Specialized Spectral Division Algorithms for Generalized Eigenproblems via the Matrix Disk Function[⋆]

Mercedes Marqués[1], Enrique S. Quintana-Ortí[1], and Gregorio Quintana-Ortí[1]

Depto. de Ingeniería y Ciencia de Computadores, Universidad Jaume I, 12.071–Castellón, Spain; {mmarques,quintana,gquintan}@icc.uji.es.

Abstract. We describe two implementations of the inverse-free iteration for the matrix disk function that reduce the computational cost of the traditional algorithm. One of the implementations is mainly composed of efficient BLAS-3 operations, and can be employed for spectral division of large-scale generalized eigenproblems on current computer architectures.

Keywords: Spectral division, generalized eigenproblem, matrix pencil, matrix disk function, orthogonal transformation.

1 Introduction

Consider the matrix pencil $A - \lambda E$, where A, $E \in \mathbb{R}^{n\times n}$, and let $\Lambda(A, E)$ denote the set of generalized eigenvalues of the pencil. In the spectral division problem we are interested in finding a pair of (orthogonal) matrices U, $V \in \mathbb{R}^{n\times n}$ such that

$$U^T AV = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \quad U^T EV = \begin{bmatrix} E_{11} & E_{12} \\ 0 & E_{22} \end{bmatrix}, \tag{1}$$

and $\Lambda(A_{11}, E_{11})$, $\Lambda(A_{22}, E_{22})$ are disjoint sets containing certain parts of $\Lambda(A, E)$. The spectral division has important applications in matrix diagonalization, computation of deflating subspaces, and related problems in control.

There are several methods to compute the decomposition in (1). In this paper we consider a spectral division approach via the matrix disk function, and we propose two variants of Malyshev's iteration [2] for the matrix disk function that significatively reduce its computational cost while maintaining the numerical accuracy. The first variant employs Givens rotations and results in an algorithm composed of BLAS-1 computations. We also propose a block generalization of this scheme that employs efficient (BLAS-3) Householder transformations.

The rest of the paper is structured as follows. In Section 2 we review the inverse-free iteration for the matrix disk function. Next, in Section 3 we expose the two variants of the iterative scheme that reduce the (theoretical) computational cost. We conclude the paper with some final remarks in Section 4.

[⋆] This research was partially supported by the DAAD programme Acciones Integradas HA2005-0081, the CICYT project TIN2005-09037-C02-02 and FEDER, and project No. P1B-2004-6 of the *Fundación Caixa-Castellón/Bancaixa and UJI.*

2 Malyshev's Inverse-Free Iteration for the Disk Function

In [2] Malyshev proposed the following "inverse-free" iteration to obtain the disk function of a matrix pencil $A - \lambda E$:

$$\begin{array}{ll} A_0 \leftarrow A, & A_{k+1} \leftarrow Q_{12}^T A_k, \\ E_0 \leftarrow E, & E_{k+1} \leftarrow Q_{22}^T E_k, \quad k = 0, 1, \ldots, \end{array} \tag{2}$$

where, at each iteration, Q_{12} and Q_{22} are obtained from the QR factorization of the $2n \times n$ matrix M_k

$$M_k := \begin{bmatrix} E_k \\ -A_k \end{bmatrix} = Q_k \bar{R}_k = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} R_k \\ 0 \end{bmatrix}. \tag{3}$$

The QR factorization in (3) is usually computed by applying n Householder reflectors, $H_1,\ H_2, \ldots, H_n$, to M_k, so that $Q_k = H_1 \cdot H_2 \cdots H_n$. Let $I_n/0_n$ denote, respectively, the (square) identity/zero matrix of order n. The appropriate part of Q_k can then be formed by "accumulating" the reflectors in reverse order as

$$H_1 \cdot H_2 \cdots H_n \begin{bmatrix} 0_n \\ I_n \end{bmatrix} = Q_k \begin{bmatrix} 0_n \\ I_n \end{bmatrix} = \begin{bmatrix} Q_{12} \\ Q_{22} \end{bmatrix}.$$

Further practical details on the inverse-free iteration can be found in [1], where the iteration was modified to made it truly inverse-free. A new strategy for subspace extraction was formulated in [3] which provides both U and V in (1) from a single iteration, thus halving the cost of the algorithm proposed in [1].

The computational cost in flops (floating-point arithmetic operations) of the iteration is reported in the column labeled as "Traditional" in Table 1. The computation of the generalized real Schur form via the QZ algorithm roughly requires $81n^3$ flops (plus the cost of the reordering procedure). Thus, in theory, 6 inverse-free iterations are about as expensive as the QZ algorithm. In practice the number is quite higher as the QZ algorithm is composed of fine-grain operations which do not attain the highest execution rate of current computer platforms.

Step	Traditional	Givens-based	Blocked Householder
QR fact.	$3n^3 + n^3/3$	$3n^3$	$3n^3$
Accumulate Q_k	$6n^3$	$3n^3$	$3n^3$
$A_{k+1} \leftarrow Q_{12}^T A_k$	$2n^3$	n^3	n^3
$E_{k+1} \leftarrow Q_{22}^T E_k$	$2n^3$	$2n^3$	$2n^3$
Total	$13n^3 + n^3/3$	$9n^3$	$9n^3$

Table 1. Computational costs (in flops) of the different algorithms for the inverse-free iteration. For the blocked Householder algorithm, the block size b is assumed to satisfy $b \ll n$.

$$M_k = \left[\begin{array}{c} E_k \\ \hline -A_k \end{array}\right] = \left[\begin{array}{ccc} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \hline \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{array}\right] \qquad \left[\begin{array}{ccc} \times & \times & \times \\ 3 & \times & \times \\ 2 & 6 & \times \\ \hline 1 & 5 & 9 \\ 0 & 4 & 8 \\ 0 & 0 & 7 \end{array}\right] \qquad \left[\begin{array}{c} Q_{12} \\ \hline Q_{22} \end{array}\right] = \left[\begin{array}{ccc} \times & 0 & 0 \\ \times & \times & 0 \\ \times & \times & \times \\ \hline \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{array}\right]$$

Fig. 1. Structure of the augmented matrix M_k before the elimination proceeds (left), order in which elements are annihilated (middle), and trailing blocks of the orthogonal matrix resulting from the factorization.

3 Reducing the Cost

Consider the QR factorization $A = U_A R_A$, where U_A is orthogonal and R_A is upper triangular, and the matrix pencil $A_0 - \lambda E_0 = R_A - \lambda(U_A^T E)$, which shares its eigenspectrum with $A - \lambda E$. The key to reducing the cost of the iteration lies in exploiting and preserving the upper triangular structure of the sequence of matrices $\{A_k\}_{k=0,1,\ldots}$ during the iteration. We next explain how to do so by using Givens rotations for a simple example involving a matrix pencil of dimension $n = 3$. During the exposition, we will use $G_l^{i,j}$ to denote a certain (l-th) Givens rotation which, applied to rows $i-1$ and i, serves to annihilate the (i,j) entry of $M_k = \left[E_k^T, \ -A_k^T\right]^T$.

Before the computations in iteration k commence, the augmented matrix M_k presents the structure in Fig. 1 (left). There, the symbol '$\times$' is used to denote a nonzero entry. Although using Givens rotations we can reduce M_k to upper triangular form in many different orders, we are particularly interested in Q_{12}^T being upper triangular, so that $A_{k+1} \leftarrow Q_{12}^T A_k$ remains upper triangular. Therefore, we apply the sequence of Givens rotations $G_1^{4,1}$, $G_2^{3,1}$, $G_3^{2,1}$, $G_4^{5,2}$, $G_5^{4,2}$, $G_6^{3,2}$, $G_7^{6,3}$, $G_8^{5,3}$, and $G_9^{4,3}$, which introduce zeros in M_k in the order specified in Fig. 1 (middle).

In order to construct the appropriate blocks of Q_k, we next proceed to apply the rotations in reverse order to the matrix $[0_n, \ I_n]^T$. The structure of the result,

$$(G_9^{4,3} G_8^{5,3} G_7^{6,3} G_6^{3,2} G_5^{4,2} G_4^{5,2} G_3^{2,1} G_2^{3,1} G_1^{4,1})^T \left[\begin{array}{c} 0_n \\ \hline I_n \end{array}\right] = Q_k \left[\begin{array}{c} 0_n \\ \hline I_n \end{array}\right] = \left[\begin{array}{c} Q_{12} \\ \hline Q_{22} \end{array}\right],$$

is shown in Fig. 1 (right).

The previous procedure yields a reduction of 32% of the computational cost when compared with that of the traditional algorihtm (see the column labeled as "Givens-based" in Table 1). However, even with these savings, we do not expect the Givens-based algorithm to outperform the traditional implementation. This is so because the application of Givens rotations is a Level-1 BLAS operation, while the traditional algorithm employs much faster Level-3 BLAS kernels. Therefore, the effect of the decrease in theoretical cost is hidden by the use of a type of operation which attains much lower performance on current architectures.

We next describe a second algorithm intended for large-scale problems, which is a blocked variant of the Givens-based algorithm. The new algorithm presents a theoretical cost similar to that of the Givens-based algorithm, but performs most of its computations in terms of Level-3 BLAS operations.

Consider the augmented matrix M_k, partitioned into 6×3 blocks of dimension $b \times b$ each, $M_k = \left[\bar{M}_{i,j}\right]$, $i = 1,\dots,6$, $j = 1,2,3$. (For simplicity, we assume n to be an exact multiple of b.) The matrix presents the structure in Fig. 1 (left) where the symbols "$\times$" represent each a block of dimension $b\times b$. Let us now use $U_l^{i,j}$ to denote the (l-th) orthogonal factor from the QR factorization of $\left[\bar{M}_{i-1,j}^T,\ \bar{M}_{i,j}^T\right]^T$. Then, we can compute and apply a sequence of orthogonal matrices $U_1^{4,1}$, $U_2^{3,1}$, $U_3^{2,1}$, $U_4^{5,2}$, $U_5^{4,2}$, $U_6^{3,2}$, $U_7^{6,3}$, $U_8^{5,3}$, and $U_9^{4,3}$, so that zeros are introduced in the blocks of M_k in the order specified in Fig. 1 (middle). Finally, by embedding each $U_l^{i,j}$ into an orthogonal matrix $Q_l^{i,j}$ of the appropriate dimensions, we can accumulate the transposed orthogonal transformations in reverse order as

$$(Q_9^{4,3}Q_8^{5,3}Q_7^{6,3}Q_6^{3,2}Q_5^{4,2}Q_4^{5,2}Q_3^{2,1}Q_2^{3,1}Q_1^{4,1})^T \left[\frac{0_n}{I_n}\right] = Q_k \left[\frac{0_n}{I_n}\right] = \left[\frac{Q_{12}}{Q_{22}}\right],$$

so that we obtain a block lower triangular matrix as that shown in Fig. 1 (right).

The (rough) overall cost of this algorithm, provided $b \ll n$, is reported in the column labeled as "Blocked Householder" in Table 1. Thus, the cost is (asymptotically) as low as that of the Givens-based algorithm, but we now enable the use of Level-3 BLAS (at least) in the application of the orthogonal transformations.

4 Concluding Remarks

We have described two variants of the inverse-free iteration for the matrix disk function that reduce the theoretical cost of the traditional iterative scheme by 32%. The first implementation employs Givens rotations to compute the QR factorization, and results in a fine-grain computation. Preliminary results show this approach does not achieve a noticeable reduction in the execution time of the algorithm. The second variant, however, carries out most of its computation in terms of Level-3 BLAS operation and is expected to outperform the traditional algorithm.

References

1. Z. Bai, J. Demmel, and M. Gu. An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems. *Numer. Math.*, 76(3):279–308, 1997.
2. A.N. Malyshev. Parallel algorithm for solving some spectral problems of linear algebra. *Linear Algebra Appl.*, 188/189:489–520, 1993.
3. X. Sun and E.S. Quintana-Ortí. Spectral division methods for block generalized Schur decompositions. *Math. Comp.*, 73:1827–1847, 2004.