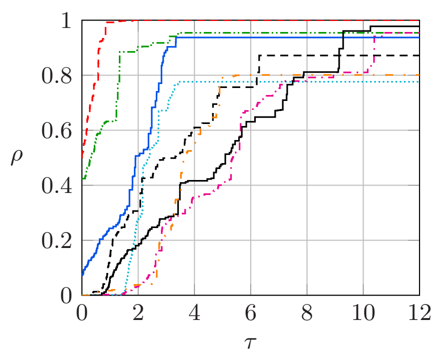

Fig. 1. *Example figure using external image files.*

Table 1 shows additional supporting evidence.

Table 1
Example table.

Species	**Mean**	**Std. Dev.**
1	3.4	1.2
2	5.4	0.6
3	7.4	2.4
4	9.4	1.8

Maecenas dui. Aliquam volutpat auctor lorem. Cras placerat est vitae lectus. Curabitur massa lectus, rutrum euismod, dignissim ut, dapibus a, odio. Ut eros erat, vulputate ut, interdum non, porta eu, erat. Cras fermentum, felis in porta congue, velit leo facilisis odio, vitae consectetuer lorem quam vitae orci. Sed ultrices, pede eu placerat auctor, ante ligula rutrum tellus, vel posuere nibh lacus nec nibh. Maecenas laoreet dolor at enim. Donec molestie dolor nec metus. Vestibulum libero. Sed quis erat. Sed tristique. Duis pede leo, fermentum quis, consectetuer eget, vulputate sit amet, erat.

6. Discussion of $\boldsymbol{Z = X \cup Y}$. Curabitur nunc magna, posuere eget, venenatis eu, vehicula ac, velit. Aenean ornare, massa a accumsan pulvinar, quam lorem laoreet purus, eu sodales magna risus molestie lorem. Nunc erat velit, hendrerit quis, malesuada ut, aliquam vitae, wisi. Sed posuere. Suspendisse ipsum arcu, scelerisque nec, aliquam eu, molestie tincidunt, justo. Phasellus iaculis. Sed posuere lorem non ipsum. Pellentesque dapibus. Suspendisse quam libero, laoreet a, tincidunt eget, consequat at, est. Nullam ut lectus non enim consequat facilisis. Mauris leo. Quisque pede ligula, auctor vel, pellentesque vel, posuere id, turpis. Cras ipsum sem, cursus et, facilisis ut, tempus euismod, quam. Suspendisse tristique dolor eu orci. Mauris mattis. Aenean semper. Vivamus tortor magna, facilisis id, varius mattis, hendrerit in, justo. Integer purus.

7. Conclusions. Some conclusions here.

Appendix A. An example appendix. Aenean tincidunt laoreet dui. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Integer ipsum lectus, fermentum ac, malesuada in, eleifend ut, lorem. Vivamus ipsum turpis, elementum vel, hendrerit ut, semper at, metus. Vivamus sapien tortor, eleifend id, dapibus in, egestas et, pede. Pellentesque faucibus. Praesent lorem neque, dignissim in, facilisis nec, hendrerit vel, odio. Nam at diam ac neque aliquet viverra. Morbi dapibus ligula sagittis magna. In lobortis. Donec aliquet ultricies libero. Nunc dictum vulputate purus. Morbi varius. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In tempor. Phasellus commodo porttitor magna. Curabitur vehicula odio vel dolor.

LEMMA A.1. *Test Lemma.*

Acknowledgments. We would like to acknowledge the assistance of volunteers in putting together this example manuscript and supplement.

REFERENCES

[1] W. E. ARNOLDI, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quarterly of applied mathematics, 9 (1951), pp. 17–29.
[2] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 4th ed., 2013.
[3] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, (1950).
[4] R. B. LEHOUCQ, D. C. SORENSEN, AND C. YANG, *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, SIAM, 1998.
[5] H. RUTIHAUSER, *Lectures on Numerical Mathematics*, Birkhauser, 1990.
[6] H. RUTISHAUSER, *Solution of eigenvalue problems with the lr-transformation*, National Bureau of Standards, Applied Mathematics Series, 49 (1958), pp. 47–81.
[7] H. RUTISHAUSER AND H. SCHWARZ, *Thelr transformation method for symmetric matrices*, Numerische Mathematik, 5 (1963), pp. 273–289.
[8] Y. SAAD, *The lanczos biorthogonalization algorithm and other oblique projection methods for solving large unsymmetric systems*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 485–506.
[9] D. C. SORENSEN, *Implicit application of polynomial filters in ak-step arnoldi method*, Siam journal on matrix analysis and applications, 13 (1992), pp. 357–385.