A SYMMETRIC BAND LANCZOS PROCESS BASED ON COUPLED RECURRENCES AND SOME APPLICATIONS*

ZHAOJUN BAI† AND AND ROLAND W. FREUND‡

Abstract. The symmetric band Lanczos process is an extension of the classical Lanczos algorithm for symmetric matrices and single starting vectors to multiple starting vectors. After n iterations, the symmetric band Lanczos process has generated an $n \times n$ projection, $\mathbf{T}_n^{\mathrm{s}}$, of the given symmetric matrix onto the n-dimensional subspace spanned by the first n Lanczos vectors. This subspace is closely related to the n-th block Krylov subspace induced by the given symmetric matrix and the given block of multiple starting vectors. The standard algorithm produces the entries of $\mathbf{T}_n^{\mathrm{s}}$ directly. In this paper, we propose a variant of the symmetric band Lanczos process that employs coupled recurrences to generate the factors of an $\mathrm{LDL}^{\mathrm{T}}$ factorization of a closely related $n \times n$ projection, rather than $\mathbf{T}_n^{\mathrm{s}}$. This is done in such a way that the factors of the $\mathrm{LDL}^{\mathrm{T}}$ factorization inherit the "fish-bone" structure of $\mathbf{T}_n^{\mathrm{s}}$. Numerical examples from reduced-order modeling of large electronic circuits and algebraic eigenvalue problems demonstrate that the proposed variant of the band Lanczos process based on coupled recurrences is more robust and accurate than the standard algorithm.

Key words. symmetric matrix, block-Krylov subspace, multiple starting vectors, orthogonal basis, projection, reduced-order modeling, passivity, circuit simulation, eigenvalue problem

AMS subject classifications. 65F15, 65F30, 65L80, 93B40

1. Introduction. In recent years, suitable extensions of the classical Lanczos process [20] for single right and left starting vectors to multiple right and left starting vectors have proven to be powerful tools for reduced-order modeling of large-scale multi-input multi-output linear dynamical systems. The most general such Lanczos-type algorithm is the one proposed in [1]. Given a square matrix $\mathbf{A}$ and two blocks of right and left starting vectors, the algorithm in [1] generates two sequences of biorthogonal basis vectors for the right and left block-Krylov subspaces induced by the given data. The algorithm can handle the most general case of right and left starting blocks of arbitrary sizes, say m and p. In [9, 10], it was shown that this Lanczos-type algorithm can be employed to generate reduced-order models of m-input p-output linear dynamical systems and that these reduced-order models correspond to matrix-Padé approximants of the system's transfer function. The resulting computational procedure is called the MPVL (Matrix-Padé Via Lanczos) algorithm. For recent surveys on MPVL, related methods, and their use in VLSI circuit simulation, we refer the reader to [11, 13].

In circuit simulation, an important special case is linear dynamical systems that describe large so-called RCL subcircuits consisting of only resistors, capacitors, and inductors. In this case, the matrix $\mathbf{A}$ is symmetric and the blocks of right and left starting vectors are identical. The SyMPVL algorithm [14, 15] is a variant of MPVL that exploits this symmetry and as a result, only requires half the computational costs and storage of MPVL. The Lanczos-type algorithm underlying SyMPVL is essentially the symmetric band Lanczos process first proposed in [23], with some additional

*Numerical Analysis Manuscript No. 00–3–04, Bell Laboratories, Murray Hill, New Jersey, May 2000. Available on WWW at `http://cm.bell-labs.com/cs/doc/00`.

†Department of Computer Science, University of California at Davis, One Shields Avenue, Davis, California 95616 (bai@cs.ucdavis.edu). Most of this work was done while this author was on sabbatical at Bell Laboratories, Lucent Technologies, Murray Hill, New Jersey.

‡Bell Laboratories, Lucent Technologies, 700 Mountain Avenue, Room 2C–525, Murray Hill, New Jersey 07974–0636 (freund@research.bell-labs.com).

features, such as deflation of Krylov vectors; see [12]. It generates a sequence of $n \times n$ projections $\mathbf{T}_n^{\mathrm{s}}$, $n \geq 1$, of the symmetric matrix $\mathbf{A}$ onto the n-dimensional subspace spanned by the first n Lanczos vectors. This subspace is closely related to the n-th block Krylov subspace induced by $\mathbf{A}$ and the given block of multiple starting vectors. As the classical Lanczos process, the symmetric band Lanczos process generates the entries of $\mathbf{T}_n^{\mathrm{s}}$ directly.

In this paper, we propose a new variant of the symmetric band Lanczos process that employs coupled recurrences to produce the factors of an $\mathrm{LDL}^{\mathrm{T}}$ factorization of an $n \times n$ matrix, $\mathbf{T}_n$, closely related to $\mathbf{T}_n^{\mathrm{s}}$, rather than $\mathbf{T}_n^{\mathrm{s}}$ itself. This work was motivated mainly by numerical experiences with the SyMPVL algorithm applied to RC subcircuits. In this case, the matrix $\mathbf{A}$ is symmetric positive semi-definite, and in order to ensure passivity of the reduced-order models produced by SyMPVL, it is crucial that the projection of $\mathbf{A}$ preserves the positive semi-definiteness of $\mathbf{A}$. In exact arithmetic, the projection $\mathbf{T}_n^{\mathrm{s}}$ of a positive semi-definite matrix $\mathbf{A}$ is guaranteed to be positive semi-definite. However, in finite-precision arithmetic, round-off may cause the computed projection $\mathbf{T}_n^{\mathrm{s}}$ to be slightly indefinite; see [4] for such examples. These problems are clearly due to the direct computation of $\mathbf{T}_n^{\mathrm{s}}$. Indeed, when the projection is obtained via the $\mathrm{LDL}^{\mathrm{T}}$ factorization produced by the proposed symmetric band Lanczos process based on coupled recurrences, the computed projection preserves the positive semi-definiteness of $\mathbf{A}$. In addition, the new variant also produces more accurate reduced-order models with the same number of iterations.

The idea of enforcing positive semi-definiteness of a matrix by generating it via a factorization is of course not new. Indeed, the very same issue arises in Kalman filtering where numerical round-off in the standard update formula of the covariance matrices may result in slightly indefinite matrices. The remedy there is to update suitable factors, rather than the covariance matrices, resulting in so-called square-root filtering; see, e.g., [3, Chapter 6.5]. The idea of square-root filtering seems to have originated with James E. Potter in 1962; see [3, Chapter 6.5] and [6, Section 13.7]. The use of square-root filtering was crucial in eliminating problems in the Apollo navigation systems caused by numerical round-off; see [6] and the references given there. The same issue also arises in the computation of positive semi-definite solutions of Lyapunov equations; see [19] and the references therein. In [19], an algorithm is proposed that directly generates the Cholesky factor of the solution matrix, rather than the solution matrix.

Finally, in the context of employing the classical Lanczos process for single starting vectors to the solution of systems of linear equations, it has been observed that coupled two-term recurrences are more robust than the mathematically-equivalent three-term recurrences; see, e.g., [16] and the references given there. Some recent analysis of this phenomenon can be found in [18].

The remainder of this paper is organized as follows. In § 2, we describe the governing equations of the symmetric band Lanczos process based on coupled recurrences, and discuss connections with the standard algorithm. In § 3, we present a complete statement of the algorithm based on coupled recurrences, and establish some of its properties. In § 4 and § 5, we discuss applications of the new variant of the symmetric band Lanczos process to reduced-order modeling and to eigenvalue computations, respectively, and we report results of numerical experiments. In § 6, we make some concluding remarks.

Throughout this article, we use boldface letters to denote vectors and matrices. Unless stated otherwise, vectors and matrices are assumed to have real entries. As usual, $\mathbf{M}^{\mathrm{T}} = [\, m_{kj} \,]$ denotes the transpose of the matrix $\mathbf{M} = [\, m_{jk} \,]$, and $\mathbf{M} \geq \mathbf{0}$ ($\mathbf{M} > \mathbf{0}$) means that $\mathbf{M} = \mathbf{M}^{\mathrm{T}}$ is symmetric positive semi-definite (symmetric positive definite). The vector norm $\|\mathbf{x}\| := \sqrt{\mathbf{x}^{\mathrm{T}}\,\mathbf{x}}$ is always the Euclidean norm, and $\|\mathbf{M}\| := \max_{\|\mathbf{x}\|=1} \|\mathbf{M}\,\mathbf{x}\|$ is the corresponding induced matrix norm. We use $\mathbf{I}_n$ to denote the $n \times n$ identity matrix and $\mathbf{0}_{n\times m}$ to denote the $n \times m$ zero matrix; we will omit these indices whenever the actual dimensions of $\mathbf{I}$ and $\mathbf{0}$ are apparent from the context. The sets of real and complex numbers are denoted by $\mathbb{R}$ and $\mathbb{C}$, respectively. For $s \in \mathbb{C}$, $\mathrm{Re}(s)$ is the real part of s.

2. The symmetric band Lanczos process based on coupled recurrences. In this section, we describe the governing equations of the symmetric band Lanczos process based on coupled recurrences, and discuss connections with the standard algorithm.

2.1. Preliminaries. In the sequel, we assume that

$$\mathbf{A} = \mathbf{A}^{\mathrm{T}} \in \mathbb{R}^{N\times N} \quad \text{and} \quad \mathbf{R} = [\, \mathbf{r}_1 \quad \mathbf{r}_2 \quad \cdots \quad \mathbf{r}_m \,] \in \mathbb{R}^{N\times m} \tag{2.1}$$

are given matrices. The columns of $\mathbf{R}$ are the multiple starting vectors. The purpose of the symmetric band Lanczos process is to generate orthogonal basis vectors for the subspaces spanned by the leading columns of the *block-Krylov matrix*,

$$\mathbf{K}(\mathbf{A},\mathbf{R}) := [\, \mathbf{R} \quad \mathbf{A}\,\mathbf{R} \quad \mathbf{A}^2\,\mathbf{R} \quad \cdots \quad \mathbf{A}^{N-1}\,\mathbf{R} \,], \tag{2.2}$$

associated with the matrices (2.1), and to compute the projection of $\mathbf{A}$ onto these subspaces. A proper definition of these subspaces is necessarily quite involved; see the discussion in [1]. The main reason is that the columns of the matrix $\mathbf{K}(\mathbf{A},\mathbf{R})$ in (2.2) are all vectors of length N, and thus at most N of them are linearly independent. As a result, in the symmetric band Lanczos process, one needs to perform so-called *deflation* of linearly dependent or in some sense almost linearly dependent vectors. As we will describe below, the symmetric band Lanczos process has a very simple built-in deflation procedure. In exact arithmetic, only the linearly dependent vectors have to be removed, and we refer to this as *exact* deflation. In the case of exact deflation, the subspaces spanned by the leading columns of the matrix (2.2) can be easily defined, and we will do so in § 2.5 below.

2.2. Governing equations. The symmetric band Lanczos algorithm is an iterative procedure. After n iterations, the algorithm has generated the first n *Lanczos vectors*,

$$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \in \mathbb{R}^N, \tag{2.3}$$

as well as the $m_{\mathrm{c}} = m_{\mathrm{c}}(n)$ vectors,

$$\hat{\mathbf{v}}_{n+1}, \hat{\mathbf{v}}_{n+2}, \ldots, \hat{\mathbf{v}}_{n+m_{\mathrm{c}}} \in \mathbb{R}^N, \tag{2.4}$$

that are the candidates for the next m_{c} Lanczos vectors, $\mathbf{v}_{n+1}, \mathbf{v}_{n+2}, \ldots, \mathbf{v}_{n+m_{\mathrm{c}}}$. Here, $m_{\mathrm{c}} = m_{\mathrm{c}}(n)$ is the *current block size* defined as follows. In the initialization phase, i.e., for $n = 0$, $m_{\mathrm{c}} := m$ is set to be the number of starting vectors in (2.1), and the candidate vectors (2.4) are set to be the starting vectors from (2.1), i.e., $\hat{\mathbf{v}}_i = \mathbf{r}_i$, $1 \leq i \leq m$. Within the algorithm, m_{c} is then reduced by one every time a deflation occurs. Moreover, in the algorithm, the vectors (2.4) are used to detect and perform deflation. More precisely, it can be shown that an exact deflation occurs during the

$(n+1)$-st iteration of the algorithm if, and only if, $\hat{\mathbf{v}}_{n+1} = \mathbf{0}$. Therefore, in the algorithm, one simply checks if the first of the candidate vectors is the zero vector or in some sense close to the zero vector and if so, one performs deflation by removing that first candidate vector.

In addition to (2.3), the symmetric band Lanczos process based on coupled recurrences also generates a second set of vectors,

$$\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n \in \mathbb{R}^N, \tag{2.5}$$

that span the same subspaces as (2.3), i.e.,

$$\operatorname{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_j\} = \operatorname{span}\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_j\} \quad \text{for all} \quad 1 \le j \le n. \tag{2.6}$$

The vectors (2.3)–(2.5) are constructed to satisfy (in exact arithmetic) the following orthogonality conditions:

$$\mathbf{V}_n^{\mathrm{T}} \mathbf{V}_n = \mathbf{I}_n, \tag{2.7}$$

$$\mathbf{V}_n^{\mathrm{T}} \left[\hat{\mathbf{v}}_{n+1} \quad \hat{\mathbf{v}}_{n+2} \quad \cdots \quad \hat{\mathbf{v}}_{n+m_c} \right] = \mathbf{0}, \tag{2.8}$$

$$\mathbf{P}_n^{\mathrm{T}} \mathbf{A} \mathbf{P}_n = \mathbf{\Delta}_n := \operatorname{diag}(\delta_1, \delta_2, \ldots, \delta_n). \tag{2.9}$$

Here and in the sequel,

$$\mathbf{V}_n := [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_n] \quad \text{and} \quad \mathbf{P}_n := [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \cdots \quad \mathbf{p}_n] \tag{2.10}$$

denote the matrices whose columns are just the vectors (2.3) and (2.5), respectively. Furthermore, in (2.9), we assume that $\delta_i \neq 0$ for all i. This implies that

$$\mathbf{\Delta}_n \quad \text{is nonsingular.} \tag{2.11}$$

Note that, by (2.7), the Lanczos vectors (2.3) are an orthonormal basis of the subspaces (2.6), while, by (2.9), the vectors (2.5) are an $\mathbf{A}$-orthogonal basis of (2.6).

The recurrence relations that are used in the algorithm to generate the vectors (2.3)–(2.5) can be stated as follows. The first m_1 vectors in (2.3) satisfy a relation of the form

$$\mathbf{R} = \mathbf{V}_{m_1} \boldsymbol{\rho}_{m_1} + \widehat{\mathbf{V}}_0^{\mathrm{df}}. \tag{2.12}$$

Here, $m_1 \ (\le m)$ denotes the number of columns of the block $\mathbf{R}$ that have not been deflated, the matrix $\widehat{\mathbf{V}}_0^{\mathrm{df}} \in \mathbb{R}^{N \times m}$ contains the $m - m_1$ deflated starting vectors and m_1 zero vectors as columns, and

$$\boldsymbol{\rho}_{m_1} = [\rho_{i,j}]_{i=1,2,\ldots,m_1;\, j=1,2,\ldots,m} \in \mathbb{R}^{m_1 \times m} \tag{2.13}$$

is an upper triangular matrix whose entries are chosen such that the columns of $\mathbf{V}_{m_1}$ are orthonormal. The main relations for generating the vectors (2.3)–(2.5) are coupled recurrences that can be summarized as follows:

$$\mathbf{A} \mathbf{P}_n = \mathbf{V}_n \mathbf{L}_n \mathbf{\Delta}_n + [\, \underbrace{\mathbf{0} \quad \mathbf{0} \quad \cdots \quad \mathbf{0}}_{n-m_c} \quad \underbrace{\hat{\mathbf{v}}_{n+1} \quad \hat{\mathbf{v}}_{n+2} \quad \cdots \quad \hat{\mathbf{v}}_{n+m_c}}_{m_c} \,] + \widehat{\mathbf{V}}_n^{\mathrm{df}}, \tag{2.14}$$

$$\mathbf{V}_n = \mathbf{P}_n \mathbf{U}_n. \tag{2.15}$$

Here, $\mathbf{\Delta}_n$ is the diagonal matrix defined in (2.9), and

$$\mathbf{L}_n = [\,\ell_{ij}\,]_{i,j=1,2,\ldots,n} \in \mathbb{R}^{n\times n} \quad \text{and} \quad \mathbf{U}_n = [\,u_{ij}\,]_{i,j=1,2,\ldots,n} \in \mathbb{R}^{n\times n} \tag{2.16}$$

are unit lower and upper triangular matrices, respectively. The entries of the matrices $\mathbf{\Delta}_n$, $\mathbf{L}_n$, and $\mathbf{U}_n$ are chosen such that the vectors (2.3)–(2.5) satisfy the orthogonality conditions (2.7)–(2.9). Furthermore, any candidate vector $\hat{\mathbf{v}}_j$ that was deflated at the j-th iteration, where $1 + m_{\rm c}(j) \le j \le n$, appears as the $(j - m_{\rm c}(j))$-th column of the matrix $\widehat{\mathbf{V}}_n^{\rm df}$ in (2.14); all other columns of $\widehat{\mathbf{V}}_n^{\rm df}$ are zero vectors. In the actual Algorithm 3.1 below, we use the index set $\mathcal{I}$ to record the positions of the potentially nonzero columns of $\widehat{\mathbf{V}}_n^{\rm df}$ due to deflation. If no deflation has occurred or if only exact deflation is performed, then clearly

$$\widehat{\mathbf{V}}_0^{\rm df} = \mathbf{0} \quad \text{and} \quad \widehat{\mathbf{V}}_n^{\rm df} = \mathbf{0} \tag{2.17}$$

in (2.12) and (2.14), respectively.

2.3. The Lanczos matrix. By multiplying (2.9) from the left by $\mathbf{U}_n^{\rm T}$ and from the right by $\mathbf{U}_n$, and by using (2.15), it follows that

$$\mathbf{T}_n := \mathbf{V}_n^{\rm T}\,\mathbf{A}\,\mathbf{V}_n = \mathbf{U}_n^{\rm T}\,\mathbf{\Delta}_n\,\mathbf{U}_n. \tag{2.18}$$

The matrix $\mathbf{T}_n$ defined in (2.18) is the so-called *n-th Lanczos matrix.* It represents the projection of $\mathbf{A}$ onto the subspaces spanned by the Lanczos vectors (2.3). Recall that $\mathbf{U}_n$ is a unit upper triangular matrix and that $\mathbf{\Delta}_n$ is a diagonal matrix. Thus, in view of (2.18), the symmetric band Lanczos process based on coupled recurrences computes the factors of an LDL$^{\rm T}$ decomposition of the Lanczos matrix $\mathbf{T}_n$, rather than $\mathbf{T}_n$ itself.

In the important special case $\mathbf{A} \ge \mathbf{0}$, in view of (2.9), the diagonal entries of $\mathbf{\Delta}_n$ satisfy

$$\delta_i = \mathbf{p}_i^{\rm T}\,\mathbf{A}\,\mathbf{p}_i \ge 0 \quad \text{for all} \quad i, \tag{2.19}$$

and hence $\mathbf{\Delta}_n \ge \mathbf{0}$. Thus, for $\mathbf{A} \ge \mathbf{0}$, generating $\mathbf{T}_n$ via the factorization (2.18) always results in a positive semi-definite matrix.

The triangular matrices $\mathbf{L}_n$ in (2.14) and $\mathbf{U}_n$ in (2.15) are closely related. Indeed, by multiplying (2.14) from the left by $\mathbf{V}_n^{\rm T}$ and from the right by $\mathbf{U}_n$, and by using (2.7), (2.8), and (2.15), we get

$$\mathbf{V}_n^{\rm T}\,\mathbf{A}\,\mathbf{V}_n = (\mathbf{L}_n\,\mathbf{\Delta}_n + \mathbf{\Sigma}_n)\,\mathbf{U}_n, \quad \text{where} \quad \mathbf{\Sigma}_n := \mathbf{V}_n^{\rm T}\,\widehat{\mathbf{V}}_n^{\rm df}. \tag{2.20}$$

By comparing (2.18) and (2.20), it follows that

$$\mathbf{U}_n = \mathbf{L}_n^{\rm T} + \mathbf{\Delta}_n^{-1}\,\mathbf{\Sigma}_n^{\rm T}. \tag{2.21}$$

In particular, if no deflation has occurred or if only exact deflation is performed, then, in view of (2.17), we have $\mathbf{\Sigma}_n = \mathbf{0}$ in (2.20) and thus $\mathbf{U}_n = \mathbf{L}_n^{\rm T}$.

2.4. Structure of the triangular factor $\mathbf{U}_n$. In this subsection, we describe the sparsity structure of the matrix $\mathbf{U}_n$.

The unit upper triangular $\mathbf{U}_n$ consists of a banded part with bandwidth decreasing from $m+1$ to $m_{\rm c}+1$ and a "spiked" part with potentially nonzero elements only in rows with index $i \in \mathcal{I}$ and to the right of the banded part. Recall that the index set $\mathcal{I}$ records deflation. It turns out that in the additive splitting (2.21) of $\mathbf{U}_n$, the first term, $\mathbf{L}_n^{\mathrm{T}}$, is the banded part, while the second term, $\mathbf{\Delta}_n^{-1}\,\mathbf{\Sigma}_n^{\mathrm{T}}$, is the spiked part.

Next, we present an example that illustrates this structure of $\mathbf{U}_n$. Consider the case that $m = 5$ and that the three candidate vectors $\hat{\mathbf{v}}_8$ (when $m_{\mathrm{c}} = 5$), $\hat{\mathbf{v}}_{11}$ (when $m_{\mathrm{c}} = 4$), and $\hat{\mathbf{v}}_{13}$ (when $m_{\mathrm{c}} = 3$) have been deflated. The associated index set is $\mathcal{I} = \{\,3, 7, 10\,\}$. After $n = 15$ iterations, we then have $m_{\mathrm{c}} = 2$, and the matrix $\mathbf{U}_{15}$ has the following structure:

$$\mathbf{U}_{15} = \begin{bmatrix}
1 & \times & \times & \times & \times & \times & & & & & & & & & \\
 & 1 & \times & \times & \times & \times & \times & & & & & & & & \\
 & & 1 & \times & \times & \times & \times & * & * & * & * & * & * & * & * \\
 & & & 1 & \times & \times & \times & \times & & & & & & & \\
 & & & & 1 & \times & \times & \times & \times & & & & & & \\
 & & & & & 1 & \times & \times & \times & \times & & & & & \\
 & & & & & & 1 & \times & \times & \times & * & * & * & * & * \\
 & & & & & & & 1 & \times & \times & \times & & & & \\
 & & & & & & & & 1 & \times & \times & \times & & & \\
 & & & & & & & & & 1 & \times & \times & * & * & * \\
 & & & & & & & & & & 1 & \times & \times & & \\
 & & & & & & & & & & & 1 & \times & \times & \\
 & & & & & & & & & & & & 1 & \times & \times \\
 & & & & & & & & & & & & & 1 & \times \\
 & & & & & & & & & & & & & & 1
\end{bmatrix}.$$

Here, potentially nonzero off-diagonal entries in the banded part of $\mathbf{U}_{15}$ are marked by "$\times$", and potentially nonzero entries in the spiked part are marked by "$*$".

2.5. Block-Krylov subspaces in the case of exact deflation. It can be shown that performing only exact deflation in the symmetric band Lanczos process is equivalent to scanning the columns of the block-Krylov matrix (2.2) from left to right and deleting each column that is linearly dependent on earlier columns; see [1]. The result of such a deletion of columns in (2.2) is the *deflated* block-Krylov matrix

$$\mathbf{K}^{\mathrm{df}}(\mathbf{A}, \mathbf{R}) := [\,\mathbf{R}_1 \quad \mathbf{A}\,\mathbf{R}_2 \quad \mathbf{A}^2\,\mathbf{R}_3 \quad \cdots \quad \mathbf{A}^{j_{\max}-1}\,\mathbf{R}_{j_{\max}}\,].$$

By the structure of (2.2), a column $\mathbf{A}^{j-1}\,\mathbf{r}_i$ being linearly dependent on earlier columns implies that all $\mathbf{A}$-multiples, $\mathbf{A}^k\mathbf{r}_i$, $k \geq j$, of that column are also linearly dependent on earlier columns. As a result, for each $j = 1, 2, \ldots, j_{\max}$, $\mathbf{R}_j$ is a submatrix of $\mathbf{R}_{j-1}$, where, for $j = 1$, we set $\mathbf{R}_0 := \mathbf{R}$.

By construction, all columns of the matrix $\mathbf{K}^{\mathrm{df}}(\mathbf{A}, \mathbf{R})$ are linearly independent. The *n-th block-Krylov subspace*, $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$, induced by $\mathbf{A}$ and $\mathbf{R}$ is now defined as the n-dimensional subspace of $\mathbb{R}^N$ spanned by the first n columns of the matrix $\mathbf{K}^{\mathrm{df}}(\mathbf{A}, \mathbf{R})$. When only exact deflation is performed in the symmetric band Lanczos process, then the Lanczos vectors (2.3) span the n-th block-Krylov subspace, i.e.,

$$\operatorname{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\} = \mathcal{K}_n(\mathbf{A}, \mathbf{R}). \tag{2.22}$$

2.6. The standard symmetric band Lanczos process. In this subsection, we review the standard symmetric band Lanczos process.

The standard process only generates Lanczos vectors (2.3) and candidate vectors (2.4), but not the second set of vectors (2.5). The vectors (2.3) and (2.4) are constructed to satisfy the orthogonality conditions (2.7) and (2.8). The first m_1 Lanczos vectors again satisfy a relation of the form (2.12). The main recurrences used in the standard algorithm can be summarized in the form

$$\mathbf{A}\,\mathbf{V}_n^{\mathrm{s}} = \mathbf{V}_n^{\mathrm{s}}\,\widetilde{\mathbf{T}}_n^{\mathrm{s}} + [\,\underbrace{\mathbf{0}\;\;\mathbf{0}\;\;\cdots\;\;\mathbf{0}}_{n-m_{\mathrm{c}}}\;\;\underbrace{\hat{\mathbf{v}}_{n+1}^{\mathrm{s}}\;\;\hat{\mathbf{v}}_{n+2}^{\mathrm{s}}\;\;\cdots\;\;\hat{\mathbf{v}}_{n+m_{\mathrm{c}}}^{\mathrm{s}}}_{m_{\mathrm{c}}}\,] + \widehat{\mathbf{V}}_n^{\mathrm{s,\,df}}. \tag{2.23}$$

Here, the upper index "s" is used to differentiate the vectors and matrices generated by the standard process from those generated by the coupled process.

In (2.23), $\widetilde{\mathbf{T}}_n^{\mathrm{s}}$ is an $n\times n$ matrix whose entries are chosen such that the vectors (2.3) and (2.4) satisfy the orthogonality conditions (2.7) and (2.8). By multiplying (2.23) from the left by $(\mathbf{V}_n^{\mathrm{s}})^{\mathrm{T}}$, and by using (2.7) and (2.8), it follows that

$$\mathbf{T}_n^{\mathrm{s}} := (\mathbf{V}_n^{\mathrm{s}})^{\mathrm{T}}\;\mathbf{A}\mathbf{V}_n^{\mathrm{s}} = \widetilde{\mathbf{T}}_n^{\mathrm{s}} + (\mathbf{V}_n^{\mathrm{s}})^{\mathrm{T}}\;\widehat{\mathbf{V}}_n^{\mathrm{s,\,df}}. \tag{2.24}$$

It turns out that $\mathbf{T}_n^{\mathrm{s}}$ consists of a symmetric banded part with bandwidth decreasing from $2m+1$ to $2m_{\mathrm{c}}+1$ and a "spiked" part with potentially nonzero elements only in rows and columns with index $i \in \mathcal{I}$ and outside of the banded part. This means that the matrix $\mathbf{T}_n^{\mathrm{s}}$ has the same "fish-bone" sparsity structure as $\mathbf{U}_n^{\mathrm{T}} + \mathbf{U}_n$, where $\mathbf{U}_n$ is the triangular factor of $\mathbf{T}_n$ described in § 2.4.

For the example considered in § 2.4, the corresponding matrix $\mathbf{T}_n^{\mathrm{s}}$ produced by the standard algorithm has the following structure:

$$\mathbf{T}_{15}^{\mathrm{s}} = \left[\begin{array}{ccccccccccccccc}
\times & \times & \times & \times & \times & \times & & & & & & & & & \\
\times & \times & \times & \times & \times & \times & \times & & & & & & & & \\
\times & \times & \times & \times & \times & \times & \times & \ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\times & \times & \times & \times & \times & \times & \times & \times & & & & & & & \\
\times & \times & \times & \times & \times & \times & \times & \times & \times & & & & & & \\
\times & \times & \times & \times & \times & \times & \times & \times & \times & \times & & & & & \\
 & \times & \times & \times & \times & \times & \times & \times & \times & \times & \ast & \ast & \ast & \ast & \ast \\
 & & \ast & \times & \times & \times & \times & \times & \times & \times & \times & & & & \\
 & & \ast & & \times & \times & \times & \times & \times & \times & \times & \times & & & \\
 & & \ast & & & \times & \times & \times & \times & \times & \times & \times & \ast & \ast & \ast \\
 & & \ast & & & & \ast & \times & \times & \times & \times & \times & \times & & \\
 & & \ast & & & & \ast & & \times & \times & \times & \times & \times & \times & \\
 & & \ast & & & & \ast & & & \ast & \times & \times & \times & \times & \times \\
 & & \ast & & & & \ast & & & \ast & & \times & \times & \times & \times \\
 & & \ast & & & & \ast & & & \ast & & & \times & \times & \times
\end{array}\right].$$

Here, potentially nonzero entries in the banded part of $\mathbf{T}_{15}^{\mathrm{s}}$ are marked by "$\times$", and potentially nonzero entries in the spiked part are marked by "$\ast$".

For further details and properties of the standard symmetric band Lanczos algorithm, we refer the reader to [12]. We remark that in the earlier work [7, 23] on symmetric band or block Lanczos algorithms, deflated vectors are simply discarded. As a result, relations such as (2.24) hold true only approximately for these simpler algorithms.

2.7. Connection with the standard process. The standard band Lanczos process and the algorithm based on coupled recurrences are mathematically equivalent in the case that no deflation occurs or that only exact deflation is performed. Indeed, in this case, the first n Lanczos vectors (2.3) generated by both algorithms are identical and, in view of (2.22), span the n-th block-Krylov subspace $\mathcal{K}_n(\mathbf{A}, \mathbf{R})$. Hence, the associated projected Lanczos matrices $\mathbf{T}_n$ and $\mathbf{T}_n^{\mathrm{s}}$ given by (2.18) and (2.24) are identical. Moreover, in (2.14) and (2.23), we have $\widehat{\mathbf{V}}_n^{\mathrm{df}} = \widehat{\mathbf{V}}_n^{\mathrm{s,df}} = \mathbf{0}$. This implies that the spiked parts of both $\mathbf{T}_n$ and $\mathbf{U}_n$ are actually zero matrices. Thus both $\mathbf{T}_n$ and $\mathbf{U}_n = \mathbf{L}_n^{\mathrm{T}}$ are banded matrices, and (2.18) reduces to the $\mathrm{LDL}^{\mathrm{T}}$ factorization

$$\mathbf{T}_n = \mathbf{T}_n^{\mathrm{s}} = \mathbf{L}_n \, \boldsymbol{\Delta}_n \, \mathbf{L}_n^{\mathrm{T}}$$

of the Lanczos matrix associated with both variants of the symmetric band Lanczos process.

As soon as deflation of almost linearly dependent vectors is performed, the two processes are no longer mathematically equivalent in general. In this case, the spiked parts of both $\mathbf{T}_n$ and $\mathbf{U}_n$ are nonzero, and thus

$$\mathbf{T}_n^{\mathrm{s}} \neq \mathbf{T}_n = \mathbf{U}_n^{\mathrm{T}} \, \boldsymbol{\Delta}_n \, \mathbf{U}_n$$

in general. Indeed, $\mathbf{T}_n^{\mathrm{s}}$ has a "fish-bone" sparsity structure, while $\mathbf{T}_n$ is a full matrix in general, even though $\mathbf{U}_n$ is sparse. However, one can show that $\|\mathbf{T}_n - \mathbf{T}_n^{\mathrm{s}}\|$ is bounded by the tolerance used to check for deflation.

3. The algorithm and its properties. In this section, we present a detailed statement of the symmetric band Lanczos process with coupled recurrences, and establish some of its properties.

3.1. Statement of the algorithm. At each pass n through the main loop of the algorithm, one first checks if the candidate vector $\hat{\mathbf{v}}_n$ needs to be deflated. If yes, it is deleted, the indices of all the remaining candidate vectors are reduced by 1, and the deflation check is repeated. If no, the candidate vector $\hat{\mathbf{v}}_n$ is normalized to become the n-th Lanczos vector $\mathbf{v}_n$. In the remaining steps of pass n, the algorithm orthogonalizes the candidate vectors against $\mathbf{v}_n$, generates the potentially nonzero entries of the n-th column of $\mathbf{U}_n$ and, if $n \leq m_1$, of the n-th row of $\boldsymbol{\rho}_{m_1}$, and computes the vector $\mathbf{p}_n$, the scalar δ_n, and a new candidate vector $\hat{\mathbf{v}}_{n+m_{\mathrm{c}}}$.

A detailed statement of the algorithm is as follows.

ALGORITHM 3.1. (Symmetric band Lanczos process with coupled recurrences)

INPUT: *A matrix* $\mathbf{A} = \mathbf{A}^{\mathrm{T}} \in \mathbb{R}^{N \times N}$, *and a block* $\mathbf{R} = [\, \mathbf{r}_1 \quad \mathbf{r}_2 \quad \ldots \quad \mathbf{r}_m \,] \in \mathbb{R}^{N \times m}$ *of* m *starting vectors.*

OUTPUT: *Matrices* $\mathbf{U}_n$, $\boldsymbol{\Delta}_n$, *and (if* $n \geq m_1$*)* $\boldsymbol{\rho}_{m_1}$.

0) *Set* $\hat{\mathbf{v}}_i = \mathbf{r}_i$ *for* $i = 1, 2, \ldots, m$.
   Set $m_{\mathrm{c}} = m$.
   Set $\mathcal{I} = \emptyset$.

For $n = 1, 2, \ldots,$ *do*:

1) (If necessary, deflate.)
   Decide if $\hat{\mathbf{v}}_n$ *should be deflated.*
   If no, continue with Step 2).
   If yes, deflate $\hat{\mathbf{v}}_n$ *by doing the following*:
   1a) *If* $n - m_{\mathrm{c}} > 0$, *set* $\mathcal{I} = \mathcal{I} \cup \{n - m_{\mathrm{c}}\}$ *and* $\hat{\mathbf{v}}_{n-m_{\mathrm{c}}}^{\mathrm{df}} = \hat{\mathbf{v}}_n$.
   1b) *Set* $m_{\mathrm{c}} = m_{\mathrm{c}} - 1$. *If* $m_{\mathrm{c}} = 0$, *set* $n = n - 1$ *and stop.*
   1c) *For* $i = n, n+1, \ldots, n + m_{\mathrm{c}} - 1$, *set* $\hat{\mathbf{v}}_i = \hat{\mathbf{v}}_{i+1}$.
   1d) *Repeat all of Step* 1).
2) (Normalize $\hat{\mathbf{v}}_n$ to become the n-th Lanczos vector $\mathbf{v}_n$.)

Set $\mathbf{v}_n = \dfrac{\hat{\mathbf{v}}_n}{\|\hat{\mathbf{v}}_n\|}$.

If $n - m_c > 0$, *set* $u_{n-m_c,n} = \dfrac{\|\hat{\mathbf{v}}_n\|}{\delta_{n-m_c}}$.

If $n - m_c \leq 0$, *set* $\rho_{n,n-m_c+m} = \|\hat{\mathbf{v}}_n\|$.

If $n = m_c$, *set* $m_1 = m_c$.

3) (Orthogonalize the vectors $\hat{\mathbf{v}}_{n+j}$, $1 \leq j < m_c$, against $\mathbf{v}_n$.)
For $j = 1, 2, \ldots, m_c - 1$, *do*:
Set $\tau = \mathbf{v}_n^{\mathrm{T}}\,\hat{\mathbf{v}}_{n+j}$ *and* $\hat{\mathbf{v}}_{n+j} = \hat{\mathbf{v}}_{n+j} - \mathbf{v}_n\,\tau$.

If $n + j - m_c > 0$, *set* $u_{n+j-m_c,n} = \dfrac{\tau}{\delta_{n+j-m_c}}$.

If $n + j - m_c \leq 0$, *set* $\rho_{n,n+j-m_c+m} = \tau$.

4) (Update the spiked part of $\mathbf{U}_n$.)

For each $j \in \mathcal{I}$, *set* $u_{jn} = \dfrac{\mathbf{v}_n^{\mathrm{T}}\,\hat{\mathbf{v}}_j^{\mathrm{df}}}{\delta_j}$.

5) (Compute the vector $\mathbf{p}_n$.)

Set $j_0 = \max\{1, n - m_c\}$ *and*

$$\mathbf{p}_n = \mathbf{v}_n - \sum_{j\in\mathcal{I}} \mathbf{p}_j\,u_{jn} - \sum_{j=j_0}^{n-1} \mathbf{p}_j\,u_{jn}. \tag{3.1}$$

Set $u_{nn} = 1$.

6) (Advance the block Krylov subspace.)
Set $\hat{\mathbf{v}}_{n+m_c} = \mathbf{A}\,\mathbf{p}_n$.

7) *Set* $\delta_n = \mathbf{p}_n^{\mathrm{T}}\,\hat{\mathbf{v}}_{n+m_c}$.
If $\delta_n = 0$, *stop*: *look-ahead would be needed to continue.*

8) *Set* $\hat{\mathbf{v}}_{n+m_c} = \hat{\mathbf{v}}_{n+m_c} - \mathbf{v}_n\,\delta_n$.

9) *Test for convergence.*

Remark 3.1 In Algorithm 3.1, only the potentially nonzero entries of the matrices $\mathbf{U}_n$ and $\boldsymbol{\rho}_{m_1}$, and the diagonal entries of the diagonal matrix $\boldsymbol{\Delta}_n$ are computed. All other entries of these matrices are set to be zero.

Remark 3.2 The entries of $\mathbf{U}_n$ generated in Step 3) of Algorithm 3.1 are just the potentially nonzero off-diagonal entries of the banded part of the unit upper triangular matrix $\mathbf{U}_n$. In view of (2.21), the unit lower triangular matrix $\mathbf{L}_n$ can be obtained by simply transposing the banded part of $\mathbf{U}_n$.

Remark 3.3 A practical criterion for deflation in Step 1) of Algorithm 3.1 is as follows. The vector $\hat{\mathbf{v}}_n$ is deflated if, and only if,

$$\|\hat{\mathbf{v}}_n\| \leq \mathtt{dtol}_n.$$

Here,

$$\mathtt{dtol}_n := \mathtt{dtol} \times \begin{cases} \|\mathbf{r}_{n+m-m_c}\| & \text{if} \quad n \leq m_1, \\ \mathrm{nest}(\mathbf{A}) & \text{if} \quad n > m_1, \end{cases} \tag{3.2}$$

is the product of an absolute deflation tolerance `dtol` and a scaling factor that makes the deflation check independent of the actual scaling of the columns $\mathbf{r}_i$ of $\mathbf{R}$ and of the matrix $\mathbf{A}$. Ideally, we would like to set $\mathrm{nest}(\mathbf{A}) = \|\mathbf{A}\|$ in (3.2). However, if $\|\mathbf{A}\|$ is not available, then $\mathrm{nest}(\mathbf{A})$ is set to be an estimate of $\|\mathbf{A}\|$. For example, we can use the estimate

$$\mathrm{nest}(\mathbf{A}) := \max_{1 \leq i \leq m_1} \frac{\|\mathbf{A}\,\mathbf{p}_i\|}{\|\mathbf{p}_i\|} \leq \|\mathbf{A}\|\,,$$

which can be obtained from the vectors generated in Algorithm 3.1 at the expense of $2m_1$ additional vector-norm computations. Based on our numerical experiences with Algorithm 3.1, we recommend the absolute deflation tolerance $\mathtt{dtol} = \sqrt{\mathtt{eps}}$, where $\mathtt{eps}$ is the machine precision.

Remark 3.4 If $\mathbf{A} \geq 0$, then $\delta_n > 0$ for all n and so Algorithm 3.1 never stops in Step 7). If $\mathbf{A}$ is indefinite, then it cannot be excluded that Algorithm 3.1 terminates prematurely due to $\delta_n = 0$ in Step 7). In this case, look-ahead can be used to continue the process, but in order to keep the algorithm relatively simple, we opted to treat only the no-look-ahead case in this paper. However, we stress that Algorithm 3.1 can be extended to include look-ahead. Such an extension again generates an $\mathrm{LDL}^{\mathrm{T}}$ factorization (2.18), where $\mathbf{\Delta}_n$ is a block-diagonal matrix in general. Furthermore, the nonsingularity (2.11) of $\mathbf{\Delta}_n$ is only guaranteed for the values of n that correspond to the end of a look-ahead step.

Remark 3.5 Algorithm 3.1 requires storage of the vector $\mathbf{v}_n$, the m_{c} candidate vectors $\hat{\mathbf{v}}_{n+1}, \hat{\mathbf{v}}_{n+2}, \ldots, \hat{\mathbf{v}}_{n+m_{\mathrm{c}}}$, the m_{c} vectors $\mathbf{p}_{n-m_{\mathrm{c}}+1}, \mathbf{p}_{n-m_{\mathrm{c}}+2}, \ldots, \mathbf{p}_n$, and all the vectors $\mathbf{p}_i$ and $\hat{\mathbf{v}}_i^{\mathrm{df}}$ with $i \in \mathcal{I}$. Since the set $\mathcal{I}$ contains at most $m - m_{\mathrm{c}}$ elements, the total number of vectors to be stored is at most

$$1 + m_{\mathrm{c}} + m_{\mathrm{c}} + 2(m - m_{\mathrm{c}}) = 2m + 1,$$

which is identical to the storage requirement of the standard symmetric band Lanczos process.

3.2. Properties. We now show that the quantities generated by Algorithm 3.1 indeed satisfy the governing equations stated in § 2.2.

THEOREM 3.2. *The vectors and matrices generated by n iterations of Algorithm* 3.1 *satisfy the recurrence relations* (2.14), (2.15)*, and* (*if* $n \geq m_1$) (2.12)*.*

Proof. Using the matrices introduced in (2.9), (2.10), (2.13), and (2.16), it is straightforward to verify that all the recurrences employed in the first n iterations of Algorithm 3.1 can indeed be summarized as the matrix relations (2.12), (2.14), and (2.15). □

THEOREM 3.3. *The vectors and matrices generated by n iterations of Algorithm* 3.1 (*run in exact arithmetic*) *satisfy the orthogonality conditions* (2.7)–(2.9).

Proof. From Steps 2), 6), and 7) of Algorithm 3.1, it follows that

$$\mathbf{v}_i^{\mathrm{T}}\,\mathbf{v}_i = 1 \quad \text{and} \quad \mathbf{p}_i^{\mathrm{T}}\,\mathbf{A}\,\mathbf{p}_i = \delta_i \quad \text{for all} \quad 1 \leq i \leq n. \tag{3.3}$$

Next, we use induction on $n\,(\geq 0)$ to show that

$$\mathbf{v}_i^{\mathrm{T}}\,\mathbf{v}_n = 0 \quad \text{for all} \quad 1 \leq i < n, \tag{3.4}$$

$$\mathbf{p}_i^{\mathrm{T}}\,\mathbf{A}\,\mathbf{p}_n = 0 \quad \text{for all} \quad 1 \leq i < n, \tag{3.5}$$

$$\mathbf{v}_i^{\mathrm{T}}\,\hat{\mathbf{v}}_{n+j} = 0 \quad \text{for all} \quad 1 \leq i \leq n,\ 1 \leq j \leq m_{\mathrm{c}}(n). \tag{3.6}$$

Note that the relations (3.4)–(3.6), together with (3.3), are equivalent to the orthogonality conditions (2.7)–(2.9).

The assertions (3.4)–(3.6) are trivially satisfied for $n = 0$, since the sets of indices i in (3.4)–(3.6) are all empty in this case. Now let $n \geq 1$, and as induction hypothesis, assume that for all $0 \leq n' < n$, we have

$$\mathbf{v}_i^{\mathrm{T}}\, \mathbf{v}_{n'} = 0 \quad \text{for all} \quad 1 \leq i < n',$$

$$\mathbf{p}_i^{\mathrm{T}}\, \mathbf{A}\, \mathbf{p}_{n'} = 0 \quad \text{for all} \quad 1 \leq i < n', \tag{3.7}$$

$$\mathbf{v}_i^{\mathrm{T}}\, \hat{\mathbf{v}}_{n'+j} = 0 \quad \text{for all} \quad 1 \leq i \leq n',\ 1 \leq j \leq m_{\mathrm{c}}(n'). \tag{3.8}$$

We need to show that the relations (3.4)–(3.6) are satisfied.

Since $\mathbf{v}_n = \hat{\mathbf{v}}_n / \|\hat{\mathbf{v}}_n\|$, the orthogonality condition (3.4) for $\mathbf{v}_n$ is an immediate consequence of (3.8) (with $n' = n-1$ and $j = 1$).

We now turn to (3.5). Note that at the end of the i-th iteration of Algorithm 3.1, we have

$$\hat{\mathbf{v}}_{i+m_{\mathrm{c}}(i)}^{(i)} := \hat{\mathbf{v}}_{i+m_{\mathrm{c}}(i)} = \mathbf{A}\, \mathbf{p}_i - \mathbf{v}_i\, \delta_i. \tag{3.9}$$

Here, the upper index "(i)" indicates that $\hat{\mathbf{v}}_{i+m_{\mathrm{c}}(i)}^{(i)}$ is the $(i + m_{\mathrm{c}}(i))$-th candidate vector at the end of the i-th iteration. Note that, by (3.9) and (3.4), we have

$$\mathbf{v}_n^{\mathrm{T}}\, \mathbf{A}\, \mathbf{p}_i = \mathbf{v}_n^{\mathrm{T}} \left(\hat{\mathbf{v}}_{i+m_{\mathrm{c}}(i)}^{(i)} + \mathbf{v}_i\, \delta_i \right) = \mathbf{v}_n^{\mathrm{T}}\, \hat{\mathbf{v}}_{i+m_{\mathrm{c}}(i)}^{(i)} \quad \text{for all} \quad 1 \leq i < n. \tag{3.10}$$

Let $1 \leq i < n$ be arbitrary, but fixed, and consider the transformations that are performed on the candidate vector $\hat{\mathbf{v}}_{i+m_{\mathrm{c}}(i)}^{(i)}$ during iterations $i+1, i+2, \ldots, n$ of Algorithm 3.1. Note that there are only two types of transformations: shift of the (lower) index of the candidate vector by -1, and orthogonalization against previous Lanczos vectors. For the transformed vector resulting from $\hat{\mathbf{v}}_{i+m_{\mathrm{c}}(i)}^{(i)}$, there are the following three cases:

(I) The transformed vector is deflated during iteration $\tilde{n}$ for some $i < \tilde{n} \leq n$;

(II) The transformed vector is one of the candidate vectors $\hat{\mathbf{v}}_n, \hat{\mathbf{v}}_{n+1}, \ldots, \hat{\mathbf{v}}_{n-m_{\mathrm{c}}-1}$ before Steps 2) and 3) are performed within the n-th iteration of Algorithm 3.1;

(III) The transformed vector is normalized to become the k-th Lanczos vector $\mathbf{v}_k$ for some $k < n$.

Case (I) occurs if, and only if, $i \in \mathcal{I}$. Indeed, let $\tilde{m}_{\mathrm{c}}$ be the value of m_{c} when the transformed vector $\hat{\mathbf{v}}_{\tilde{n}}$ is checked for deflation during iteration $\tilde{n}$. Since $\hat{\mathbf{v}}_{\tilde{n}}$ resulted from $\hat{\mathbf{v}}_{i+m_{\mathrm{c}}(i)}^{(i)}$, we have

$$\tilde{n} = i + m_{\mathrm{c}}(i) - (m_{\mathrm{c}}(i) - \tilde{m}_{\mathrm{c}}) = i + \tilde{m}_{\mathrm{c}},$$

and thus $i = \tilde{n} - \tilde{m}_{\mathrm{c}} > 0$. By Step 1a) of Algorithm 3.1, deflation of the vector $\hat{\mathbf{v}}_{\tilde{n}}$ is equivalent to $i = \tilde{n} - \tilde{m}_{\mathrm{c}} \in \mathcal{I}$. Moreover, note that $\hat{\mathbf{v}}_i^{\mathrm{df}} = \hat{\mathbf{v}}_{\tilde{n}}$ and that $\hat{\mathbf{v}}_{\tilde{n}}$ was obtained by orthogonalizing $\hat{\mathbf{v}}_{i+m_{\mathrm{c}}(i)}^{(i)}$ against Lanczos vectors $\mathbf{v}_k$ with $k < \tilde{n} \leq n$. In view of (3.4) and Step 4) of Algorithm 3.1, it follows that

$$\mathbf{v}_n^{\mathrm{T}}\, \hat{\mathbf{v}}_{i+m_{\mathrm{c}}(i)}^{(i)} = \mathbf{v}_n^{\mathrm{T}}\, \hat{\mathbf{v}}_i^{\mathrm{df}} = \delta_i\, u_{in} \quad \text{if} \quad i \in \mathcal{I}. \tag{3.11}$$

Case (II) occurs if, and only if, $i \geq j_0 := \max\{1,\, n - m_{\mathrm{c}}\}$. Indeed, the transformed vector has index $i + m_{\mathrm{c}}$ and is thus one of the candidate vectors $\hat{\mathbf{v}}_n, \ldots, \hat{\mathbf{v}}_{n-m_{\mathrm{c}}-1}$ if, and only if,

$$n \leq i + m_{\mathrm{c}} \leq n - m_{\mathrm{c}} - 1. \tag{3.12}$$

Since $1 \leq i < n$, the condition (3.12) is equivalent to $i \geq j_0$. Note that, in view of (3.4) and Steps 2) and 3) of Algorithm 3.1, we have

$$\mathbf{v}_n^{\mathrm{T}}\,\hat{\mathbf{v}}_{i+m_c(i)}^{(i)} = \mathbf{v}_n^{\mathrm{T}}\,\hat{\mathbf{v}}_{i+m_c} = \delta_i\,u_{in} \quad \text{if} \quad i \geq j_0. \tag{3.13}$$

Case (III) complements cases (I) and (II), and thus occurs if, and only if, $i < j_0$ and $i \notin \mathcal{I}$. In this case, in view of (3.4), we have

$$\mathbf{v}_n^{\mathrm{T}}\,\hat{\mathbf{v}}_{i+m_c(i)}^{(i)} = \mathbf{v}_n^{\mathrm{T}}\,\hat{\mathbf{v}}_k = \left(\mathbf{v}_n^{\mathrm{T}}\,\mathbf{v}_k\right)\,\|\hat{\mathbf{v}}_k\| = 0. \tag{3.14}$$

Here, $k < n$ is the index of the Lanczos vector $\mathbf{v}_k$ that resulted from $\hat{\mathbf{v}}_{i+m_c(i)}^{(i)}$.

By combining (3.10) with (3.11), (3.13), and (3.14), we get

$$\mathbf{v}_n^{\mathrm{T}}\,\mathbf{A}\,\mathbf{p}_i = \mathbf{v}_n^{\mathrm{T}}\,\hat{\mathbf{v}}_{i+m_c(i)}^{(i)} = \begin{cases} \delta_i\,u_{in} & \text{if} \quad i \in \mathcal{I} \quad \text{or} \quad j_0 \leq i < n, \\ 0 & \text{if} \quad i < j_0 \quad \text{and} \quad i \notin \mathcal{I}. \end{cases} \tag{3.15}$$

On the other hand, by transposing the relation (3.1), multiplying it from the right by $\mathbf{A}\,\mathbf{p}_i$, and using (3.3) and (3.7), it follows that

$$\mathbf{p}_n^{\mathrm{T}}\,\mathbf{A}\,\mathbf{p}_i = \mathbf{v}_n^{\mathrm{T}}\,\mathbf{A}\,\mathbf{p}_i - \begin{cases} \delta_i\,u_{in} & \text{if} \quad i \in \mathcal{I} \quad \text{or} \quad j_0 \leq i < n, \\ 0 & \text{if} \quad i < j_0 \quad \text{and} \quad i \notin \mathcal{I}. \end{cases} \tag{3.16}$$

Inserting (3.15) into (3.16) gives (3.5).

Finally, we show (3.6). Note that $m_c(i)$ is a non-increasing function of i, and in particular, $m_c' \geq m_c(n) =: m_c$. By (3.8) (for $n' = n-1$ and $2 \leq j \leq m_c$), the candidate vectors $\hat{\mathbf{v}}_{n+1}, \ldots, \hat{\mathbf{v}}_{n+m_c-1}$ are orthogonal to $\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}$ before Step 3) is performed within the n-th iteration of Algorithm 3.1. The update

$$\hat{\mathbf{v}}_{n+j} = \hat{\mathbf{v}}_{n+j} - \mathbf{v}_n\,\left(\mathbf{v}_n^{\mathrm{T}}\,\hat{\mathbf{v}}_{n+j}\right), \quad 1 \leq j < m_c, \tag{3.17}$$

in Step 3) then makes these candidate vectors also orthogonal to $\mathbf{v}_n$. Moreover, in view of (3.4), the update (3.17) does not destroy the orthogonality to $\mathbf{v}_1, \ldots, \mathbf{v}_{n-1}$, and thus (3.6) is satisfied for all $1 \leq j < m_c$. In order to prove (3.6) for $j = m_c$, we first note that, by (3.9) (for $i = n$),

$$\hat{\mathbf{v}}_{n+m_c} = \mathbf{A}\,\mathbf{p}_n - \mathbf{v}_n\,\delta_n.$$

Multiplying this relation from the left by $\mathbf{v}_i^{\mathrm{T}}$ and using (2.6), (3.4), and (3.5), it follows that

$$\mathbf{v}_i^{\mathrm{T}}\,\hat{\mathbf{v}}_{n+m_c} = \mathbf{v}_i^{\mathrm{T}}\,\mathbf{A}\,\mathbf{p}_n - \left(\mathbf{v}_i^{\mathrm{T}}\,\mathbf{v}_n\right)\,\delta_n = 0 \quad \text{for all} \quad 1 \leq i < n.$$

We have thus established all three relations (3.4)–(3.6), and the proof of Theorem 3.3 is complete. □

4. Application to passive reduced-order modeling. In this section, we discuss the application of the band Lanczos process with coupled recurrences to construct passive reduced-order models.

4.1. The problem. Consider *symmetric* m-input m-output time-invariant linear dynamical systems of the form

$$\begin{aligned} \mathbf{C}\,\frac{d}{dt}\mathbf{x}(t) &= -\mathbf{G}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{B}^{\mathrm{T}}\,\mathbf{x}(t). \end{aligned} \tag{4.1}$$

Here, $\mathbf{C} = \mathbf{C}^{\mathrm{T}}$, $\mathbf{G} = \mathbf{G}^{\mathrm{T}} \in \mathbb{R}^{N\times N}$, and $\mathbf{B} \in \mathbb{R}^{N\times m}$ are given matrices. Moreover, we assume that the matrix pencil $\mathbf{G} + s\,\mathbf{C}$ is *regular*, i.e., $\det(\mathbf{G} + s\,\mathbf{C}) = 0$ for only finitely many values of $s \in \mathbb{C}$. In (4.1), the components of the given vector-valued function $\mathbf{u}\colon [0,\infty) \mapsto \mathbb{R}^m$ are the inputs, $\mathbf{y}\colon [0,\infty) \mapsto \mathbb{R}^m$ is the unknown function of outputs, the components of the unknown vector-valued function $\mathbf{x}\colon [0,\infty) \mapsto \mathbb{R}^N$ are the state variables, and N is the state-space dimension. Systems of the form (4.1) can be used to model so-called RCL electrical networks consisting of resistors, capacitors, and inductors; see, e.g., [11] and the references given there. An important special case is RC networks consisting of only resistors and capacitors; in this case, the matrices $\mathbf{C}$ and $\mathbf{G}$ in (4.1) are sparse and positive semi-definite.

A *reduced-order model* of (4.1) is a system of the same form as (4.1), but of smaller state-space dimension $n < N$. A reduced-order model of dimension n is thus of the form

$$\begin{aligned} \mathbf{C}_n\,\frac{d}{dt}\mathbf{z}(t) &= -\mathbf{G}_n\,\mathbf{z}(t) + \mathbf{B}_n\,\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{B}_n^{\mathrm{T}}\,\mathbf{x}(t), \end{aligned} \tag{4.2}$$

where $\mathbf{C}_n = \mathbf{C}_n^{\mathrm{T}}$, $\mathbf{G}_n = \mathbf{G}_n^{\mathrm{T}} \in \mathbb{R}^{n\times n}$, and $\mathbf{B}_n \in \mathbb{R}^{n\times m}$. These three matrices should be chosen such that the input-output mapping $\mathbf{u}(t) \mapsto \mathbf{y}(t)$ of (4.2) somehow approximates the input-output mapping of the original system (4.1).

The input-output behavior of the original system (4.1), respectively the reduced-order model (4.2), can be described by the associated *transfer function*

$$\mathbf{Z}(s) = \mathbf{B}^{\mathrm{T}}(\mathbf{G} + s\,\mathbf{C})^{-1}\mathbf{B}, \quad \text{respectively} \quad \mathbf{Z}_n(s) = \mathbf{B}_n^{\mathrm{T}}(\mathbf{G}_n + s\,\mathbf{C}_n)^{-1}\mathbf{B}_n, \tag{4.3}$$

where $s \in \mathbb{C}$ is a complex variable. Note that both $\mathbf{Z}$ and $\mathbf{Z}_n$ are $m \times m$-matrix-valued rational functions.

Now, let $s_0 \in \mathbb{C}$ be any expansion point such that the matrix $\mathbf{G} + s_0\,\mathbf{C}$ is nonsingular. The reduced-order model (4.2) of (4.1) is called a *matrix-Padé model* (associated with the expansion point s_0) if the matrices $\mathbf{C}_n$, $\mathbf{G}_n$, and $\mathbf{B}_n$ in (4.2) are such that the transfer functions (4.3) satisfy

$$\mathbf{Z}_n(s) = \mathbf{Z}(s) + \mathcal{O}\big((s - s_0)^{q(n)}\big), \tag{4.4}$$

where the integer $q(n)$ is as large as possible. The condition (4.4) means that the Taylor expansions of $\mathbf{Z}_n$ and $\mathbf{Z}$ about s_0 agree in as many leading ($m \times m$-matrix-valued) coefficients as possible, and thus $\mathbf{Z}_n$ is a matrix-Padé approximant of $\mathbf{Z}$ about the expansion point s_0.

4.2. Reduced-order models via the symmetric band Lanczos process. In the sequel, we assume that the matrices $\mathbf{C}$ and $\mathbf{G}$ in (4.1) are sparse and positive semi-definite. Moreover, let $s_0 \geq 0$ be any real expansion point such that $\mathbf{G} + s_0\,\mathbf{C} > 0$, and let

$$\mathbf{G} + s_0\,\mathbf{C} = \mathbf{M}\,\mathbf{M}^{\mathrm{T}} \tag{4.5}$$

be a sparse Cholesky factorization of $\mathbf{G} + s_0\,\mathbf{C}$. Note that, in general, $\mathbf{M}$ is a product of a permutation matrix, which records any pivoting for sparsity, and a sparse lower triangular matrix. Using (4.5), the transfer function $\mathbf{Z}$ in (4.3) can be recast as follows:

$$
\begin{aligned}
&\mathbf{Z}(s) = \mathbf{R}^{\mathrm{T}}\left(\mathbf{I} + (s - s_0)\,\mathbf{A}\right)^{-1}\mathbf{R},\\
&\text{where}\quad \mathbf{A} := \mathbf{M}^{-1}\,\mathbf{C}\,\mathbf{M}^{-\mathrm{T}} \geq 0 \quad\text{and}\quad \mathbf{R} := \mathbf{M}^{-1}\,\mathbf{B}.
\end{aligned}
\tag{4.6}
$$

In [14, 15], it was proposed to obtain reduced-order models via the symmetric band Lanczos process applied to the matrix $\mathbf{A}$ and the block of starting vectors $\mathbf{R}$ given in (4.6). More precisely, after $n\,(\geq m_1)$ iterations of the algorithm, a reduced-order transfer function of dimension n is defined as follows:

$$
\mathbf{Z}_n(s) = \boldsymbol{\rho}_n^{\mathrm{T}}\left(\mathbf{I}_n + (s - s_0)\,\mathbf{T}_n)\right)^{-1}\boldsymbol{\rho}_n, \quad \text{where} \quad \boldsymbol{\rho}_n = \begin{bmatrix} \boldsymbol{\rho}_{m_1} \\ \mathbf{0}_{n-m_1\times m} \end{bmatrix}. \tag{4.7}
$$

Here, $\mathbf{T}_n$ is the $n\times n$ projected Lanczos matrix and $\boldsymbol{\rho}_n$ is the $m_1\times m$ matrix containing the coefficients used to orthogonalize the starting block $\mathbf{R}$. In [14, 15], the standard symmetric band Lanczos process was used to generate $\mathbf{T}_n$ and $\boldsymbol{\rho}_n$, and the resulting algorithm was termed SyMPVL.

Here, we propose to employ the symmetric band Lanczos process based on coupled recurrences, instead of the standard algorithm, and we call the resulting computational procedure SyMPVL2.

ALGORITHM 4.1. (SyMPVL2)

INPUT: *Matrices* $\mathbf{C},\,\mathbf{G} \in \mathbb{R}^{N\times N}$ *such that* $\mathbf{C},\,\mathbf{G} \geq \mathbf{0}$ *and* $\mathbf{G} + s\,\mathbf{C}$ *is a regular pencil. A matrix* $\mathbf{B} \in \mathbb{R}^{N\times m}$.

OUTPUT: *Transfer function* $\mathbf{Z}_n$ *of a reduced-order model of dimension* n.

1) *Select expansion point* $s_0 \geq 0$ *with* $\mathbf{G} + s_0\,\mathbf{C} > \mathbf{0}$.
2) *Compute Cholesky decomposition* $\mathbf{G} + s_0\,\mathbf{C} = \mathbf{M}\,\mathbf{M}^{\mathrm{T}}$.
3) *Run* $n\,(\geq m_1)$ *iterations of Algorithm* 3.1 (*applied to* $\mathbf{A} := \mathbf{M}^{-1}\,\mathbf{C}\,\mathbf{M}^{-\mathrm{T}}$ *and* $\mathbf{R} := \mathbf{M}^{-1}\,\mathbf{B}$) *to generate matrices* $\mathbf{U}_n$, $\boldsymbol{\Delta}_n$, *and* $\boldsymbol{\rho}_{m_1}$.
4) *Set*

$$
\mathbf{Z}_n(s) = \boldsymbol{\rho}_n^{\mathrm{T}}\left(\mathbf{I}_n + (s - s_0)\,\mathbf{U}_n^{\mathrm{T}}\,\boldsymbol{\Delta}_n\,\mathbf{U}_n\right)^{-1}\boldsymbol{\rho}_n, \quad \textit{where} \quad \boldsymbol{\rho}_n = \begin{bmatrix} \boldsymbol{\rho}_{m_1} \\ \mathbf{0}_{n-m_1\times m} \end{bmatrix}.
$$

Remark 4.6 As we mentioned in § 2.7, the standard band Lanczos process and the algorithm based on coupled recurrences are mathematically equivalent in the case that no deflation occurs or that only exact deflation is performed. Consequently, in this case, SyMPVL and SyMPVL2 are mathematically equivalent. Furthermore, in [10, 15], it was shown that the reduced-order model defined by the transfer function (4.7) is a matrix-Padé model.

4.3. Passivity. Linear dynamical systems of the form (4.1) with $\mathbf{C},\,\mathbf{G} \geq \mathbf{0}$ are *passive*. Roughly speaking, a system (4.1) is passive if it does not generate energy. In terms of the transfer function (4.3), $\mathbf{Z}$, passivity of (4.1) is equivalent to the *positive realness* of $\mathbf{Z}$; see, e.g., [2, 24]. A precise definition of positive realness is as follows.

DEFINITION 4.2. (Positive realness)

An $m \times m$*-matrix-valued rational function* $\mathbf{Z}$ *is called positive real if*

(i) $\mathbf{Z}$ *has no poles in* $\mathbb{C}_+ := \{\, s \in \mathbb{C} \mid \mathrm{Re}(s) > 0 \,\}$;
(ii) $\mathbf{Z}(\bar{s}) = \overline{\mathbf{Z}(s)}$ *for all* $s \in \mathbb{C}$;
(iii) $\mathrm{Re}\left(\mathbf{x}^{\mathrm{H}}\,\mathbf{Z}(s)\,\mathbf{x}\right) \geq 0$ *for all* $\mathbf{x} \in \mathbb{C}^m$ *and all* $s \in \mathbb{C}_+$.

Passivity is a stronger condition than stability of a linear dynamical system. When reduced-order models of passive linear subsystems are used within a simulation of a (not necessarily linear) system, then passivity of the reduced-order models is needed to guarantee stability of the overall simulation; see the references given in [13].

By [13, Theorem 13], a reduced-order model (4.2) with transfer function $\mathbf{Z}_n$ given by (4.3) is passive if $\mathbf{C}_n \geq \mathbf{0}$ and $\mathbf{G}_n \geq \mathbf{0}$. By applying this result to the reduced-order transfer function (4.7), it follows that passivity is guaranteed if

$$\mathbf{T}_n \geq \mathbf{0} \quad \text{and} \quad \mathbf{I}_n - s_0\, \mathbf{T}_n \geq \mathbf{0}. \tag{4.8}$$

In exact arithmetic, the two conditions (4.8) are always fulfilled. Indeed, the first relation in (4.8) follows from $\mathbf{T}_n = \mathbf{V}_n^{\mathrm{T}}\, \mathbf{A}\, \mathbf{V}_n$ and $\mathbf{A} \geq \mathbf{0}$, and the second relation in (4.8) follows from

$$\mathbf{M}\, (\mathbf{I}_N - s_0\, \mathbf{A})\, \mathbf{M}^{\mathrm{T}} = \mathbf{G} \geq \mathbf{0}$$

and

$$\mathbf{I}_n - s_0\, \mathbf{T}_n = \mathbf{V}_n^{\mathrm{T}}\, (\mathbf{I}_N - s_0\, \mathbf{A})\, \mathbf{V}_n \geq \mathbf{0}.$$

Unfortunately, in finite-precision arithmetic, round-off may cause the matrix $\mathbf{T}_n$ generated by the standard band Lanczos process to be indefinite. The negative eigenvalues of $\mathbf{T}_n$ translate into positive poles of $\mathbf{Z}_n$, causing the reduced-order model given by (4.7) to be non-passive; see the examples given in § 4.5 below. This problem can easily be remedied by employing the SyMPVL2 Algorithm 4.1. Instead of $\mathbf{T}_n$, it generates an $\mathrm{LDL}^{\mathrm{T}}$ factorization, $\mathbf{U}_n^{\mathrm{T}}\, \boldsymbol{\Delta}_n\, \mathbf{U}_n$, of $\mathbf{T}_n$. In view of (2.9) and (2.19), we have $\boldsymbol{\Delta}_n \geq \mathbf{0}$ and thus the first relation in (4.8) is satisfied.

4.4. An error bound. An important practical issue for the SyMPVL2 Algorithm 4.1 is to determine if the reduced-order model $\mathbf{Z}_n$ is a sufficiently good approximation to $\mathbf{Z}$ for some prescribed range of s. For the special case of single-input single-output linear dynamical systems, a computable bound for the error $\mathbf{Z} - \mathbf{Z}_n$ was proposed in [5]. It is straightforward to apply the technique from [5] to the reduced-order transfer function $\mathbf{Z}_n$ generated by Algorithm 4.1. The resulting expression for the error is as follows:

$$\mathbf{Z}(s) - \mathbf{Z}_n(s) = \sigma^2\, (\mathbf{F}_m(\sigma))^{\mathrm{T}} \left(\widehat{\mathbf{V}}_{m_c}^{\mathrm{T}}\, (\mathbf{I} + \sigma\, \mathbf{A})^{-1}\, \widehat{\mathbf{V}}_{m_c}\right) \mathbf{F}_m(\sigma) + \mathbf{N}_n(\sigma). \tag{4.9}$$

Here, we set $\sigma := s - s_0$ and

$$\begin{aligned} \mathbf{F}_n(\sigma) &:= [\, \mathbf{0}_{m_c \times n - m_c} \quad \mathbf{I}_{m_c} \,] \left(\mathbf{I}_n + \sigma\, \mathbf{U}_n^{\mathrm{T}}\, \boldsymbol{\Delta}_n\, \mathbf{U}_n\right)^{-1} \begin{bmatrix} \boldsymbol{\rho}_m^{\mathrm{T}} \\ \mathbf{0}_{n-m \times m} \end{bmatrix}, \\ \widehat{\mathbf{V}}_{m_c} &:= [\, \hat{\mathbf{v}}_{n+1} \quad \hat{\mathbf{v}}_{n+2} \quad \cdots \quad \hat{\mathbf{v}}_{n+m_c} \,]. \end{aligned}$$

Note that $\mathbf{F}_n$ is an $m_c \times m$-matrix-valued function. Furthermore, in (4.9), $\mathbf{N}_n$ is an $m \times m$-matrix-valued function that can be shown to be of the order of the deflation tolerance (3.2), $\mathtt{dtol}_n$.

Provided that $|\sigma| < 1/\, \|\mathbf{A}\|$, the bracketed term in (4.9) can be bounded as follows:

$$\left\| \widehat{\mathbf{V}}_{m_c}^{\mathrm{T}}\, (\mathbf{I} + \sigma\, \mathbf{A})^{-1}\, \widehat{\mathbf{V}}_{m_c} \right\| \;\leq\; \frac{\|\widehat{\mathbf{V}}_{m_c}\|^2}{1 - |\sigma|\; \|\mathbf{A}\|}. \tag{4.10}$$

By taking norms in (4.9) and using (4.10), it follows that

$$\|\mathbf{Z}(s) - \mathbf{Z}_n(s)\| \;\le\; |\sigma|^2 \, \|\mathbf{F}_m(\sigma)\|^2 \, \frac{\|\widehat{\mathbf{V}}_{m_c}\|^2}{1 - |\sigma| \; \|\mathbf{A}\|} + \mathcal{O}\left(\mathtt{dtol}_n\right) \tag{4.11}$$

for all $|\sigma| < 1/\,\|\mathbf{A}\|$. In practice, we drop the last term, $\mathcal{O}\left(\mathtt{dtol}_n\right)$, and only evaluate the remaining terms of the error bound (4.11). The $m_{\mathrm{c}} \times m$-matrix-valued function $\mathbf{F}_m(\sigma)$ can be computed explicitly, and sufficiently good approximations of the norms $\|\mathbf{F}_m(\sigma)\|$ and $\|\widehat{\mathbf{V}}_{m_c}\|$ can easily be obtained, for instance, using the Matlab function `normest`. The norm $\|\mathbf{A}\|$ can be estimated by using the largest eigenvalue of the projected Lanczos matrix $\mathbf{U}_n^{\mathrm{T}}\, \mathbf{\Delta}_n \, \mathbf{U}_n$; see, e.g., [17, Section 9.1.4].

We have performed extensive numerical tests with SyMPVL2 using the convergence check based on evaluating the error bound (4.11). We found that, typically, the computational costs for this convergence check is about 5% to 8% of the cost of the underlying symmetric band Lanczos process.

4.5. Numerical examples. In this subsection, we present results of three numerical examples that demonstrate the superiority, both in terms of robustness and accuracy, of the SyMPVL2 Algorithm 4.1 based on coupled recurrences over the original SyMPVL algorithm [14, 15] based on the standard symmetric band Lanczos process.

All three examples are passive linear dynamical systems (4.1) describing RC networks arising in VLSI circuit simulation. The frequency range of interest is always $s = 2\pi\,\mathrm{i}\,\omega$, where $1 \le \omega \le 10^9$. The goal is to compute a reduced-order transfer function $\mathbf{Z}_n$ that approximates the transfer function $\mathbf{Z}$ of (4.1) well in this frequency range. In all three examples, $\mathbf{C} \ge \mathbf{0}$, $\mathbf{G} > 0$, and the expansion point $s_0 = 0$ is used. All experiments were performed in Matlab. The symmetric band Lanczos process was run with deflation tolerance (3.2), where $\mathtt{dtol} = \sqrt{\mathtt{eps}}$ and `eps` is the machine precision.

Example 4.1. In this example, $\mathbf{C}$ and $\mathbf{G}$ are matrices of order $N = 1346$, and $m = 10$. Both SyMPVL and SyMPVL2 required $n = 60$ iterations. However, the reduced-order model generated via SyMPVL is not passive, and not even stable. The reason is that the computed matrix $\mathbf{T}_{60}$ is indefinite, causing some positive poles of the transfer function $\mathbf{Z}_{60}$. On the other hand, SyMPVL2 generates a diagonal matrix $\mathbf{\Delta}_{60}$ with only positive diagonal entries, and the matrix $\mathbf{U}_{60}^{\mathrm{T}}\, \mathbf{\Delta}_{60}\, \mathbf{U}_{60}$ is positive definite. In particular, the transfer function $\mathbf{Z}_{60}$ does not have any positive poles. In Fig. 4.1, we show the dominant poles of the reduced-order models obtained from SyMPVL and SyMPVL2, as well as the dominant poles of the transfer function $\mathbf{Z}$ of the original linear dynamical system. Note that SyMPVL not only generated two unstable poles, but also one of the negative poles close to zero is wrong. In Fig. 4.2, we plot the SyMPVL and SyMPVL2 errors $\|\mathbf{Z}(s) - \mathbf{Z}_{60}(s)\|$ for all $s = 2\pi\,\mathrm{i}\,\omega$, $1 \le \omega \le 10^9$. Clearly, the SyMPVL2 model is also more accurate than the SyMPVL model.

Example 4.2. In this example, $\mathbf{C}$ and $\mathbf{G}$ are matrices of order $N = 13875$, and $m = 150$. A reduced-order model of dimension $n = 300$ is needed to achieve convergence in the frequency range of interest. Again, SyMPVL produced an indefinite matrix $\mathbf{T}_{300}$, resulting in some positive poles of $\mathbf{Z}_{300}$, while the reduced-order model generated via SyMPVL2 has no positive poles and is indeed passive. In Fig. 4.3, we show the dominant poles of the reduced-order models obtained from SyMPVL and SyMPVL2.

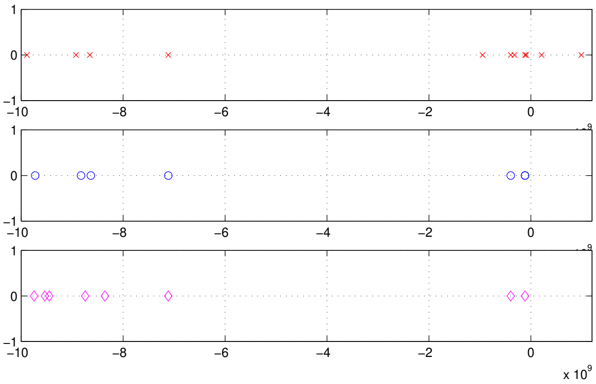

FIG. 4.1. *Dominant poles of SyMPVL reduced-order model (top), SyMPVL2 reduced-order model (middle), and the exact dominant poles (bottom) for Example 4.1.*

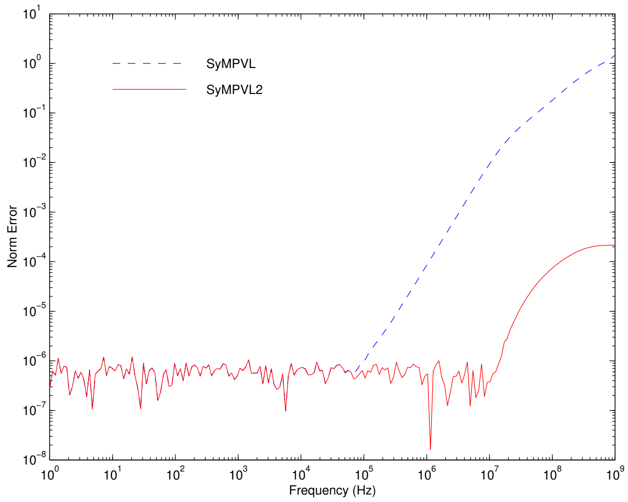

FIG. 4.2. *Error* $\|\mathbf{Z}(s) - \mathbf{Z}_n(s)\|$ *of SyMPVL and SyMPVL2 models for Example 4.1.*

Example 4.3. This example illustrates the behavior of the error bound described in § 4.4. We use the same matrices $\mathbf{C}$ and $\mathbf{G}$ as in Example 4.2, but now $\mathbf{B}$ has only $m = 50$ columns. The left plot in Fig. 4.4 shows the norm of the reduced-order transfer function $\mathbf{Z}_{300}(2\pi\,\mathrm{i}\,\omega)$ for the frequency range $1 \leq \omega \leq 10^9$. The right plot in Fig. 4.4 shows the upper bound (4.11) for the corresponding error norm $\|\mathbf{Z}(s) - \mathbf{Z}_n(s)\|$ obtained after $n = 100$, 200, and 300 iterations of Algorithm 3.1.

5. Application to eigenvalue computations. In this section, we present some preliminary numerical examples to illustrate the application of Algorithm 3.1 to the solution of generalized symmetric definite eigenvalue problems of the form

$$\mathbf{K}\,\mathbf{x} = \lambda\,\mathbf{M}\,\mathbf{x}, \tag{5.1}$$

where $\mathbf{K} = \mathbf{K}^{\mathrm{T}} \in \mathbb{R}^{N \times N}$, $\mathbf{M} = \mathbf{M}^{\mathrm{T}} \in \mathbb{R}^{N \times N}$, and $\mathbf{M} > \mathbf{0}$.

The usual approach is to first compute a Cholesky decomposition $\mathbf{M} = \mathbf{L}\,\mathbf{L}^{\mathrm{T}}$ of

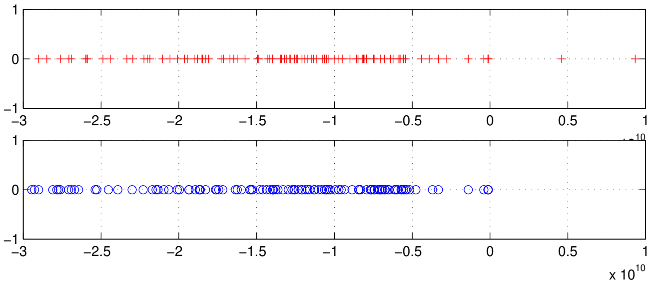

FIG. 4.3. *Dominant poles of SyMPVL reduced-order model (top) and SyMPVL2 reduced-order model (bottom) for Example 4.2.*

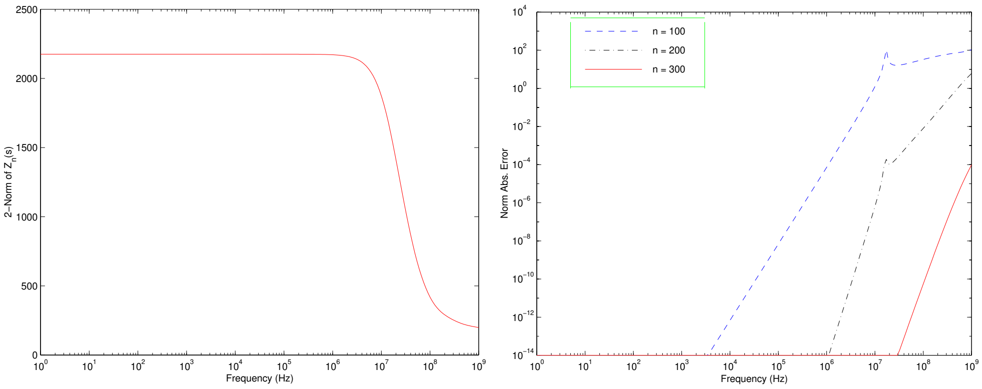

FIG. 4.4. $\|\mathbf{Z}_{300}(s)\|$ *(left) and upper bounds for* $\|\mathbf{Z}(s) - \mathbf{Z}_n(s)\|$ *(right) for Example 4.3.*

$\mathbf{M}$, and then convert (5.1) to the standard eigenvalue problem

$$(5.2) \qquad \left(\mathbf{L}^{-1}\,\mathbf{K}\,\mathbf{L}^{-\mathrm{T}}\right)\mathbf{y} = \lambda\,\mathbf{y}, \quad \text{where} \quad \mathbf{y} := \mathbf{L}^{\mathrm{T}}\,\mathbf{x}.$$

The band Lanczos process is then applied to the symmetric matrix $\mathbf{A} := \mathbf{L}^{-1}\,\mathbf{K}\,\mathbf{L}^{-\mathrm{T}}$ and a random block $\mathbf{B} \in \mathbb{R}^{N \times m}$ of m starting vectors. Note that the Lanczos process only requires matrix-vector products with $\mathbf{A}$. These can be computed by means of multiplications with $\mathbf{K}$ and backsolves with $\mathbf{L}$ and $\mathbf{L}^{\mathrm{T}}$, without explicitly forming $\mathbf{A}$.

Next, we present two numerical examples. These experiments were performed in Matlab, and in both cases, $m = 2$ starting vectors were used. In both examples, $\mathbf{K} \geq \mathbf{0}$, and hence the eigenvalues λ_k, $1 \leq k \leq N$, of (5.2) are all real and nonnegative.

Example 5.1. The matrices $\mathbf{K}$ and $\mathbf{M}$ in this example are of order $N = 1346$, and are taken from an application in circuit simulation. We ran both the standard symmetric band Lanczos process and the coupled Algorithm 3.1 for $n = 20$ and $n = 40$ iterations. The Ritz values are then computed as the eigenvalues of the matrices $\mathbf{T}_n$ (for the standard algorithm) and $\mathbf{U}_n^{\mathrm{T}}\,\boldsymbol{\Delta}_n\,\mathbf{U}_n$ (for Algorithm 3.1). In Fig. 5.1, we plot the computed Ritz values θ_i and their relative errors vs. their index i, $i = 1, 2, \ldots, n$, for both variants of the Lanczos process, and for $n = 20$ and 40. Here, the relative error of a computed Ritz value θ_i is defined as

$$(5.3) \qquad \frac{\min_{1 \leq k \leq N} |\theta_i - \lambda_k|}{|\lambda_{k_i}|}, \quad \text{where} \quad k_i := \operatorname{argmin}_{1 \leq k \leq N} |\theta_i - \lambda_k|.$$

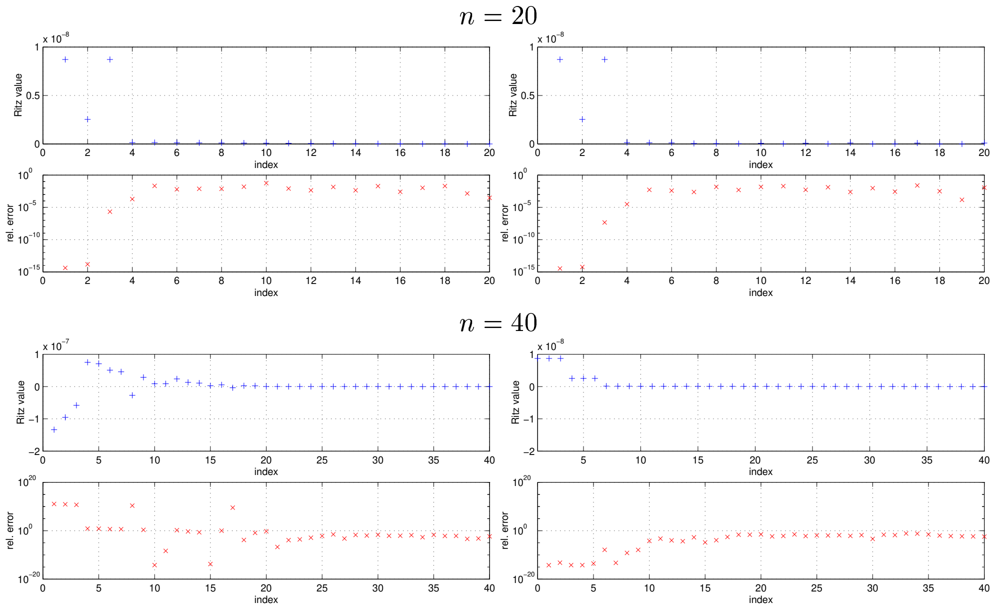

FIG. 5.1. *The standard (left) and the coupled (right) band Lanczos process for Example 5.1.*

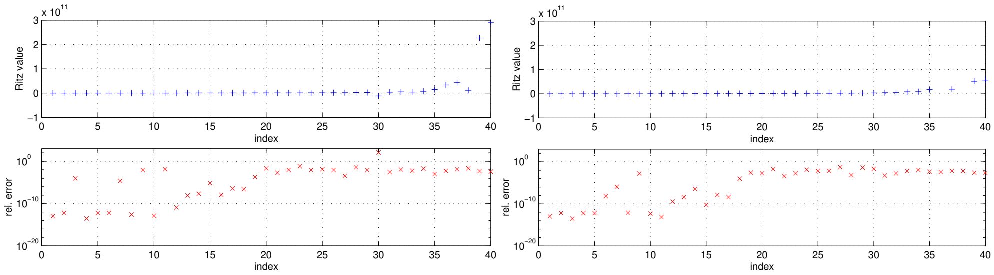

FIG. 5.2. *The standard (left) and the coupled (right) band Lanczos process for Example 5.2.*

Note that for $n = 20$, the computed Ritz values and their relative errors for both variants of the band Lanczos process are essentially the same. However, for $n = 40$, the standard algorithm is significantly worse than the coupled algorithm, and even has generated negative Ritz values. On the other hand, all the Ritz values from the coupled Algorithm 3.1 are positive.

Example 5.2. In this example, $\mathbf{K}$ and $\mathbf{M}$ are 834×834 stiffness and mass matrices arising in a structural analysis within MSC's NASTRAN application. In Fig. 5.2, we show the computed Ritz values, as well as their relative errors (5.3), that were obtained after $n = 40$ iterations of both variants of the band Lanczos process. Note that the standard algorithm produced one negative Ritz value, namely θ_{30}, while all Ritz values from the coupled Algorithm 3.1 are positive.

6. Concluding remarks. We proposed a variant of the band Lanczos process for symmetric matrices and multiple starting vectors that uses coupled recurrences involving two sets of basis vectors, instead of the recurrences involving only one set in the standard algorithm. The new variant generates the factors of an $\mathrm{LDL}^{\mathrm{T}}$ factorization of the Lanczos matrix, rather than the Lanczos matrix directly. Numerical experiments suggest that the coupled algorithm is superior to the standard algorithm, both in terms of accuracy and robustness. However, a precise round-off error analysis that would provide a theoretical basis for the superiority of the coupled algorithm still remains to be done.

For a single starting vector, the symmetric band Lanczos process reduces to the classical symmetric Lanczos algorithm [20] and the Lanczos matrix is tridiagonal. In the last few years, it has gradually become clear that the standard representation of a tridiagonal matrix via its entries is an unfortunate one, and that it is better, both for accuracy and efficiency, to represent the matrix as a product of bidiagonals; see, e.g., [8, 21, 22]. This paper has demonstrated that the same is true for the Lanczos matrix associated with the symmetric band Lanczos process. Instead of representing that matrix via its entries, it is preferable to present it via the entries of an $\mathrm{LDL}^{\mathrm{T}}$ factorization.

Acknowledgment. We would like to thank Peter Feldmann for his help with some of the numerical examples.

REFERENCES

[1] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernández, *A Lanczos-type method for multiple starting vectors*, Math. Comp., (2000, to appear).
Also available online from http://cm.bell-labs.com/cs/doc/98.

[2] B. Anderson and S. Vongpanitlerd, *Network Analysis and Synthesis*, Prentice-Hall, Englewood Cliffs, NJ, 1973.

[3] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, NJ, 1979.

[4] Z. Bai, P. Feldmann, and R. W. Freund, *How to make theoretically passive reduced-order models passive in practice*, in Proc. IEEE 1998 Custom Integrated Circuits Conference, Piscataway, NJ, 1998, IEEE, pp. 207–210.

[5] Z. Bai and Q. Ye, *Error estimation of the Padé approximation of transfer functions via the Lanczos process*, Electron. Trans. Numer. Anal., 7 (1998), pp. 1–17.

[6] R. H. Battin, *An Introduction to The Mathematics and Methods of Astrodynamics*, American Institute of Aeronautics and Astronautics, Inc., New York, 1987.

[7] J. K. Cullum and W. E. Donath, *A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace for large, sparse symmetric matrices*, in Proc. 1974 IEEE Conference on Decision and Control, New York, 1974, IEEE Press, pp. 505–509.

[8] I. S. Dhillon, *A new $O(n^2)$ algorithm for the symmetric tridiagonal eigenvalue/eigenvector problem*, PhD thesis, Department of Mathematics, University of California, Berkeley, CA, 1997.

[9] P. Feldmann and R. W. Freund, *Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm*, in Proc. 32nd ACM/IEEE Design Automation Conference, New York, 1995, ACM, pp. 474–479.

[10] R. W. Freund, *Computation of matrix Padé approximations of transfer functions via a Lanczos-type process*, in Approximation Theory VIII, Vol. 1: Approximation and Interpolation, C. Chui and L. Schumaker, eds., World Scientific Publishing Co., Inc., Singapore, 1995, pp. 215–222.

[11] ———, *Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation*, in Applied and Computational Control, Signals, and Circuits, B. N. Datta, ed., vol. 1, Birkhäuser, Boston, 1999, pp. 435–498.

[12] ———, *The Hermitian band Lanczos method*, in Templates for the Solution of Algebraic Eigenvalue Problems: a Practical Guide, Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., SIAM Publications, Philadelphia, PA, 2000, to appear.
Also available online from http://cm.bell-labs.com/cs/doc/99.

[13] ———, *Krylov-subspace methods for reduced-order modeling in circuit simulation*, J. Comput. Appl. Math., (2000, to appear).
Also available online from http://cm.bell-labs.com/cs/doc/99.

[14] R. W. Freund and P. Feldmann, *The SyMPVL algorithm and its applications to interconnect simulation*, in Proc. 1997 International Conference on Simulation of Semiconductor Processes and Devices, Piscataway, NJ, 1997, IEEE, pp. 113–116.

[15] ———, *Reduced-order modeling of large linear passive multi-terminal circuits using matrix-Padé approximation*, in Proc. Design, Automation and Test in Europe Conference 1998, Los Alamitos, CA, 1998, IEEE Computer Society Press, pp. 530–537.

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

[17] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, third ed., 1996.

[18] 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., (2000, to appear).

[19] S. Hammarling, *Numerical solution of the stable, non-negative definite Lyapunov equation*, IMA J. Numer. Anal., 2 (1982), pp. 303–323.

[20] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards, 45 (1950), pp. 255–282.

[21] B. N. Parlett, *The new QD algorithm*, Acta Numerica, 4 (1995), pp. 459–491.

[22] B. N. Parlett and K. Fernando, *Accurate singular values and differential QD algorithms*, Numer. Math., 67 (1994), pp. 191–229.

[23] A. Ruhe, *Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices*, Math. Comp., 33 (1979), pp. 680–687.

[24] M. Wohlers, *Lumped and Distributed Passive Networks*, Academic Press, New York, 1969.