Electronic Transactions on Numerical Analysis.
Volume 47, pp. 153–178, 2017.

ISSN 1068–9613.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

WEIGHTED GOLUB-KAHAN-LANCZOS BIDIAGONALIZATION ALGORITHMS*

HONG-XIU ZHONG† AND HONGGUO XU‡

Abstract. We present weighted Golub-Kahan-Lanczos algorithms. We demonstrate their applications to the eigenvalue problem of a product of two symmetric positive definite matrices and an eigenvalue problem for the linear response problem. A convergence analysis is provided and numerical test results are reported. As another application we make a connection between the proposed algorithms and the preconditioned conjugate gradient (PCG) method.

Key words. weighted Golub-Kahan-Lanczos bidiagonalization algorithm, eigenvalue, eigenvector, Ritz value, Ritz vector, linear response eigenvalue problem, Krylov subspace, bidiagonal matrices

AMS subject classifications. 65F15, 15A18

1. Introduction. The Golub-Kahan bidiagonalization factorization is fundamental for the QR-like singular value decomposition (SVD) method [7]. Based on this factorization, a Krylov subspace type method, called Golub-Kahan-Lanczos (GKL) algorithm, was developed in [11]. The Golub-Kahan-Lanczos algorithm provides a powerful tool for solving large-scale singular value and related eigenvalue problems, as well as least-squares and saddle-point problems [11, 12]. Recently, a generalized Golub-Kahan-Lanczos (gGKL) algorithm was introduced for solving generalized least-squares and saddle-point problems [1, 4].

In this paper we propose certain types of weighted Golub-Kahan-Lanczos bidiagonalization (wGKL) algorithms. The algorithms are based on the fact that for given symmetric positive definite matrices K and M, there exist a K-orthogonal matrix Y and an M-orthogonal matrix X such that $KY = XB$ and $MX = YB^T$, where B is either upper or lower bidiagonal. Two algorithms will be presented depending on whether B is upper or lower bidiagonal. The above relations are equivalent to $KMX = XBB^T$ and $MKY = YB^TB$. Since both BB^T and B^TB are symmetric tridiagonal, the wGKL algorithms are mathematically equivalent to the weighted Lanczos algorithm applied to the matrices KM and MK or the preconditioned Lanczos algorithms if K or M is the inverse of a matrix. However, in practice there is an important difference. The weighted Lanczos algorithm computes the columns of either X or Y and a leading principal submatrix of either BB^T or B^TB. The wGKL algorithms, on the other hand, compute both the columns of X and Y and a leading principal submatrix of B. In fact, as shown in the next section, the proposed algorithms can be viewed as a generalization of GKL [11] and also as a special case of gGKL [1]. Another feature of the wGKL algorithms is that they treat the matrices K and M equally.

The wGKL algorithms can be employed to compute the extreme eigenvalues and associated eigenvectors of the matrix products KM and MK $(= (KM)^T)$. The generalized eigenvalue problem $\lambda A - M$ with symmetric positive definite matrices A and M is one example, which is equivalent to the eigenvalue problem of KM with $K = A^{-1}$. Another application of the wGKL algorithms is the eigenvalue problem for matrices such as $\mathbf{H} = \begin{bmatrix} 0 & M \\ K & 0 \end{bmatrix}$ with symmetric positive definite K and M. Such an eigenvalue problem arises from the linear response problem in the time-dependent density functional theory and in the excitation energies of physical systems in the study of the collective motion of many-particle systems,

*Received December 15, 2016. Accepted September 18, 2017. Published online on November 3, 2017. Recommended by Gerard Meurant.

†School of Science, Jiangnan University, Wuxi, Jiangsu, 214122, P.R. China (zhonghongxiu@126.com).
‡Department of Mathematics, University of Kansas, Lawrence, KS, 66046, USA (xu@math.ku.edu).

which has applications in silicon nanoparticles and nanoscale materials and in the analysis of interstellar clouds [5, 9, 10, 14].

For a positive definite linear system, it is well known that the conjugate gradient (CG) method is equivalent to the standard Lanczos method, e.g., [15, Section 6.7] and [8]. As another application, we demonstrate that in the case when K or M is the identity matrix, the Krylov subspace linear system solver based on a wGKL algorithm provides a simpler and more direct connection to the CG method. In its original version (when neither K nor M is the identity matrix), such a solver is mathematically equivalent to a preconditioned CG (PCG) method.

The paper is organized as follows. In Section 2 we present the basic iteration schemes of the wGKL algorithms. In Section 3, we describe how to apply the wGKL algorithms to the eigenvalue problems with matrices KM or $\mathbf{H}$. A convergence analysis is provided as well. In Section 4, numerical examples for the eigenvalue problems are reported. In Section 5, the relation between wGKL and PCG is discussed, and Section 6 contains concluding remarks.

Throughout the paper, $\mathbb{R}$ is the real field, $\mathbb{R}^{m\times n}$ is the set of $m\times n$ real matrices, $\mathbb{R}^n$ is the n-dimensional real vector space, I_n is the $n\times n$ identity matrix, and e_j is its jth column. The notation $A>0$ (≥ 0) means that the matrix A is symmetric positive definite (semidefinite). For a given matrix $A\in\mathbb{R}^{n\times n}$ and a vector $b\in\mathbb{R}^n$, the kth Krylov subspace of A with b, i.e., the subspace spanned by the set of k vectors $\{b, Ab, \ldots, A^{k-1}b\}$, is denoted by $\mathcal{K}_k(A,b)$. $\|\cdot\|$ is the spectral norm for matrices and the 2-norm for vectors. For a given $n\times n$ symmetric positive definite matrix A, we introduce the weighted inner product $(x,y)_A = x^TAy$ in $\mathbb{R}^n$. The corresponding weighted norm, called A-norm, is defined by $\|x\|_A=\sqrt{(x,x)_A}$. A matrix X is A-orthonormal if $X^TAX=I$ (and it is A-orthogonal if X is a square matrix). A set of vectors $\{x_1,\ldots,x_k\}$ is also called A-orthonormal if $X=\begin{bmatrix}x_1 & \ldots & x_k\end{bmatrix}$ is A-orthonormal and A-orthogonal if $(x_i,x_j)_A=0$ for $i\neq j$. For any matrix A, $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ are the largest and the smallest singular values of A, respectively, and $\kappa_2(A)=\sigma_{\max}(A)/\sigma_{\min}(A)$ (when $\sigma_{\min}(A)>0$) is the condition number of A in the spectral norm.

In the paper we restrict ourselves to the real case. All the results can be easily extended to the complex case.

2. Weighted Golub-Kahan-Lanczos bidiagonalization (wGKL) algorithms.

The proposed wGKL algorithms are based on the following factorizations.

LEMMA 2.1. *Suppose that $0<K$ and $M\in\mathbb{R}^{n\times n}$. Then there exist an M-orthogonal matrix $X\in\mathbb{R}^{n\times n}$ and a K-orthogonal matrix $Y\in\mathbb{R}^{n\times n}$ such that*

$$KY = XB, \qquad MX = YB^T, \tag{2.1}$$

where B is either upper bidiagonal or lower bidiagonal.

Proof. In [7], it is shown that for any matrix A, there exist real orthogonal matrices U, V such that

$$AV = UB, \qquad A^TU = VB^T, \tag{2.2}$$

where B is either upper or lower bidiagonal. Since both $K, M>0$, one has the factorizations

$$K = LL^T, \qquad M = RR^T, \tag{2.3}$$

where both L and R are invertible. Take $A=R^TL$ in (2.2), and set

$$X = R^{-T}U, \qquad Y = L^{-T}V.$$

Then (2.2) becomes

$$R^TLL^TY = R^TXB, \qquad L^TRR^TX = L^TYB^T.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

By eliminating R^T in the first equation and L^T in the second equation, one has (2.1). Clearly $X^T MX = U^T U = I$ and $Y^T KY = V^T V = I$. ∎

The proof shows that (2.1) is a generalization of (2.2) by replacing the orthogonal matrices by weighted orthogonal matrices.

In [1] it is shown that for any matrix A, there exist an M-orthogonal matrix X and a K-orthogonal matrix Y such that

$$AY = MXB, \qquad A^T X = KYB^T.$$

By setting $A = MK$, we have again (2.1). Following these connections, the proposed wGKL algorithms can be considered a generalized version of GKL [11] and a special case of gGKL [1].

Based on the relations in (2.1) and the orthogonality of X and Y, we now construct two Lanczos-type iteration procedures corresponding to B being upper and lower bidiagonal, respectively. We first consider the upper bidiagonal case, and we call the procedure the upper bidiagonal version of the weighted Golub-Kahan-Lanczos algorithm (wGKL$_u$). Denote

$$X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}, \qquad Y = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix},$$

and

$$B = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ & \alpha_2 & \ddots & \\ & & \ddots & \beta_{n-1} \\ & & & \alpha_n \end{bmatrix}.$$

By comparing the columns of the relations in (2.1), one has

$$\begin{aligned} Ky_1 &= \alpha_1 x_1, & Mx_1 &= \alpha_1 y_1 + \beta_1 y_2, \\ Ky_2 &= \beta_1 x_1 + \alpha_2 x_2, & Mx_2 &= \alpha_2 y_2 + \beta_2 y_3, \\ &\ \vdots & &\ \vdots \\ Ky_k &= \beta_{k-1} x_{k-1} + \alpha_k x_k, & Mx_k &= \alpha_k y_k + \beta_k y_{k+1}, \\ &\ \vdots & &\ \vdots \\ Ky_n &= \beta_{n-1} x_{n-1} + \alpha_n x_n, & Mx_n &= \alpha_n y_n. \end{aligned}$$

Choosing an initial vector y_1 satisfying $y_1^T K y_1 = 1$ and using the orthogonality relation $x_i^T M x_j = y_i^T K y_j = \delta_{ij}$, where δ_{ij} is 0 if $i \neq j$ and 1 if $i = j$, the columns of X and Y as well as the entries of B can be computed by the following iterations:

$$\begin{aligned} \alpha_j &= \|Ky_j - \beta_{j-1} x_{j-1}\|_M, \\ x_j &= (Ky_j - \beta_{j-1} x_{j-1})/\alpha_j, \\ \beta_j &= \|Mx_j - \alpha_j y_j\|_K, \\ y_{j+1} &= (Mx_j - \alpha_j y_j)/\beta_j, \end{aligned}$$

with $x_0 = 0$ and $\beta_0 = 1$, for $j = 1, 2, \ldots$

We provide a concrete computational procedure that reduces the number of matrix-vector multiplications. Computing α_j requires the vector $f_j := M(Ky_j - \beta_{j-1} x_{j-1})$, which equals $\alpha_j M x_j$. The vector Mx_j appears in $Mx_j - \alpha_j y_j$ in the computation of β_j and y_{j+1}, which can now be obtained using f_j/α_j. In this way, we save one matrix-vector multiplication. Similarly, computing β_j needs the vector $g_{j+1} := K(Mx_j - \alpha_j y_j) = \beta_j K y_{j+1}$. The vector Ky_{j+1} is involved in the formulas for α_{j+1} and x_{j+1} and can thus be computed in the next iteration using g_{j+1}/β_j. Hence, another matrix-vector multiplication can be saved. The algorithm is detailed below.

ALGORITHM 1 (wGKL$_u$).
Choose y_1 satisfying $\|y_1\|_K = 1$, and set $\beta_0 = 1$, $x_0 = 0$. Compute $g_1 = Ky_1$.
For $j = 1, 2, \cdots$
  $s_j = g_j/\beta_{j-1} - \beta_{j-1}x_{j-1}$
  $f_j = Ms_j$
  $\alpha_j = (s_j^T f_j)^{\frac{1}{2}}$
  $x_j = s_j/\alpha_j$
  $t_{j+1} = f_j/\alpha_j - \alpha_j y_j$
  $g_{j+1} = Kt_{j+1}$
  $\beta_j = (t_{j+1}^T g_{j+1})^{\frac{1}{2}}$
  $y_{j+1} = t_{j+1}/\beta_j$
End

In each iteration, this algorithm requires two matrix-vector multiplications, and it needs five vectors $f_k, x_{k-1}, x_k, y_k, y_{k+1}$ to store the data (x_k, y_{k+1}, f_k may overwrite s_k, t_{k+1} and g_{k+1}.)

Suppose Algorithm 1 is run for k iterations. We then have $x_1, \ldots, x_k, y_1, \ldots, y_{k+1}$, and α_j, β_j for $j = 1, \ldots, k$. For any $j \geq 0$, define

$$X_j = \begin{bmatrix} x_1 & \ldots & x_j \end{bmatrix}, \quad Y_j = \begin{bmatrix} y_1 & \ldots & y_j \end{bmatrix}, \quad B_j = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ & \ddots & \ddots & \\ & & \ddots & \beta_{j-1} \\ & & & \alpha_j \end{bmatrix}.$$

Then we have the relations

$$KY_k = X_k B_k, \qquad MX_k = Y_k B_k^T + \beta_k y_{k+1} e_k^T = Y_{k+1} \begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}^T, \tag{2.4}$$

and

$$X_k^T M X_k = I_k = Y_k^T K Y_k.$$

Algorithm 1 may break down, but this happens only when $\beta_k = 0$ for some k. To see this, if $\prod_{j=1}^{k-1} \alpha_j \beta_j \neq 0$ but $\alpha_k = 0$, then one still has $KY_k = X_k B_k$ with the last column of X_k being zero. Since $K > 0$ and Y_k has full column rank, $\operatorname{rank} KY_k = k$. On the other hand, $\operatorname{rank} X_k B_k < k$, resulting in a contradiction. When $k = n$, β_n must be zero and (2.4) becomes (2.1).

From (2.4), one has

$$\begin{aligned} MKY_k &= Y_k B_k^T B_k + \alpha_k \beta_k y_{k+1} e_k^T, \\ KMX_k &= X_k (B_k B_k^T + \beta_k^2 e_k e_k^T) + \alpha_{k+1} \beta_k x_{k+1} e_k^T. \end{aligned} \tag{2.5}$$

Since $B_k B_k^T + \beta_k^2 e_k e_k^T$ and $B_k^T B_k$ are symmetric tridiagonal, it is obvious that wGKL$_u$ is equivalent to a weighted Lanczos algorithm applied to the matrices MK and $(MK)^T$, respectively. So we have

$$\operatorname{range} Y_k = \mathcal{K}_k(MK, y_1), \qquad \operatorname{range} X_k = \mathcal{K}_k(KM, Ky_1) = K\mathcal{K}_k(MK, y_1), \tag{2.6}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where we use the fact that x_1 is parallel to Ky_1 in the second relation.

When the matrix B in (2.1) is lower bidiagonal, a corresponding lower bidiagonal version of the weighted Golub-Kahan-Lanczos bidiagonalization algorithm (wGKL$_l$) can be derived in the same way. wGKL$_l$ is actually identical to wGKL$_u$ if we interchange the roles of K and M and X and Y in (2.1). In order to avoid confusion we use $\tilde{X}$, $\tilde{Y}$, $\tilde{B}$ instead of X, Y, B in (2.1), and we have

$$K\tilde{Y} = \tilde{X}\tilde{B}, \quad M\tilde{X} = \tilde{Y}\tilde{B}^T \qquad \text{with} \qquad \tilde{B} = \begin{bmatrix} \tilde{\alpha}_1 & & & \\ \tilde{\beta}_1 & \ddots & & \\ & \ddots & \ddots & \\ & & \tilde{\beta}_{n-1} & \tilde{\alpha}_n \end{bmatrix}, \tag{2.7}$$

and the wGKL$_l$ method is described by the following algorithm.

ALGORITHM 2 (wGKL$_l$).
Choose $\tilde{x}_1$ satisfying $\|\tilde{x}_1\|_M = 1$, and set $\tilde{\beta}_0 = 1$, $\tilde{y}_0 = 0$. Compute $g_1 = M\tilde{x}_1$.
For $j = 1, 2, \cdots$
  $s_j = g_j/\tilde{\beta}_{j-1} - \tilde{\beta}_{j-1}\tilde{y}_{j-1}$
  $f_j = Ks_j$
  $\tilde{\alpha}_j = (s_j^T f_j)^{\frac{1}{2}}$
  $\tilde{y}_j = s_j/\tilde{\alpha}_j$
  $t_{j+1} = f_j/\tilde{\alpha}_j - \tilde{\alpha}_j\tilde{x}_j$
  $g_{j+1} = Mt_{j+1}$
  $\tilde{\beta}_j = (t_{j+1}^T g_{j+1})^{\frac{1}{2}}$
  $\tilde{x}_{j+1} = t_{j+1}/\tilde{\beta}_j$
End

Similarly, by defining

$$\tilde{X}_j = \begin{bmatrix} \tilde{x}_1 & \dots & \tilde{x}_j \end{bmatrix}, \qquad \tilde{Y}_j = \begin{bmatrix} \tilde{y}_1 & \dots & \tilde{y}_j \end{bmatrix}, \qquad \tilde{B}_j = \begin{bmatrix} \tilde{\alpha}_1 & & & \\ \tilde{\beta}_1 & \ddots & & \\ & \ddots & \ddots & \\ & & \tilde{\beta}_{j-1} & \tilde{\alpha}_j \end{bmatrix},$$

one has

$$K\tilde{Y}_k = \tilde{X}_k\tilde{B}_k + \tilde{\beta}_k\tilde{x}_{k+1}e_k^T = \tilde{X}_{k+1}\begin{bmatrix} \tilde{B}_k \\ \tilde{\beta}_k e_k^T \end{bmatrix}, \qquad M\tilde{X}_k = \tilde{Y}_k\tilde{B}_k^T$$

and

$$\tilde{X}_k^T M\tilde{X}_k = I = \tilde{Y}_k^T K\tilde{Y}_k.$$

Also,

$$\begin{aligned} KM\tilde{X}_k &= \tilde{X}_k\tilde{B}_k\tilde{B}_k^T + \tilde{\alpha}_k\tilde{\beta}_k\tilde{x}_{k+1}e_k^T, \\ MK\tilde{Y}_k &= \tilde{Y}_k(\tilde{B}_k^T\tilde{B}_k + \tilde{\beta}_k^2 e_k e_k^T) + \tilde{\alpha}_{k+1}\tilde{\beta}_k\tilde{y}_{k+1}e_k^T, \end{aligned} \tag{2.8}$$

and

$$\operatorname{range}\tilde{X}_k = \mathcal{K}_k(KM, \tilde{x}_1), \qquad \operatorname{range}\tilde{Y}_k = \mathcal{K}_k(MK, M\tilde{x}_1) = M\mathcal{K}_k(KM, \tilde{x}_1).$$

Algorithm 2 breaks down only when $\tilde{\beta}_k = 0$ for some k.

3. Application to eigenvalue problems. In this section we discuss how to apply wGKL$_u$ and wGKL$_l$ to solve the eigenvalue problem for KM, MK, and $\mathbf{H} = \begin{bmatrix} 0 & M \\ K & 0 \end{bmatrix}$.

3.1. The eigenvalue problem for KM and MK. The relations in (2.5) and (2.8) show that Algorithms 1 and 2 can be employed to compute the eigenvalues of the matrices MK and KM. Note that $KM = (MK)^T$. So in the following, the discussion is mainly focused on the case for the matrix MK.

We first consider the approximations based on the first relation of (2.5) produced by wGKL$_u$. Suppose that B_k has an SVD

$$B_k = \Phi_k \Sigma_k \Psi_k^T, \qquad \Phi_k = \begin{bmatrix} \phi_1 & \dots & \phi_k \end{bmatrix}, \qquad \Psi_k = \begin{bmatrix} \psi_1 & \dots & \psi_k \end{bmatrix}, \qquad \Sigma_k = \operatorname{diag}(\sigma_1, \dots, \sigma_k), \tag{3.1}$$

with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k > 0$. Then, from the first relation in (2.5) for each $j \in \{1, \dots, k\}$, we may take σ_j^2 as a Ritz value of MK and $Y_k\psi_j$ as a corresponding right Ritz vector. Since Y_k is K-orthonormal and Ψ_k is real orthogonal, $Y_k\psi_1, \dots, Y_k\psi_k$ are K-orthonormal. Also, we have the residual formula

$$(MK - \sigma_j^2 I)Y_k\psi_j = \alpha_k\beta_k\psi_{jk}y_{k+1},$$

where ψ_{jk} is the kth component of ψ_j, for $j = 1, \dots, k$. Similarly, from the second relation in (2.5), for each $j \in \{1, \dots, k\}$, we may take $X_k\phi_j$ as a corresponding left Ritz vector of MK corresponding to the Ritz value σ_j^2. Note that $X_k\phi_1, \dots, X_k\phi_k$ are M-orthonormal, and from the first relation in (2.4),

$$X_k\phi_j = \sigma_j^{-1} X_k B_k \psi_j = \sigma_j^{-1} K Y_k \psi_j, \qquad j = 1, \dots, k. \tag{3.2}$$

Also, based on the second relation in (2.5) and the first relation in (2.4), one has the following residual formula (transposed)

$$(KM - \sigma_j^2 I)X_k\phi_j = \beta_k\phi_{jk}(\beta_k x_k + \alpha_{k+1}x_{k+1}) = \beta_k\phi_{jk}Ky_{k+1},$$

for $j = 1, \dots, k$, where ϕ_{jk} is the kth component of ϕ_j. In practice, we may use the residual norms

$$\begin{aligned} \|(MK - \sigma_j^2 I)Y_k\psi_j\|_K &= \alpha_k\beta_k|\psi_{jk}|, \\ \|(KM - \sigma_j^2 I)X_k\phi_j\|_M &= \beta_k|\phi_{jk}|\sqrt{\beta_k^2 + \alpha_{k+1}^2} \end{aligned} \tag{3.3}$$

to design a stopping criterion for wGKL$_u$.

The convergence properties can be readily established by employing the convergence theory of the standard Lanczos algorithm [8, 13, 16]. We need the following properties of the eigenvalue and eigenvectors of MK.

PROPOSITION 3.1. *The matrix MK has n positive eigenvalues $\lambda_1^2 \geq \lambda_2^2 \geq \ldots \geq \lambda_n^2$ with $\lambda_j > 0$ ($j = 1, \dots, n$). The corresponding right eigenvectors $\xi_1, \dots, \xi_n$ can be chosen K-orthonormal, and the corresponding left eigenvectors $\eta_1, \dots, \eta_n$ can be chosen M-orthonormal. In particular, for given $\{\xi_j\}$, one can choose $\eta_j = \lambda_j^{-1}K\xi_j$, for $j = 1, 2, \dots, n$, and for given $\{\eta_j\}$, $\xi_j = \lambda_j^{-1}M\eta_j$, for $j = 1, 2, \dots, n$.*

Proof. Using the factorization $K = LL^T$, MK is similar to $L^TML > 0$. Let

$$L^TML = Q\operatorname{diag}(\lambda_1^2, \dots, \lambda_n^2)Q^T,$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where Q is real orthogonal. Then $\lambda_1^2, \ldots, \lambda_n^2$ are the eigenvalues of MK, and $\xi_j = L^{-T}Qe_j$, for $j = 1, \ldots, n$, are the corresponding right eigenvectors. Clearly, $\xi_1, \ldots, \xi_n$ are K-orthonormal.

For each $\eta_j = \lambda_j^{-1}K\xi_j$, by premultiplying $\lambda_j^{-1}K$ to $MK\xi_j = \lambda_j^2\xi_j$, one has the relation $KM\eta_j = \lambda_j^2\eta_j$ or, equivalently, $\eta_j^T MK = \lambda_j^2\eta_j^T$. So $\eta_1, \ldots, \eta_n$ are the corresponding left eigenvectors of MK. The M-orthonormality can be obtained from

$$\eta_i^T M\eta_j = \lambda_i^{-1}\lambda_j^{-1}\xi_i^T KMK\xi_j = \frac{\lambda_j}{\lambda_i}\xi_i^T K\xi_j.$$

Thus, $\eta_i^T M\eta_j$ equals 1 if $i = j$ and 0 if $i \neq j$.

In the same way, we can show that $\xi_j = \lambda_j^{-1}M\eta_j$, for $j = 1, 2, \ldots, n$, are K-orthonormal right eigenvectors if $\{\eta_j\}$ is a set of M-orthonormal left eigenvectors. □

We need the following definitions. For two vectors $0 \neq x, y \in \mathbb{R}^n$ and $0 < A \in \mathbb{R}^{n\times n}$, we define the angles

$$\theta(x, y) = \arccos \frac{|x^T y|}{\|x\|\|y\|}, \qquad \theta_A(x, y) = \arccos \frac{|(x, y)_A|}{\|x\|_A\|y\|_A}.$$

We also denote by $C_j(x)$ the degree-j Chebyshev polynomial of the first kind.

The following convergence results are based on the theory given in [16].

THEOREM 3.2. *Let $\lambda_1^2 \geq \lambda_2^2 \geq \ldots \geq \lambda_n^2 > 0$ be the eigenvalues of MK with $\lambda_j > 0$, for $j = 1, \ldots, n$. Let $\xi_1, \ldots, \xi_n$ be the corresponding K-orthonormal right eigenvectors, and following Proposition 3.1, let $\eta_j := \lambda_j^{-1}K\xi_j$, $j = 1, \ldots, n$, be the corresponding M-orthonormal left eigenvectors. Suppose that B_k has an SVD (3.1) with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k > 0$. Consider the Ritz values $\sigma_1^2, \ldots, \sigma_k^2$, the corresponding K-orthonormal right Ritz vectors $Y_k\psi_1, \ldots, Y_k\psi_k$, and the M-orthonormal left Ritz vectors associated with MK, $X_k\phi_1, \ldots, X_k\phi_k$. Let*

$$\gamma_j = \frac{\lambda_j^2 - \lambda_{j+1}^2}{\lambda_{j+1}^2 - \lambda_n^2}, \qquad \tilde{\gamma}_j = \frac{\lambda_{n-k+j-1}^2 - \lambda_{n-k+j}^2}{\lambda_1^2 - \lambda_{n-k+j-1}^2}, \qquad 1 \leq j \leq k,$$

and y_1 be the initial vector in Algorithm 1. Then, for $j = 1, \ldots, k$,

$$0 \leq \lambda_j^2 - \sigma_j^2 \leq (\lambda_1^2 - \lambda_n^2)\left(\frac{\pi_{j,k}\tan\theta_K(y_1, \xi_j)}{C_{k-j}(1 + 2\gamma_j)}\right)^2 \tag{3.4}$$

with

$$\pi_{1,k} = 1, \quad \pi_{j,k} = \prod_{i=1}^{j-1}\frac{\sigma_i^2 - \lambda_n^2}{\sigma_i^2 - \lambda_j^2}, \qquad j > 1,$$

and

$$0 \leq \sigma_j^2 - \lambda_{n-k+j}^2 \leq (\lambda_1^2 - \lambda_n^2)\left(\frac{\tilde{\pi}_{j,k}\tan\theta_K(y_1, \xi_{n-k+j})}{C_{j-1}(1 + 2\tilde{\gamma}_j)}\right)^2 \tag{3.5}$$

with

$$\tilde{\pi}_{k,k} = 1, \quad \tilde{\pi}_{j,k} = \prod_{i=j+1}^{k}\frac{\sigma_i^2 - \lambda_1^2}{\sigma_i^2 - \lambda_{n-k+j}^2}, \qquad j < k.$$

The corresponding Ritz vectors have the following bounds:

$$\tag{3.6} \begin{aligned} &\sqrt{\left(\frac{\sigma_j}{\lambda_j}\right)^2 \sin^2\theta_M(X_k\phi_j,\eta_j) + 1 - \left(\frac{\sigma_j}{\lambda_j}\right)^2} \\ &\qquad = \sin\theta_K(Y_k\psi_j,\xi_j) \le \frac{\pi_j\sqrt{1+(\alpha_k\beta_k)^2/\delta_j^2}}{C_{k-j}(1+2\gamma_j)}\sin\theta_K(y_1,\xi_j) \end{aligned}$$

with $\delta_j = \min_{i\neq j}|\lambda_j^2 - \sigma_i^2|$ *and*

$$\pi_1 = 1, \quad \pi_j = \prod_{i=1}^{j-1}\frac{\lambda_i^2-\lambda_n^2}{\lambda_i^2-\lambda_j^2}, \qquad j>1,$$

and

$$\tag{3.7} \begin{aligned} &\sqrt{\left(\frac{\sigma_j}{\lambda_{n-k+j}}\right)^2 \sin\theta_M(X_k\phi_j,\eta_{n-k+j}) + 1 - \left(\frac{\sigma_j}{\lambda_{n-k+j}}\right)^2} \\ &\qquad = \sin\theta_K(Y_k\psi_j,\xi_{n-k+j}) \le \frac{\tilde\pi_j\sqrt{1+(\alpha_k\beta_k)^2/\tilde\delta_j^2}}{C_{j-1}(1+2\tilde\gamma_j)}\sin\theta_K(y_1,\xi_{n-k+j}) \end{aligned}$$

with $\tilde\delta_j = \min_{i\neq j}|\lambda_{n-k+j}^2 - \sigma_i^2|$ *and*

$$\tilde\pi_k = 1, \quad \tilde\pi_j = \prod_{i=n-k+j+1}^{n}\frac{\lambda_i^2-\lambda_1^2}{\lambda_i^2-\lambda_{n-k+j}^2}, \qquad j<k.$$

Proof. We first prove (3.4) and (3.6). As shown in the proof of Proposition 3.1, for any j, the vector $L^T\xi_j$ is a unit eigenvector of L^TML corresponding to the eigenvalue λ_j^2, and $L^T\xi_1,\ldots,L^T\xi_n$ are orthonormal. The first equation of (2.5) can be transformed to

$$\tag{3.8} L^TMLV_k = V_kB_k^TB_k + \alpha_k\beta_kv_{k+1}e_k^T,$$

where $V_k = L^TY_k$, $v_{k+1} = L^Ty_{k+1}$, and V_{k+1} is orthonormal, which can be considered as the relation derived by applying the standard Lanczos algorithm to L^TML. Hence, $\sigma_1^2,\ldots,\sigma_k^2$ are the Ritz values of L^TML, and $V_k\psi_1,\ldots,V_k\psi_k$ are the corresponding orthonormal right (left) Ritz vectors. Applying the standard Lanczos convergence results in [16, Section 6.6] to (3.8), one has

$$\begin{aligned} &0 \le \lambda_j^2 - \sigma_j^2 \le (\lambda_1^2-\lambda_n^2)\left(\frac{\pi_{j,k}\tan\theta(L^Ty_1,L^T\xi_j)}{C_{k-j}(1+2\gamma_j)}\right)^2, \\ &\sin\theta(V_k\psi_j,L^T\xi_j) \le \frac{\pi_j\sqrt{1+(\alpha_k\beta_k)^2/\delta_j^2}}{C_{k-j}(1+\gamma_j)}\sin\theta(L^Ty_1,L^T\xi_j), \end{aligned}$$

where $\pi_{j,k},\pi_j,\delta_j,\gamma_j$ are defined in the theorem. The bounds (3.4) and (3.6) can be derived simply by using the identities

$$\theta(L^Ty_1,L^T\xi_j) = \theta_K(y_1,\xi_j), \qquad \theta(V_k\psi_j,L^T\xi_j) = \theta(L^TY_k\psi_j,L^T\xi_j) = \theta_K(Y_k\psi_j,\xi_j).$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

We still need to prove equality in (3.6). By (3.2),

$$\begin{aligned}\cos\theta_M(\eta_j, X_k\phi_j) &= \cos\theta_M(K\xi_j/\lambda_j, KY_k\psi_j/\sigma_j) = \frac{|\psi_j^T Y_k^T KMK\xi_j|}{\sigma_j\lambda_j}\\ &= \frac{\lambda_j}{\sigma_j}|\psi_j^T Y_k^T K\xi_j| = \frac{\lambda_j}{\sigma_j}\cos\theta_K(Y_k\psi_j, \xi_j).\end{aligned} \tag{3.9}$$

Hence

$$\cos\theta_K(Y_k\psi_j, \xi_j) = \frac{\sigma_j}{\lambda_j}\cos\theta_M(X_k\phi_j, \eta_j), \tag{3.10}$$

from which one obtains

$$\sin\theta_K(Y_k\psi_j, \xi_j) = \sqrt{\left(\frac{\sigma_j}{\lambda_j}\right)^2 \sin^2\theta_M(X_k\phi_j, \eta_j) + 1 - \left(\frac{\sigma_j}{\lambda_j}\right)^2}.$$

The bounds (3.5) and (3.7) can be proved by applying these results to the matrix $(-MK)$. The equality in (3.7) can be established from the identity

$$\cos\theta_K(Y_k\psi_j, \xi_{n-k+j}) = \frac{\sigma_j}{\lambda_{n-k+j}}\cos\theta_M(X_k\phi_j, \eta_{n-k+j}), \tag{3.11}$$

which can be derived in the same way as (3.10). □

Clearly, the second relation in (2.5) can also be used to approximate the eigenvalues and eigenvectors of MK by using the SVD

$$\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix} \begin{bmatrix} \omega_1 & \dots & \omega_{k+1} \end{bmatrix} = \begin{bmatrix} \zeta_1 & \dots & \zeta_k \end{bmatrix} \begin{bmatrix} \rho_1 & 0 & & 0\\ & \ddots & \ddots & \\ 0 & & \rho_k & 0 \end{bmatrix}. \tag{3.12}$$

In this situation, $\rho_1^2, \dots, \rho_k^2$ are the Ritz values and $X_k\zeta_1, \dots, X_k\zeta_k$ are the corresponding M-orthonormal left (right) Ritz vectors of MK (KM). The residual formula transposed yields

$$(KM - \rho_j^2 I)X_k\zeta_j = \alpha_{k+1}\beta_k\zeta_{jk}x_{k+1},$$

where ζ_{jk} is the kth component of ζ_j. From the first equation of (2.5) with k replaced by $k+1$,

$$\begin{aligned}MKY_{k+1} &= Y_{k+1}\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}^T \begin{bmatrix} B_k & \beta_k e_k \end{bmatrix} + \alpha_{k+1}(\alpha_{k+1}y_{k+1} + \beta_{k+1}y_{k+2})e_{k+1}^T\\ &= Y_{k+1}\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}^T \begin{bmatrix} B_k & \beta_k e_k \end{bmatrix} + \alpha_{k+1}Mx_{k+1}e_{k+1}^T.\end{aligned}$$

So for each $j \in \{1, \dots, k\}$,

$$(MK - \rho_j^2 I)Y_{k+1}\omega_j = \alpha_{k+1}\omega_{j,k+1}Mx_{k+1} = \alpha_{k+1}\omega_{j,k+1}(\alpha_{k+1}y_{k+1} + \beta_{k+1}y_{k+2}),$$

where $\omega_{j,k+1}$ is the $(k+1)$st component of ω_j. Hence $Y_{k+1}\omega_1, \dots, Y_{k+1}\omega_k$ can be taken as the right Ritz vectors of MK, and we have the following residual norm formulas

$$\begin{aligned}\|(KM - \rho_j^2 I)X_k\zeta_j\|_M &= \alpha_{k+1}\beta_k|\zeta_{jk}|,\\ \|(MK - \rho_j^2 I)Y_{k+1}\omega_j\|_K &= \alpha_{k+1}|\omega_{j,k+1}|\sqrt{\alpha_{k+1}^2 + \beta_{k+1}^2}.\end{aligned} \tag{3.13}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Note that by post-multiplying the second equation in (2.4) with ω_j, one has

$$Y_{k+1}\omega_j = \rho_j^{-1} Y_{k+1} \begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}^T \zeta_j = \rho_j^{-1} M X_k \zeta_j, \qquad j = 1, \ldots, k. \tag{3.14}$$

The same type of convergence theory can be established.

THEOREM 3.3. *Let $\lambda_1^2 \geq \lambda_2^2 \geq \ldots \geq \lambda_n^2 > 0$ be the eigenvalues of MK with $\lambda_j > 0$, for $j = 1, \ldots, n$. Let $\eta_1, \ldots, \eta_n$ be the corresponding M-orthonormal left eigenvectors associated with MK, and following Proposition 3.1, let $\xi_j = \lambda_j^{-1} M \eta_j$, $j = 1, \ldots, n$, be the corresponding K-orthonormal right eigenvectors. Suppose that $\rho_1 \geq \ldots \geq \rho_k$ are the singular values of $\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}$, $\zeta_1, \ldots, \zeta_k$ the corresponding orthonormal left singular vectors, and $\omega_1, \ldots, \omega_k$ the corresponding orthonormal right singular vectors as defined in (3.12). Let γ_j, $\tilde{\gamma}_j$, π_j, and $\tilde{\pi}_j$ be defined in Theorem 3.2 and $x_1 = Ky_1/\|Ky_1\|_M$ be generated by Algorithm 1. Then, for $j = 1, \ldots, k$,*

$$0 \leq \lambda_j^2 - \rho_j^2 \leq (\lambda_1^2 - \lambda_n^2) \left(\frac{\kappa_{j,k} \tan \theta_M(x_1, \eta_j)}{C_{k-j}(1 + 2\gamma_j)} \right)^2$$

with

$$\kappa_{1,k} = 1, \qquad \kappa_{j,k} = \prod_{i=1}^{j-1} \frac{\rho_i^2 - \lambda_n^2}{\rho_i^2 - \lambda_j^2}, \qquad j > 1,$$

and

$$0 \leq \rho_j^2 - \lambda_{n-k+j}^2 \leq (\lambda_1^2 - \lambda_n^2) \left(\frac{\tilde{\kappa}_{j,k} \tan \theta_M(x_1, \eta_{n-k+j})}{C_{j-1}(1 + 2\tilde{\gamma}_j)} \right)^2$$

with

$$\tilde{\kappa}_{k,k} = 1, \qquad \tilde{\kappa}_{j,k} = \prod_{i=j+1}^{k} \frac{\rho_i^2 - \lambda_1^2}{\rho_i^2 - \lambda_{n-k+j}^2}, \qquad j < k.$$

The corresponding Ritz vectors of MK have the following bounds:

$$\begin{aligned} &\sqrt{\left(\frac{\rho_j}{\lambda_j}\right)^2 \sin^2 \theta_K(Y_{k+1}\omega_j, \xi_j) + 1 - \left(\frac{\rho_j}{\lambda_j}\right)^2} \\ &\quad = \sin \theta_M(X_k \zeta_j, \eta_j) \leq \frac{\pi_j \sqrt{1 + (\alpha_{k+1}\beta_k)^2/\epsilon_j^2}}{C_{k-j}(1 + 2\gamma_j)} \sin \theta_M(x_1, \eta_j), \end{aligned} \tag{3.15}$$

with $\epsilon_j = \min_{i \neq j} |\lambda_j^2 - \rho_i^2|$, and

$$\begin{aligned} &\sqrt{\left(\frac{\rho_j}{\lambda_{n-k+j}}\right)^2 \sin^2 \theta_K(Y_{k+1}\omega_j, \xi_{n-k+j}) + 1 - \left(\frac{\rho_j}{\lambda_{n-k+j}}\right)^2} \\ &\quad = \sin \theta_M(X_k \zeta_j, \eta_{n-k+j}) \leq \frac{\tilde{\pi}_j \sqrt{1 + (\alpha_{k+1}\beta_k)^2/\tilde{\epsilon}_j^2}}{C_{j-1}(1 + 2\tilde{\gamma}_j)} \sin \theta_M(x_1, \eta_{n-k+j}), \end{aligned} \tag{3.16}$$

with $\tilde{\epsilon}_j = \min_{i \neq j} |\lambda_{n-k+j}^2 - \rho_i^2|$.

Proof. The bounds can be established by applying the standard Lanczos convergence results to

$$R^TKRU_k = U_k(B_kB_k^T + \beta_k^2 e_k e_k^T) + \alpha_{k+1}\beta_k u_{k+1} e_k^T,$$

which is obtained from the second relation of (2.5) with $M = RR^T$, $U_k = R^TX_k$, and $u_{k+1} = R^Tx_{k+1}$.

By (3.14),

$$\begin{aligned}\cos\theta_K(\xi_j, Y_{k+1}\omega_j) &= |\lambda_j^{-1}\rho_j^{-1}\eta_j^T MKMX_k\zeta_j| \\ &= \frac{\lambda_j}{\rho_j}|\eta_j^T MX_k\zeta_j| = \frac{\lambda_j}{\rho_j}\cos\theta_M(X_k\zeta_j, \eta_j),\end{aligned}$$

from which equality in (3.15) can be derived.

Similarly, one has

$$\cos\theta_K(\xi_{n-k+j}, Y_{k+1}\omega_j) = \frac{\lambda_{n-k+j}}{\rho_j}\cos\theta_M(X_k\zeta_j, \eta_{n-k+j}),$$

which yields equality in (3.16). ∎

REMARK 3.4. Since $\sigma_1, \ldots, \sigma_k$ are the singular values of B_k and $\rho_1, \ldots, \rho_k$ are the singular values of $\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}$, from the interlacing properties [8, Corollary 8.6.3], one has

$$\rho_1 \ge \sigma_1 \ge \rho_2 \ge \sigma_2 \ge \ldots \ge \rho_k \ge \sigma_k.$$

From Theorems 3.2 and 3.3, for approximating a large eigenvalue λ_j^2 of MK, ρ_j^2 will be more accurate than σ_j^2 since ρ_j^2 is closer to λ_j^2. Similarly, for approximating a small eigenvalue λ_j^2, σ_j^2 will be more accurate than ρ_j^2. For instance, if we need to approximate λ_1^2, ρ_1^2 is more precise than σ_1^2, and for λ_n^2, σ_k^2 is preferable over ρ_k^2.

REMARK 3.5. Theorems 3.2 and 3.3 provide convergence results for both the left and right eigenvectors of MK as well as $KM = (MK)^T$. The values of $\sin\theta_K(y_1, \xi_j)$ and $\sin\theta_M(x_1, \eta_j)$ represent the influence of the initial vectors y_1 and x_1 to the approximated eigenvectors (and also the approximated eigenvalues). In general, the angles $\theta_K(y_1, \xi_j)$ and $\theta_M(x_1, \eta_j)$ are different, but they are related. Recall that, $x_1 = Ky_1/\|Ky_1\|_M$, $\eta_j = \lambda_j^{-1}K\xi_j$, and $MK\xi_j = \lambda_j^2\xi_j$. So

$$\begin{aligned}\cos\theta_M(x_1, \eta_j) &= |x_1^TM\eta_j| = \frac{|y_1^TKMK\xi_j|}{\lambda_j\|Ky_1\|_M} \\ &= \frac{\lambda_j}{\|Ky_1\|_M}|y_1^TK\xi_j| = \frac{\lambda_j}{\|Ky_1\|_M}\cos\theta_K(y_1, \xi_j).\end{aligned}$$

Because

$$\|Ky_1\|_M^2 = y_1^TKMKy_1 = (L^Ty_1)^T(L^TML)(L^Ty_1), \quad (L^Ty_1)^T(L^Ty_1) = y_1^TKy_1 = 1,$$

and $\lambda_1^2, \ldots, \lambda_n^2$ are the eigenvalues of L^TML, one has

$$\lambda_n \le \|Ky_1\|_M \le \lambda_1.$$

Therefore

$$\frac{\lambda_j}{\lambda_1}\cos\theta_K(y_1, \xi_j) \le \cos\theta_M(x_1, \eta_j) \le \frac{\lambda_j}{\lambda_n}\cos\theta_K(y_1, \xi_j).$$

REMARK 3.6. The convergence results established in Theorems 3.2 and 3.3 are similar to the ones given in [16] for the standard Lanczos algorithm applied to the symmetric matrix L^TML or R^TKR, where $K = LL^T$ and $M = RR^T$. The results indicate that the Ritz values and Ritz vectors corresponding to the extreme eigenvalues λ_1^2 and λ_n^2 converge faster than the rest. Unlike the standard results, where the left and right Ritz vectors corresponding to the same Ritz value can be the same, for each j, the angles between left and right Ritz vectors and the corresponding eigenvectors are different, cf., (3.10) and (3.11). On the other hand, these relations show that the two angles are essentially the same when the Ritz value is close to the corresponding eigenvalue.

REMARK 3.7. From the first relation in (2.5), Algorithm 1 (wGKL$_u$) is mathematically equivalent to a weighted Lanczos algorithm applied to MK (by forcing $Y_k^TKY_k = I$). Algorithm 1 needs two additional scalar-vector multiplications per iteration and additional storage for saving the vectors $x_1, \ldots, x_k$. On the other hand, with Algorithm 1 we are able to provide both left and right Ritz vectors simultaneously. Another advantage of Algorithm 1 is that the eigenvalues of MK can be approximated by using the singular values of $\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}$, which may yield more accurate approximations for the large eigenvalues of MK. If we use the singular values and vectors of $\tilde{B}_k$ and $\begin{bmatrix} \tilde{B}_k \\ \tilde{\beta}_k e_k^T \end{bmatrix}$, which are generated by wGKL$_l$, to approximate the eigenvalues and eigenvectors of MK and KM, a convergence theory as in the Theorems 3.2, 3.3 can be established in the same way.

For the rest of this section we discuss the relations between the two algorithms wGKL$_u$ and wGKL$_l$. Denote $U = R^TX$, $V = L^TY$, $\tilde{U} = R^T\tilde{X}$, $\tilde{V} = L^T\tilde{Y}$, where the matrices are those from (2.1), (2.3), and (2.7). All of them are orthogonal matrices. Note that from (2.2) with $A = R^TL$, one finds

$$R^TL = UBV^T = \tilde{U}\tilde{B}\tilde{V}^T.$$

Thus,

$$\tilde{U}^TUB = \tilde{B}\tilde{V}^TV.$$

If we choose y_1 and set $\tilde{x}_1 = x_1 = Ky_1/\|Ky_1\|_M$, then the first columns of $\tilde{U}$ and U are identical or the first column of $\tilde{U}^TU$ is e_1. Since $\tilde{U}^TUBB^T(\tilde{U}^TU)^T = \tilde{B}\tilde{B}^T$ is a tridiagonal reduction of BB^T, if all $\beta_j, \alpha_j, \tilde{\beta}_j, \tilde{\alpha}_j$ are positive, then by the implicit-Q Theorem [8], $\tilde{U}^TU = I$, i.e., $U = \tilde{U}$, or equivalently, $X = \tilde{X}$. Then $\tilde{B} = BQ$ with $Q = V^T\tilde{V}$ is an RQ factorization of the lower bidiagonal matrix $\tilde{B}$. Hence, when wGKL$_u$ starts with y_1 and wGKL$_l$ starts with $\tilde{x}_1 = Ky_1/\|Ky_1\|_M$, if both algorithms can be run for n iterations, then the generated matrices satisfy $\tilde{X} = X$, $\tilde{Y} = YQ$. Since

$$\tilde{B}^T\tilde{B} = Q^TB^TBQ,$$

it is not difficult to see that Q is just the orthogonal matrix generated by applying one QR iteration from B^TB to $\tilde{B}^T\tilde{B}$ with zero shift [8].

Clearly, one has

$$BB^T = \tilde{B}\tilde{B}^T.$$

For any integer $1 \le k \le n$, by comparing the leading $k \times k$ principal submatrices of BB^T and $\tilde{B}\tilde{B}^T$, one has

$$\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix} \begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}^T = \tilde{B}_k\tilde{B}_k^T. \tag{3.17}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

So the singular values of $\tilde{B}_k$ and $\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}$ are identical.

Now we have four matrices

$$\begin{bmatrix} \tilde{B}_k \\ \tilde{\beta}_k e_k^T \end{bmatrix}, \qquad \tilde{B}_k, \qquad \begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}, \qquad B_k,$$

and the singular values of each matrix can be used for eigenvalue approximations. Following the same arguments given in Remark 3.4, the squares of the large singular values of $\begin{bmatrix} \tilde{B}_k \\ \tilde{\beta}_k e_k^T \end{bmatrix}$ are closer to the large eigenvalues of MK than those of $\tilde{B}_k$. So they are also closer than those of $\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}$ and B_k. Similarly, the squares of the small singular values of B_k are closest to the small eigenvalues of MK among those of the above four matrices. We illustrate this feature by a numerical example in the next section.

Similarly, when wGKL$_l$ starts with $\tilde{x}_1$ and wGKL$_u$ starts with $y_1 = \tilde{y}_1 = M\tilde{x}_1/\|M\tilde{x}_1\|_K$, we have $\tilde{Y} = Y$ and $\tilde{X} = X\tilde{Q}$ with the orthogonal matrix $\tilde{Q}$ satisfying $B = \tilde{Q}\tilde{B}$. This has the interpretation that $\tilde{Q}$ is obtained by performing one QR iteration on $\tilde{B}\tilde{B}^T$ with zero shift. In this case, among the above four matrices $\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}$ will provide the best approximations to the large eigenvalues of MK, and $\tilde{B}_k$ will provide the best approximations to the small eigenvalues of MK.

3.2. The linear response eigenvalue problem. In this section we apply the algorithms wGKL$_u$ and wGKL$_l$ to solve the eigenvalue problem for the matrix

$$\mathbf{H} = \begin{bmatrix} 0 & M \\ K & 0 \end{bmatrix}, \qquad 0 < K, M \in \mathbb{R}^{n\times n}.$$

Such an eigenvalue problem arises in the linear response problem [2, 3, 5, 9, 10, 14]. We only consider wGKL$_u$ since the results about wGKL$_l$ can be established in the same way. Let X_k, Y_k, B_k be generated by Algorithm 1 after k iterations. Define

$$\mathbf{X}_j = \begin{bmatrix} Y_j & 0 \\ 0 & X_j \end{bmatrix}, \qquad \mathbf{B}_j = \begin{bmatrix} 0 & B_j^T \\ B_j & 0 \end{bmatrix}.$$

Then from (2.4),

$$\mathbf{H}\mathbf{X}_k = \mathbf{X}_k\mathbf{B}_k + \beta_k \begin{bmatrix} y_{k+1} \\ 0 \end{bmatrix} e_{2k}^T. \tag{3.18}$$

Let

$$\tilde{\mathbf{P}}_k = \begin{bmatrix} e_1 & e_{k+1} & e_2 & e_{k+2} & \dots & e_k & e_{2k} \end{bmatrix}.$$

One has

$$\mathbf{H}(\mathbf{X}_k\tilde{\mathbf{P}}_k) = (\mathbf{X}_k\tilde{\mathbf{P}}_k)(\tilde{\mathbf{P}}_k^T\mathbf{B}_k\tilde{\mathbf{P}}_k) + \beta_k \begin{bmatrix} y_{k+1} \\ 0 \end{bmatrix} e_{2k}^T, \tag{3.19}$$

where $\tilde{\mathbf{P}}_k^T\mathbf{B}_k\tilde{\mathbf{P}}_k$ is a symmetric tridiagonal matrix with zero diagonal entries and

$$\mathbf{X}_k\tilde{\mathbf{P}}_k = \begin{bmatrix} y_1 & 0 & y_2 & 0 & \dots & y_{k-1} & 0 & y_k & 0 \\ 0 & x_1 & 0 & x_2 & \dots & 0 & x_{k-1} & 0 & x_k \end{bmatrix}.$$

Using (2.6), one has

$$\operatorname{range} \mathbf{X}_k = \operatorname{range} \mathbf{X}_k\tilde{\mathbf{P}}_k = \mathcal{K}_{2k}\left(\mathbf{H}, \begin{bmatrix} y_1 \\ 0 \end{bmatrix}\right).$$

So, running k iterations of wGKL$_u$ is just the same as running $2k$ iterations of a weighted Lanczos algorithm with $\mathbf{H}$ and an initial vector of the special form $\begin{bmatrix} y_1^T & 0 \end{bmatrix}^T$.

Define

$$\mathbf{K} = \begin{bmatrix} K & 0 \\ 0 & M \end{bmatrix}, \qquad \mathbf{M} = \begin{bmatrix} M & 0 \\ 0 & K \end{bmatrix}.$$

Suppose B_k has an SVD (3.1) with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k > 0$. From (3.18), we may take $\pm\sigma_1, \ldots, \pm\sigma_k$ as Ritz values of $\mathbf{H}$ and

$$\mathbf{v}_j^{\pm} = \frac{1}{\sqrt{2}} \begin{bmatrix} Y_k \psi_j \\ \pm X_k \phi_j \end{bmatrix}, \qquad j = 1, \ldots, k,$$

as the corresponding $\mathbf{K}$-orthonormal right Ritz vectors, and from

$$\mathbf{H}^T = \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix} \mathbf{H} \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix}, \tag{3.20}$$

one may take

$$\mathbf{u}_j^{\pm} = \frac{1}{\sqrt{2}} \begin{bmatrix} \pm X_k \phi_j \\ Y_k \psi_j \end{bmatrix}, \qquad j = 1, \ldots, k,$$

as the corresponding $\mathbf{M}$-orthonormal left Ritz vectors.

From (3.18), for any $j \in \{1, \ldots, k\}$,

$$\mathbf{H}\mathbf{v}_j^{\pm} = \pm\sigma_j \mathbf{v}_j^{\pm} \pm \frac{\beta_k \phi_{jk}}{\sqrt{2}} \begin{bmatrix} y_{k+1} \\ 0 \end{bmatrix}, \qquad \mathbf{H}^T \mathbf{u}_j^{\pm} = \pm\sigma_j \mathbf{u}_j^{\pm} \pm \frac{\beta_k \phi_{jk}}{\sqrt{2}} \begin{bmatrix} 0 \\ y_{k+1} \end{bmatrix},$$

where ϕ_{jk} is the kth component of ϕ_j. In practice, we may use the residual norm

$$\|\mathbf{H}\mathbf{v}_j^{+} - \sigma_j \mathbf{v}_j^{+}\|_{\mathbf{K}} = \|\mathbf{H}^T \mathbf{u}_j^{+} - \sigma_j \mathbf{u}_j^{+}\|_{\mathbf{M}} = \frac{1}{\sqrt{2}} \|M X_k \phi_j - \sigma_j Y_k \psi_j\|_K = \frac{\beta_k |\phi_{jk}|}{\sqrt{2}} \tag{3.21}$$

to design a stopping criterion. When $\beta_k = 0$ for some k, all $\pm\sigma_j$ are eigenvalues of $\mathbf{H}$ and $\mathbf{u}_j^{\pm}$ and $\mathbf{v}_j^{\pm}$ are the corresponding left and right eigenvectors for $j = 1, \ldots, k$.

REMARK 3.8. In general, based on (3.19), if (θ_j, g_j), $j = 1, \ldots, 2k$, are the eigenpairs of $\tilde{\mathbf{P}}_k^T \mathbf{B}_k \tilde{\mathbf{P}}_k$, i.e., $\tilde{\mathbf{P}}_k^T \mathbf{B}_k \tilde{\mathbf{P}}_k g_j = \theta_j g_j$ with $g_1, \ldots, g_{2k}$ orthonormal, then (θ_i, q_i), for $i = 1, \ldots, 2k$, are the approximate eigenpairs of $\mathbf{H}$, where $q_i = \mathbf{X}_k \tilde{\mathbf{P}}_k g_i$ and

$$\|\mathbf{H} q_i - \theta_i q_i\|_{\mathbf{K}} = \left\| \beta_k \begin{bmatrix} y_{k+1} \\ 0 \end{bmatrix} e_{2k}^T g_i \right\|_{\mathbf{K}} = \beta_k |g_{i,2k}|, \tag{3.22}$$

where $g_{i,2k}$ is the $2k$th component of g_i.

REMARK 3.9. Although it is quite natural to use the weighted norms in (3.21) and (3.22) to measure the residual errors, in the numerical examples given below, we will use the 1-norm instead to keep the computations simple.

A basic algorithm for solving the linear response eigenvalue problem reads as follows.

ALGORITHM 3 (wGKL$_u$-LREP).

1. Run k steps of Algorithm 1 with an initial y_1 and an appropriate integer k to generate B_k, Y_k, and X_k.
2. Compute an SVD of B_k as in (3.1), select $l (\leq k)$ wanted singular value σ_j, and the associated left and right singular vector ϕ_j and ψ_j, $j = 1, \ldots, l$.

3. Form $\pm\sigma_j$, $\mathbf{v}_j^{\pm} = \frac{1}{\sqrt{2}}\begin{bmatrix} Y_k\psi_j \\ \pm X_k\phi_j \end{bmatrix}$, and $\mathbf{u}_j^{\pm} = \frac{1}{\sqrt{2}}\begin{bmatrix} \pm X_k\phi_j \\ Y_k\psi_j \end{bmatrix}$, for $j = 1, \ldots, l$.

For a convergence analysis we need some basic properties about the eigenvalues and eigenvectors of $\mathbf{H}$. From (2.1) and the fact that X is M-orthogonal and Y is K-orthogonal, for $\mathbf{X} = \begin{bmatrix} Y & 0 \\ 0 & X \end{bmatrix}$, one has

$$\mathbf{HX} = \mathbf{XB}, \qquad \mathbf{B} = \begin{bmatrix} 0 & B^T \\ B & 0 \end{bmatrix}, \qquad \mathbf{X}^T\mathbf{KX} = I_{2n}.$$

Thus, $\mathbf{H}$ is similar to the symmetric matrix $\mathbf{B}$ with a $\mathbf{K}$-orthogonal transformation matrix $\mathbf{X}$. Moreover, suppose $B = \Phi\Lambda\Psi^T$ is an SVD of B. Define the symmetric orthogonal matrix $\mathbf{P} = \frac{1}{\sqrt{2}}\begin{bmatrix} I_n & I_n \\ I_n & -I_n \end{bmatrix}$. Then,

$$\mathbf{H}\begin{bmatrix} Y\Psi & 0 \\ 0 & X\Phi \end{bmatrix}\mathbf{P} = \begin{bmatrix} Y\Psi & 0 \\ 0 & X\Phi \end{bmatrix}\mathbf{P}\begin{bmatrix} \Lambda & 0 \\ 0 & -\Lambda \end{bmatrix}.$$

Hence, $\pm\lambda_1, \ldots, \pm\lambda_n$ are the eigenvalues of $\mathbf{H}$. Define

$$\xi_j = Y\Psi e_j, \qquad \eta_j = X\Phi e_j,$$

for $j = 1, 2, \ldots, n$. Then $\xi_1, \ldots, \xi_n$ are K-orthonormal, and $\eta_1, \ldots, \eta_n$ are M-orthonormal, and by defining

$$\left[\mathbf{x}_1^+, \ldots, \mathbf{x}_n^+, \mathbf{x}_1^-, \ldots \mathbf{x}_n^-\right] := \begin{bmatrix} Y\tilde{\Psi} & 0 \\ 0 & X\tilde{\Phi} \end{bmatrix}\mathbf{P}_n = \frac{1}{\sqrt{2}}\begin{bmatrix} \xi_1 & \ldots & \xi_1 & \xi_1 & \ldots & \xi_n \\ \eta_1 & \ldots & \eta_n & -\eta_1 & \ldots & -\eta_n \end{bmatrix},$$

the vectors $\mathbf{x}_j^{\pm}$, $j = 1, \ldots, n$, are the corresponding $\mathbf{K}$-orthonormal right eigenvectors of $\mathbf{H}$. By (3.20), $\mathbf{y}_j^{\pm} := \begin{bmatrix} \pm\eta_j \\ \xi_j \end{bmatrix}$ are the corresponding $\mathbf{M}$-orthonormal left eigenvectors of $\mathbf{H}$. Note that the reason for using the same notation for ξ_j and η_j here as in Proposition 3.1 is that they are indeed the right and left eigenvectors of MK corresponding to the eigenvalue λ_j^2 as described in Proposition 3.1. This can be easily verified by using (2.1) and the SVD of B.

The following convergence results can be deduced from Theorem 3.2.

THEOREM 3.10. *Let γ_j, $\tilde{\gamma}_j$, π_j, $\tilde{\pi}_j$, $\pi_{j,k}$, $\tilde{\pi}_{j,k}$, δ_j, and $\tilde{\delta}_j$ be defined as in Theorem 3.2. Then, for $j = 1, \ldots, k$,*

$$0 \le \lambda_j - \sigma_j = (-\sigma_j) - (-\lambda_j) \le \frac{\lambda_1^2 - \lambda_n^2}{\lambda_j + \sigma_j}\left(\frac{\pi_{j,k}\tan\theta_K(y_1, \xi_j)}{C_{k-j}(1 + 2\gamma_j)}\right)^2,$$

$$0 \le \sigma_j - \lambda_{n-k+j} = (-\lambda_{n-k+j}) - (-\sigma_j) \le \frac{\lambda_1^2 - \lambda_n^2}{\lambda_{n-k+j} + \sigma_j}\left(\frac{\tilde{\pi}_{j,k}\tan\theta_K(y_1, \xi_{n-k+j})}{C_{j-1}(1 + 2\tilde{\gamma}_j)}\right)^2,$$

and for the Ritz vectors one has the bounds,

$$\begin{aligned} &\sin\theta_{\mathbf{K}}\left(\mathbf{v}_j^{\pm}, , \mathbf{x}_j^{\pm}\right) = \sin\theta_{\mathbf{M}}\left(\mathbf{u}_j^{\pm}, \mathbf{y}_j^{\pm}\right) \\ &\quad \le \frac{1}{\cos\varrho_j}\sqrt{\frac{\pi_j^2(1 + (\alpha_k\beta_k)^2/\delta_j^2)}{C_{k-j}^2(1 + 2\gamma_j)}\sin^2\theta_K(y_1, \xi_j) - \sin^2\varrho_j} \\ &\sin\theta_{\mathbf{K}}\left(\mathbf{v}_j^{\pm}, \mathbf{x}_{n-k+j}^{\pm}\right) = \sin\theta_{\mathbf{M}}\left(\mathbf{u}_j^{\pm}, \mathbf{y}_{n-k+j}^{\pm}\right) \\ &\quad \le \sqrt{\sin^2\tilde{\varrho}_j + \cos^2\tilde{\varrho}_j\frac{\tilde{\pi}_j^2(1 + (\alpha_k\beta_k)^2/\tilde{\delta}_j^2)}{C_{j-1}^2(1 + 2\tilde{\gamma}_j)}\sin^2\theta_K(y_1, \xi_{n-k+j})}, \end{aligned}$$

where

$$\varrho_j = \arccos \frac{2\sigma_j}{\lambda_j + \sigma_j}, \qquad \tilde{\varrho}_j = \arccos \frac{\sigma_j + \lambda_{n-k+j}}{2\sigma_j}.$$

Proof. The first two bounds are obtained easily from (3.4) and (3.5). For the last two relations, the equalities are trivial. So we only need to prove the upper bounds.

Following (3.10) and the fact that $\xi_j^T K Y_k \psi_j$ and $\eta_j^T M X_k \phi_j$ have the same sign, which is a consequence of (3.9),

$$\begin{aligned}\cos\theta_{\mathbf{K}}\left(\mathbf{v}_j^{\pm}, \mathbf{x}_j^{\pm}\right) &= \frac{1}{2}|\xi_j^T K Y_k \psi_j + \eta_j^T M X_k \phi_j| \\ &= \frac{1}{2}(\cos\theta_K(Y_k\psi_j, \xi_j) + \cos\theta_M(X_k\phi_j, \eta_j)) = \frac{\lambda_j + \sigma_j}{2\sigma_j}\cos\theta_K(Y_k\psi_j, \xi_j).\end{aligned}$$

Since $0 \le 2\sigma_j/(\lambda_j + \sigma_j) \le 1$, $\cos\varrho_j = \frac{2\sigma_j}{\lambda_j+\sigma_j}$ is well defined. Then, from

$$\cos\theta_K(Y_k\psi_j, \xi_j) = \cos\varrho_j \cos\theta_{\mathbf{K}}\left(\mathbf{v}_j^{\pm}, \mathbf{x}_j^{\pm}\right),$$

one has

$$\sin^2\theta_K(Y_k\psi_j, \xi_j) = 1 - \cos^2\varrho_j \cos^2\theta_{\mathbf{K}}\left(\mathbf{v}_j^{\pm}, \mathbf{x}_j^{\pm}\right) = \sin^2\varrho_j + \cos^2\varrho_j \sin^2\theta_{\mathbf{K}}\left(\mathbf{v}_j^{\pm}, \mathbf{x}_j^{\pm}\right).$$

Hence,

$$\sin\theta_{\mathbf{K}}\left(\mathbf{v}_j^{\pm}, \mathbf{x}_j^{\pm}\right) = \frac{1}{\cos\varrho_j}\sqrt{\sin^2\theta_K(Y_k\psi_j, \xi_j) - \sin^2\varrho_j}\,.$$

The bound for $\sin\theta_{\mathbf{K}}\left(\mathbf{v}_j^{\pm}, \mathbf{x}_j^{\pm}\right)$ then follows from (3.6). The last bound can be proved in the same way by using the relation (3.11). □

REMARK 3.11. With the factorizations in (2.3), it is straightforward to show that (3.19) is equivalent to

$$\begin{bmatrix} 0 & (R^T L)^T \\ R^T L & 0 \end{bmatrix} (\mathbf{Z}_k \tilde{\mathbf{P}}_k) = (\mathbf{Z}_k \tilde{\mathbf{P}}_k)(\tilde{\mathbf{P}}_k^T \mathbf{B}_k \tilde{\mathbf{P}}_k) + \beta_k \begin{bmatrix} v_{k+1} \\ 0 \end{bmatrix} e_{2k}^T, \quad \mathbf{Z}_k = \begin{bmatrix} V_k & 0 \\ 0 & U_k \end{bmatrix}$$

with $V_k = L^T Y_k$, $U_k = R^T X_k$, $v_{k+1} = L^T y_{k+1}$, and $\mathbf{Z}_k^T \mathbf{Z}_k = I_{2k}$. This is an identity resulting in the standard symmetric Lanczos algorithm with the initial vector $\begin{bmatrix} v_1^T & 0 \end{bmatrix}^T$, $v_1 = L^T y_1$. So we can establish the following convergence results directly: for $j = 1, \ldots, k$,

$$\begin{aligned}0 \le \lambda_j - \sigma_j = (-\sigma_j) - (-\lambda_j) &\le 2\lambda_1 \left(\frac{\hat{\pi}_{j,k}\tan\theta_K(y_1, \xi_j)}{C_{2k-j}(1 + 2\hat{\gamma}_j)}\right)^2 \\ \sin\theta_{\mathbf{K}}\left(\mathbf{v}_j^{\pm}, \mathbf{x}_j^{\pm}\right) &\le \frac{\hat{\pi}_j\sqrt{1 + (\alpha_k\beta_k)^2/\hat{\delta}_j^2}}{C_{2k-j}(1 + \hat{\gamma}_j)}\sin\theta_K(y_1, \xi_j),\end{aligned}$$

where

$$\hat{\gamma}_j = \frac{\lambda_j - \lambda_{j+1}}{\lambda_{j+1} + \lambda_1}, \qquad \hat{\pi}_{1,k} = \hat{\pi}_1 = 1,$$

$$\hat{\pi}_{j,k} = \prod_{i=1}^{j-1} \frac{\sigma_i + \lambda_1}{\sigma_i - \lambda_j}, \qquad \hat{\pi}_j = \prod_{i=1}^{j-1} \frac{\lambda_i + \lambda_1}{\lambda_i - \lambda_j}, \qquad \hat{\delta}_j = \min_{i \ne j} |\lambda_j - \sigma_i|.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

However, it seems nontrivial to derive a bound for $\sigma_j - \lambda_{n-k+j}$ since the small positive eigenvalues of $\mathbf{H}$ are the interior eigenvalues.

In [17], another type of Lanczos algorithms was proposed for solving the eigenvalue problem of $\mathbf{H}$. The algorithms are based on the factorizations

$$KU = VT, \qquad MV = UD, \qquad U^T V = I_n$$

with the assumption that $M > 0$, where T is symmetric tridiagonal and D is diagonal, so that

$$\mathbf{H}\begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} = \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}\begin{bmatrix} 0 & D \\ T & 0 \end{bmatrix}.$$

The first Lanczos-type algorithm in [17] computes the columns of U, V and the entries of D and T by enforcing the columns of V to be unit vectors. By running k iterations with the first column of V as an initial vector, the leading principal $k \times k$ submatrices D_k and T_k of D and T, respectively, are computed. Then the eigenvalues of $\begin{bmatrix} 0 & D_k \\ T_k & 0 \end{bmatrix}$ are used to approximate the eigenvalues of $\mathbf{H}$. This algorithm works even when K is indefinite. On the other hand, when $K > 0$, Algorithms 1 and 2 exploit the symmetry of the problem and treat K and M equally, which seem more natural.

4. Numerical examples.

In this section, three examples are presented to illustrate our algorithms. All the numerical results are computed by using Matlab 8.4 (R2014b) on a laptop with an Intel Core i5-4590M @ 3.3GHz CPU and 4GB memory.

Example 1. In this example, we investigate the singular values of the following four matrices

$$\begin{bmatrix} \tilde{B}_k \\ \tilde{\beta}_k e_k^T \end{bmatrix}, \qquad \tilde{B}_k, \qquad \begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}, \qquad B_k.$$

The latter two blocks are generated by Algorithm 1 (wGKL$_u$) with an initial vector y_1 satisfying $\|y_1\|_K = 1$, which is a normalized random vector generated by the Matlab command *randn*. The former two blocks are generated by Algorithm 2 (wGKL$_l$) with the initial vector $\tilde{x}_1 = y_1/\|Ky_1\|_M$ with the same y_1 used in Algorithm 1. The singular values of all four matrices can be used for eigenvalue approximations of the matrices MK and $\mathbf{H}$. We test, which one can provide the best approximations.

The tested positive definite matrices K and M of order $n = 1862$ are from a problem in [17] related to the sodium dimer Na2. Only the largest and the smallest eigenvalues of MK are computed. Assuming σ_j is the jth singular value of each of the above four matrices, we report the relative errors for the largest Ritz value σ_1^2 of MK: $e(\sigma_1^2) := \frac{|\lambda_1^2 - \sigma_1^2|}{\lambda_1^2}$, and the smallest Ritz value σ_k^2 of MK: $e(\sigma_k^2) := \frac{|\lambda_n^2 - \sigma_k^2|}{\lambda_n^2}$, respectively. The “exact" eigenvalues $\lambda_1^2 \approx 1.25 \times 10^2$ and $\lambda_n^2 \approx 0.41$ of MK are computed by using the MATLAB command *eig*.

We set $k = 1, \ldots, 15$ for the largest eigenvalue case and $k = 1, \ldots, 150$ for the smallest eigenvalue case. The numerical results are reported in Figure 4.1. From the figures we can see, as discussed in the last part of Section 3.1, that the square of the largest singular value of $\begin{bmatrix} \tilde{B}_k \\ \tilde{\beta}_k e_k^T \end{bmatrix}$ is closer to the largest eigenvalue of MK than that of $\tilde{B}_k$. Thus, they are also closer than those of $\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}$ and B_k. The square of the smallest singular value of B_k is the closest to the smallest eigenvalue of MK among those of the above four matrices. We can also see from the figures, because of equation (3.17), that the extreme singular values of $\tilde{B}_k$ and $\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}$ coincide.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

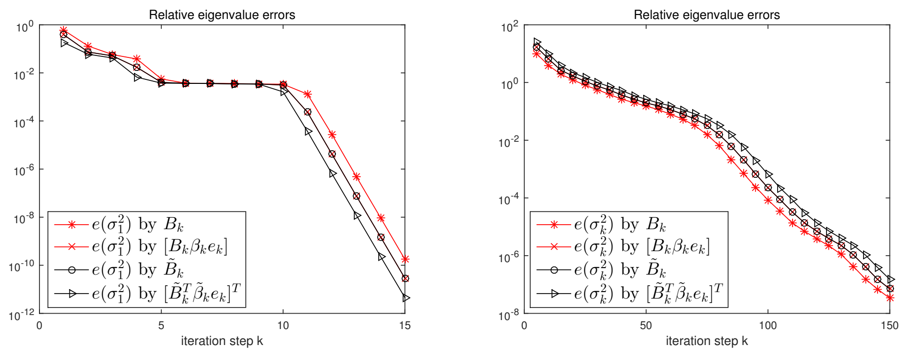

FIG. 4.1. *Relative errors of the extreme eigenvalues of MK in Example 1.*

We also used the same matrices M and K to verify the residual formulas in (3.3) and (3.13) for the extreme eigenvalues. The actual residuals

$$r_{R1,j} := \|(MK - \sigma_j^2 I)Y_k\psi_j\|_K, \qquad r_{L1,j} := \|(KM - \sigma_j^2 I)X_k\phi_j\|_M,$$
$$r_{R2,j} := \|(MK - \rho_j^2 I)Y_{k+1}\omega_j\|_K, \qquad r_{L2,j} := \|(KM - \rho_j^2 I)X_k\zeta_j\|_M,$$

and the corresponding quantities

$$q_{R1,j} := \alpha_k\beta_k|\psi_{jk}|, \qquad q_{L1,j} := \beta_k|\phi_{jk}|\sqrt{\beta_k^2 + \alpha_{k+1}^2},$$
$$q_{R2,j} := \alpha_{k+1}|\omega_{j,k+1}|\sqrt{\alpha_{k+1}^2 + \beta_{k+1}^2}, \qquad q_{L2,j} := \alpha_{k+1}\beta_k|\zeta_{jk}|,$$

for $j = 1, k$ with various values of k, are depicted Figure 4.2. The results show that the quantities are close to the actual residuals.

Example 2. In this example, we compare Algorithm 1 (wGKL$_u$) with the weighted Lanczos algorithm for the eigenvalues of MK. The weighted Lanczos algorithm is based on the relations given in (2.5). The singular values of both B_k and $\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}$ generated by wGKL$_u$ are used to approximate the eigenvalues of MK. The numerical results computed by wGKL$_u$ are labeled with Alg-1 and those computed by the weighted Lanczos algorithm with Alg-WL.

We performed a comparison with four pairs of matrices K and M:

1. K and M are of order $n = 1000$ with $K = QDQ^T$ and $M = Q\widehat{D}Q^T$, where Q is orthogonal generated from the QR factorization of a random matrix, $D = \mathrm{diag}(d_1, \dots, d_n)$, with $d_i = 10^{i-7}$ for $i = 1, \dots, 6$, and the rest of the diagonal elements generated by the Matlab command *rand*. $\widehat{D}$ is another diagonal matrix formed by reversing the order of the diagonal elements of D. The extreme eigenvalues of MK are $\lambda_1 \approx 0.98$ and $\lambda_n \approx 5.14 \times 10^{-7}$.
2. K and M are of order $n = 2000$ with K constructed in exactly the same way as before and $M = I_n$. The extreme eigenvalues of MK are $\lambda_1 \approx 0.9999$ and $\lambda_n = 10^{-6}$.
3. $K = I_n$ and M is the matrix K in the matrix pair of item 2. Note that for such a pair, since $K = I$, the weighted Lanczos algorithm is just the standard Lanczos algorithm.
4. K and M are of order $n = 1000$ with $K = QDQ^T$ and $M = \widehat{Q}\widehat{D}\widehat{Q}^T$, where both Q and $\widehat{Q}$ are orthogonal, Q is generated from the QR factorization of a random

ETNA
Kent State University and
Johann Radon Institute (RICAM)

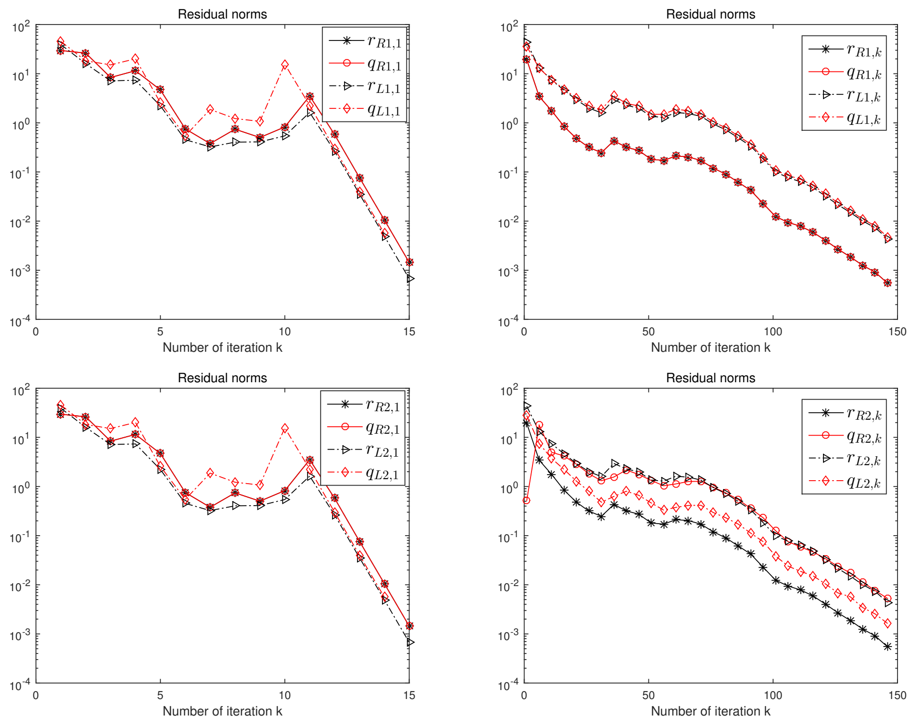

FIG. 4.2. *Residual norms of the extreme eigenvalues of MK in Example 1.*

matrix, $\widehat{Q}$ is generated from the QR factorization of $Q*(I+10^{-10}E)$ with E being a random matrix, $D = \text{diag}(d_1, \dots, d_n)$, and $\hat{D} = \text{diag}(\hat{d}_1, \dots, \hat{d}_n)$ with all diagonal elements generated by the Matlab command *rand* but $d_{n/2} = 10^{-7}$, $d_{n/2+1} = 10^{-8}$, $\hat{d}_1 = 10^{-7}$, and $\hat{d}_n = 10^{-8}$. The "exact" extreme eigenvalues of MK are $\lambda_1 \approx 0.93$ and $\lambda_n \approx 1.93 \times 10^{-9}$ computed with the Matlab command *eig*.

For each pair we run k steps of both algorithms to compute the extreme Ritz values. A scaled randomly generated vector y_1 satisfying $y_1^T K y_1 = 1$ serves as the initial vector for both of the algorithms. The extreme Ritz values computed by wGKL$_u$ are denoted by σ_1^2 and σ_k^2, where σ_1 and σ_k are the extreme singular values of either B_k or $\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}$, and those by the weighted Lanczos algorithm are denoted by ν_1 and ν_k. We measure the accuracy by the absolute errors $e(\hat{\lambda}_1) = |\hat{\lambda}_1 - \lambda_1|$ and $e(\hat{\lambda}_k) = |\hat{\lambda}_k - \lambda_n|$, where $\hat{\lambda}_1$ is either σ_1^2 or ν_1 and $\hat{\lambda}_k$ is either σ_k^2 or ν_k. The Figures 4.3–4.6 display the absolute errors for the pairs in the items 1–4 for various values of k.

The numerical results show that both algorithms behave essentially the same in practice. The only place where wGKL$_u$ does slightly better is in approximating the smallest eigenvalue of MK from the pair in item 4. wGKL$_u$ converges eventually while the weighted Lanczos algorithm stagnates. In all the cases, for the largest eigenvalue of MK, the largest singular value of $\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}$ gives a slightly better approximation than the rest. For the smallest eigenvalue of MK, the smallest singular value of $\begin{bmatrix} B_k & \beta_k e_k \end{bmatrix}$ gives the worst approximation.

We ran the tests with many other pairs of M and K. No significant difference between the two algorithms was observed.

Example 3. In this example, we test Algorithm 3 (wGKL$_u$-LREP) for solving the eigenvalue problem of a matrix $\mathbf{H}$ given in [17]. The matrices K and M in $\mathbf{H}$ are extracted

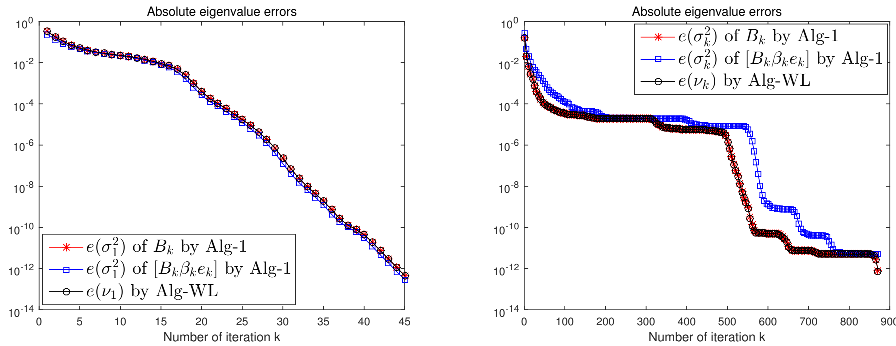

FIG. 4.3. *Absolute errors of the extreme eigenvalues of MK for pair 1 in Example 2.*

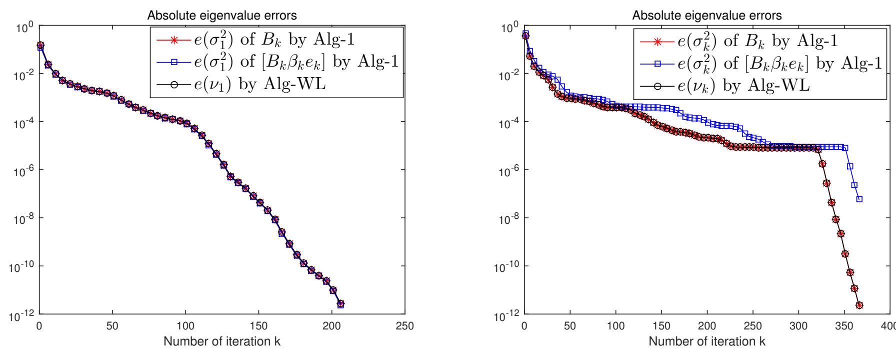

FIG. 4.4. *Absolute errors of the extreme eigenvalues of MK for pair 2 in Example 2.*

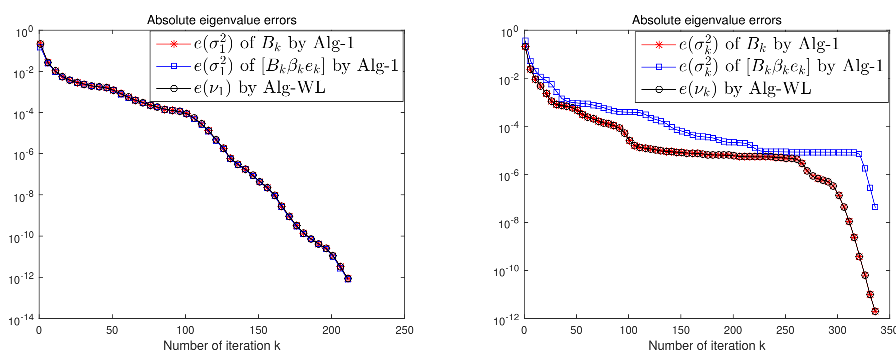

FIG. 4.5. *Absolute errors of the extreme eigenvalues of MK for pair 3 in Example 2.*

from the University of Florida sparse matrix collection [6]: K is $fv1$ with $n = 9604$, and M is the $n \times n$ leading principal submatrix of $finan512$. Both K and M are symmetric positive definite. The two smallest eigenvalues of $\mathbf{H}$ are approximately 1.15, 1.17, and the two largest ones are approximately 9.80, 9.75.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

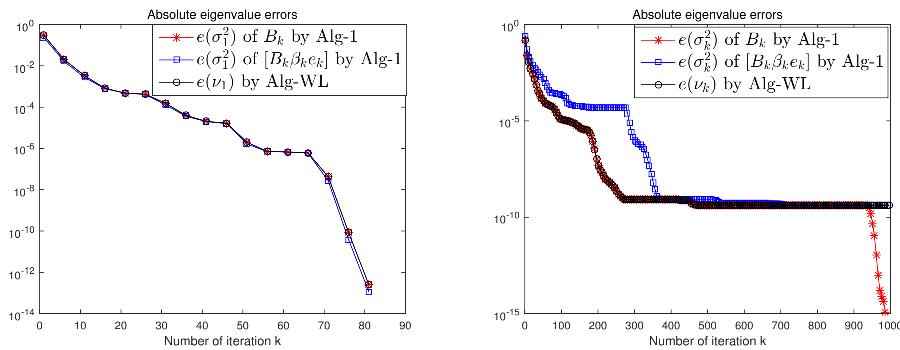

FIG. 4.6. *Absolute errors of the extreme eigenvalues of MK for pair 4 in Example 2.*

The initial vector y_1 for wGKL$_u$-LREP is randomly selected satisfying $\|y_1\|_K = 1$. The numerical results are labeled with Alg-3. For comparison, we also test the first algorithm presented in [17] with the initial vector $y_1/\|y_1\|$. The numerical results are labeled with Alg-TL. We also run the weighted Lanczos algorithm based on the relation (3.19) with $\mathbf{H}$ being treated as a full matrix and $\mathbf{X}_k$ being a $\mathbf{K}$-orthonormal matrix. The initial vector is $\begin{bmatrix} y_1^T & 0 \end{bmatrix}^T$. The numerical results are labeled with Alg-Full. We only compute the two largest and two smallest positive eigenvalues of $\mathbf{H}$. For the two largest positive eigenvalues we run $m = 50$ iterations with Alg-3 and Alg-TL and $2m = 100$ iterations with Alg-Full. For the two smallest positive eigenvalues we run $m = 200$ for the former two algorithms and $2m = 400$ iterations with the latter. (Recall that two iterations of Alg-Full are equivalent to one iteration of Alg-3 and Alg-TL.)

We report the relative eigenvalue error and the magnitude of the normalized residuals in the 1-norm for each of the 4 Ritz pair $(\sigma_j, \mathbf{v}_j^+)$:

$$e(\sigma_j) := \begin{cases} \frac{|\lambda_j - \sigma_j|}{\lambda_j}, & j = 1, 2, \\ \frac{|\lambda_{n+j-k} - \sigma_j|}{\lambda_{n+j-k}}, & j = k-1, k, \end{cases}$$

$$r(\sigma_j) := \frac{\|\mathbf{H}\mathbf{v}_j^+ - \sigma_j \mathbf{v}_j^+\|_1}{(\|\mathbf{H}\|_1 + \sigma_j)\|\mathbf{v}_j^+\|_1}, \qquad j = 1, 2, k-1, k,$$

for each of the iterations $k = 1, 2, \ldots, m$ of Alg 3 and Alg-TL (and k is supposed to be $2k$ for Alg-Full). The “exact" eigenvalues λ_j are computed by the MATLAB command *eig*.

The testing results associated with the two smallest positive eigenvalues are shown in Figure 4.7, and the results associated with the two largest eigenvalues are shown in Figure 4.8. For the two smallest positive eigenvalues, Alg-3 runs for about 4.515 seconds, Alg-Full about 4.556 seconds, and Alg-TL about 15.314 seconds. For the two largest eigenvalues the runtime is about 0.313, 0.344, 0.469 seconds, respectively. Alg-TL needs to compute the eigenvalues of $\begin{bmatrix} 0 & D_k \\ T_k & 0 \end{bmatrix}$, which is treated as a general nonsymmetric matrix. This is the part that slows down Alg-TL. On the other hand, Alg-Full gives less accurate numerical results than the other two algorithms. This example shows that Alg-3 works well. It takes less time than Alg-TL to obtain almost the same numerical results.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

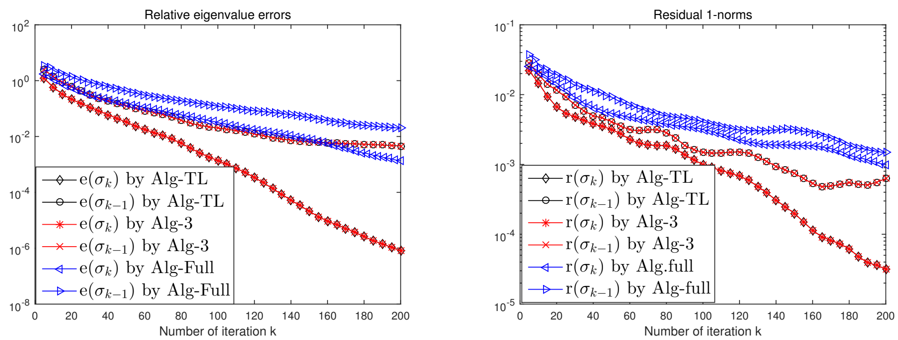

FIG. 4.7. *Errors and residuals of the two smallest positive eigenvalues in Example 3.*

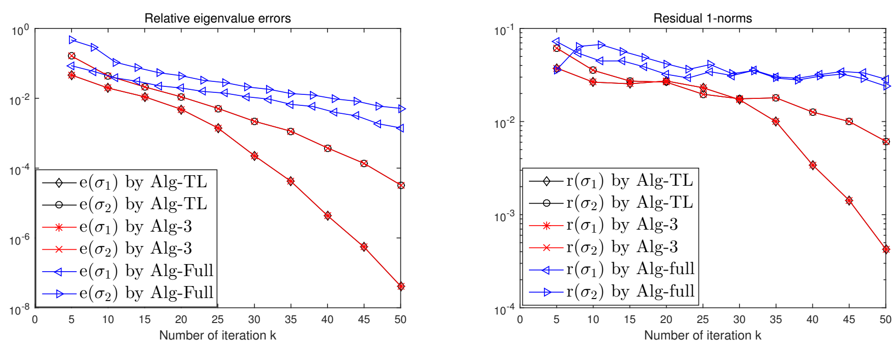

FIG. 4.8. *Errors and residuals of the two largest eigenvalues in Example 3.*

5. Connection with weighted conjugate gradient methods. Consider the system of linear equations

$$Mz = b, \qquad M > 0. \tag{5.1}$$

Let z_0 be an initial guess of the solution $z_e = M^{-1}b$ and $r_0 = b - Mz_0 = M(z_e - z_0)$ be the corresponding residual. Assume that X_k, Y_k, and B_k are computed by wGKL$_u$ with M and another matrix $K > 0$ and $y_1 = r_0/\|r_0\|_K$. Then they satisfy (2.4) and (2.6).

We approximate the solution z_e by a vector $z_k \in z_0 + K\mathcal{K}_k(MK, y_1)$ for some $k \in \{1, \ldots, n\}$. From (2.6), we may express

$$z_k = z_0 + X_k w_k$$

for some $w_k \in \mathbb{R}^k$. We take the approximation z_k (or equivalently w_k) as the solution of the minimization problem

$$\min_{w_k} J(w_k), \qquad J(w_k) = \varepsilon_k^T M \varepsilon_k, \qquad \varepsilon_k = z_e - z_k = \varepsilon_0 - X_k w_k.$$

Since

$$\begin{aligned} J(w_k) &= w_k^T X_k^T M X_k w_k - 2w_k^T X_k^T M\varepsilon_0 + \varepsilon_0^T M \varepsilon_0 \\ &= w_k^T X_k^T M X_k w_k - 2w_k^T X_k^T r_0 + \varepsilon_0^T M \varepsilon_0, \end{aligned}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

the functional $J(w_k)$ is minimized when w_k satisfies

$$X_k^T M X_k w_k = X_k^T r_0.$$

Using $r_0 = \|r_0\|_K y_1$, $X_k^T M X_k = I_k$, $Y_k^T K Y_k = I_k$, and the first relation of (2.4), one has

$$X_k^T r_0 = \|r_0\|_K (K Y_k B_k^{-1})^T y_1 = \|r_0\|_K B_k^{-T} Y_k^T K y_1 = \|r_0\|_K B_k^{-T} e_1.$$

Hence the minimizer is

$$z_k = z_0 + X_k w_k, \quad \text{with} \quad w_k = \|r_0\|_K B_k^{-T} e_1.$$

The vector w_k can be computed in an iterative way along with the iterations of wGKL$_u$. Note that

$$B_k^T = \begin{bmatrix} \alpha_1 & & & \\ \beta_1 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_k \end{bmatrix} = \begin{bmatrix} B_{k-1}^T & 0 \\ \beta_{k-1} e_{k-1}^T & \alpha_k \end{bmatrix}.$$

So

$$B_k^{-T} = \begin{bmatrix} B_{k-1}^{-T} & 0 \\ -\frac{\beta_{k-1}}{\alpha_k} e_{k-1}^T B_{k-1}^{-T} & \alpha_k^{-1} \end{bmatrix},$$

and by denoting $w_k = \begin{bmatrix} \varphi_1 & \dots & \varphi_k \end{bmatrix}^T$, one has

$$w_k = \|r_0\|_K B_k^{-T} e_1 = \|r_0\|_K \begin{bmatrix} B_{k-1}^{-T} e_1 \\ -\frac{\beta_{k-1}}{\alpha_k} e_{k-1}^T B_{k-1}^{-T} e_1 \end{bmatrix} = \begin{bmatrix} w_{k-1} \\ \varphi_k \end{bmatrix},$$

where φ_k follows the iteration

$$\varphi_k = -\frac{\beta_{k-1}}{\alpha_k} e_{k-1}^T w_{k-1} = -\frac{\beta_{k-1}}{\alpha_k} \varphi_{k-1}, \qquad k \geq 1, \quad \beta_0 = 1, \quad \varphi_0 = -\|r_0\|_K. \tag{5.2}$$

Therefore,

$$z_k = z_0 + X_k w_k = z_{k-1} + \varphi_k x_k, \qquad k \geq 1,$$

and using $B_k^T w_k = \|r_0\|_K e_1$ and the second relation in (2.4), the corresponding residual is

$$\begin{aligned} r_k &= b - M z_k = r_0 - M X_k w_k = r_0 - (Y_k B_k^T + \beta_k y_{k+1} e_k^T) w_k \\ &= r_0 - \|r_0\|_K Y_k e_1 - \beta_k \varphi_k y_{k+1} = -\beta_k \varphi_k y_{k+1}, \quad k \geq 0. \end{aligned}$$

Hence, we have the following algorithm for solving (5.1).

ALGORITHM 4 (wGKL$_u$-Lin).

Choose z_0 and compute $r_0 = b - M z_0$, $\varphi_0 = -\|r_0\|_K$, and $y_1 = r_0/\|r_0\|_K$. Set $\beta_0 = 1$, $x_0 = 0$. Compute $g_1 = K y_1$.

For $j = 1, 2, \cdots$

$s_j = g_j/\beta_{j-1} - \beta_{j-1} x_{j-1}$

$f_j = M s_j$

$\alpha_j = (s_j^T f_j)^{\frac{1}{2}}$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

$x_j = s_j/\alpha_j$
$\varphi_j = -\beta_{j-1}\varphi_{j-1}/\alpha_j$
$z_j = z_{j-1} + \varphi_j x_j$
$t_{j+1} = f_j/\alpha_j - \alpha_j y_j$
$g_{j+1} = K t_{j+1}$
$\beta_j = (t_{j+1}^T g_{j+1})^{\frac{1}{2}}$
$y_{j+1} = t_{j+1}/\beta_j$
$r_j = -\varphi_j t_{j+1}$
End

We show that Algorithm 4 is equivalent to a weighted conjugate gradient (CG) method. By introducing the vectors $p_{k-1} = \alpha_k^2 \varphi_k x_k$, for $k \geq 1$, with

$$p_0 = \alpha_1^2 \varphi_1 x_1 = \alpha_1 \varphi_1 K y_1 = \|r_0\|_K K y_1 = K r_0,$$

one has

$$p_{k-1}^T M p_{k-1} = \alpha_k^4 \varphi_k^2 x_k^T M x_k = \alpha_k^4 \varphi_k^2.$$

Since $r_k = -\beta_k \varphi_k y_{k+1}$, using (5.2), one has

$$r_k^T K r_k = \beta_k^2 \varphi_k^2 y_{k+1}^T K y_{k+1} = \beta_k^2 \varphi_k^2 = \alpha_{k+1}^2 \varphi_{k+1}^2.$$

We then have

$$\alpha_{k+1}^2 = \frac{\alpha_{k+1}^4 \varphi_{k+1}^2}{\alpha_{k+1}^2 \varphi_{k+1}^2} = \frac{p_k^T M p_k}{r_k^T K r_k}.$$

Now,

$$\begin{aligned} z_k &= z_{k-1} + \varphi_k x_k = z_{k-1} + \alpha_k^{-2} p_{k-1} = z_{k-1} + \gamma_{k-1} p_{k-1}, \\ \gamma_{k-1} &= \alpha_k^{-2} = \frac{r_{k-1}^T K r_{k-1}}{p_{k-1}^T M p_{k-1}}, \end{aligned}$$

and

$$r_k = b - M z_k = r_{k-1} - \gamma_{k-1} M p_{k-1}.$$

By multiplying the equation

$$K y_{k+1} = \beta_k x_k + \alpha_{k+1} x_{k+1},$$

with $\alpha_{k+1}\varphi_{k+1}$ and using (5.2), one has

$$\begin{aligned} p_k &= \alpha_{k+1}^2 \varphi_{k+1} x_{k+1} = -\alpha_{k+1} \beta_k \varphi_{k+1} x_k + \alpha_{k+1} \varphi_{k+1} K y_{k+1} \\ &= \beta_k^2 \varphi_k x_k - \beta_k \varphi_k K y_{k+1} = \frac{\beta_k^2}{\alpha_k^2} p_{k-1} + K r_k. \end{aligned}$$

Since

$$\vartheta_{k-1} := \frac{\beta_k^2}{\alpha_k^2} = \frac{\beta_k^2 \varphi_k^2}{\alpha_k^2 \varphi_k^2} = \frac{r_k^T K r_k}{r_{k-1}^T K r_{k-1}},$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

we have

$$p_k = Kr_k + \vartheta_{k-1}p_{k-1}, \quad \vartheta_{k-1} = \frac{r_k^T K r_k}{r_{k-1}^T K r_{k-1}}.$$

As a consequence, by using r_k and p_k instead of y_k and x_k, we have the following simplified algorithm.

ALGORITHM 5.
Choose z_0 and compute $r_0 = b - Mz_0$ and $p_0 = Kr_0$.
For $j = 0, 1, 2, \cdots$
  $\gamma_j = r_j^T K r_j / p_j^T M p_j$
  $z_{j+1} = z_j + \gamma_j p_j$
  $r_{j+1} = r_j - \gamma_j M p_j$
  $\vartheta_j = r_{j+1}^T K r_{j+1} / r_j^T K r_j$
  $p_{j+1} = K r_{j+1} + \vartheta_j p_j$
End

Algorithm 5 is a weighted CG algorithm, which is alike the standard CG but with the residuals r_j being forced to be K-orthogonal. It is just the preconditioned CG (PCG) if K is a matrix inverse. In particular, it is the standard CG if $K = I$. On the other hand, based on PCG theory, the vector sequences $\{r_j\}$ and $\{p_j\}$ produced by Algorithm 5 are K and M-orthogonal, respectively. By normalizing the vectors, we obtain $\{y_j\}$ and $\{x_j\}$, and by replacing $\{r_j\}$ and $\{p_j\}$ in Algorithm 5 with $\{y_i\}$ and $\{x_j\}$, we recover Algorithm 4. Therefore, Algorithms 4 and 5 are equivalent.

This equivalence provides another way to connect the PCG to Krylov subspace methods. Commonly, a connection is made for PCG and the preconditioned Lanczos algorithm [15], where the Cholesky factorization of the computed symmetric tridiagonal matrix is involved. Since Algorithm 4 computes the Cholesky factor directly (even when $K = I$), the new connection is more direct and compact.

Finally, we point out that wGKL$_l$ can be employed to solve (5.1) as well.

6. Conclusions. We have proposed two weighted Golub-Kahan-Lanczos bidiagonalization algorithms wGKL$_u$ and wGKL$_l$ associated with two symmetric positive definite matrices K and M. We have shown that the algorithms can be implemented naturally to solve the large-scale eigenvalue problems of MK and the matrix $\mathbf{H} = \begin{bmatrix} 0 & M \\ K & 0 \end{bmatrix}$. For these eigenvalue solvers, convergence results have been established. Besides the eigenproblems, the algorithms can also be implemented to solve linear equations with a positive definite coefficient matrix, yielding a method that is equivalent to PCG. Several numerical examples have been given to illustrate the effectiveness of our algorithms.

The proposed algorithms are still in basic form. In order to develop more practical algorithms, additional techniques need to be employed. There are well-developed techniques for Krylov subspace methods, many of which can be incorporated into the proposed algorithms. For instance, in order to compute the smallest eigenvalues of $\mathbf{H}$, one may apply the wGKL algorithms to the pair (K^{-1}, M^{-1}), following the shift-and-invert idea. There are also some open questions concerning the proposed algorithms. For instance, it is not clear whether the use of the weighted norm will affect the numerical efficiency and stability of the algorithms. All these require further investigations.

Acknowledgment. The authors thank Prof. Gang Wu for helpful discussions on the numerical examples and Dr. Zhongming Teng for providing the codes for Alg-TL and the data of the matrix Na2. We also thank the anonymous referees for their constructive comments that helped to improve the quality of the paper. The first author was supported in part by the China Scholarship Council (CSC) during a visit at the Department of Mathematics, University of Kansas, and was supported by the National Natural Science Foundation of China (No. 11701225), the Fundamental Research Funds for the Central Universities (No. JUSRP11719), and the Natural Science Foundation of Jiangsu Province (No. BK20170173) and was partially supported by the National Natural Science Foundation of China (No. 11471122).

REFERENCES

[1] M. ARIOLI, *Generalized Golub-Kahan bidiagonalization and stopping criteria*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 571–592.

[2] Z. BAI AND R.-C. LI, *Minimization principles for the linear response eigenvalue problem I: Theory*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 1075–1100.

[3] ———, *Minimization principles for the linear response eigenvalue problem II: Computation*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 392–416.

[4] S. J. BENBOW, *Solving generalized least-squares problems with LSQR*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 166–177.

[5] M. E. CASIDA, *Time-dependent density-functional response theory for molecules*, in Recent Advances in Density Functional Methods, D. P. Chong, ed., World Scientific, Singapore, 1995, pp. 155–192.

[6] T. A. DAVIS AND Y. HU, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), Art. 1, (25 pages).

[7] G. GOLUB AND W. KAHAN, *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2 (1965), pp. 205–224.

[8] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.

[9] M. J. LUCERO, A. M. N. NIKLASSON, S. TRETIAK, AND M. CHALLACOMBE, *Molecular-orbitalfree algorithm for excited states in time-dependent perturbation theory*, J. Chem. Phys., 14 (2008), Art. 064114, (8 pages).

[10] G. ONIDA, L. REINING, AND A. RUBIO, *Electronic excitations: density-functional versus many-body green's function approaches*, Rev. Modern Phys., 74 (2002), pp. 601–659.

[11] C. C. PAIGE, *Bidiagonalization of matrices and solutions of the linear equations*, SIAM J. Numer. Anal., 11 (1974), pp. 197–209.

[12] C. C. PAIGE AND M. A. SAUNDERS, *LSQR: an algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[13] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998.

[14] D. ROCCA, *Time-dependent Density Functional Perturbation Theory: New Algorithms with Applications to Molecular Spectra*, PhD. Thesis, International School for Advanced Studies, Trieste, Italy, 2007.

[15] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[16] ———, *Numerical Methods for Large Eigenvalue Problems.*, 2nd ed., SIAM, Philadelphia, 2011.

[17] Z.-M. TENG AND R.-C. LI, *Convergence analysis of Lanczos-type methods for the linear response eigenvalue problem*, J. Comput. Appl. Math., 247 (2013), pp. 17–33.