Universiteit Utrecht

Department
of Mathematics

Jacobi-Davidson algorithms for various eigenproblems
- A working document -

by

Gerard L.G. Sleijpen, Henk A. Van der Vorst, and Zhaojun Bai

Preprint
nr. 1114
August 17, 1999

JACOBI-DAVIDSON ALGORITHMS FOR VARIOUS EIGENPROBLEMS
- *A WORKING DOCUMENT* -

GERARD L.G. SLEIJPEN,* HENK A. VAN DER VORST,* AND ZHAOJUN BAI‡

The sections in this report will appear as sections in the book:

Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. Van der Vorst (editors),
Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide,
SIAM, to appear in 2000.

The numbering of the Chapters and Sections in this report refers to the numbers that these sections will have in that book. The page-numbers and the equation-numbers, however, are different.

* Departement of Mathematics, Utrecht University, P.O. Box 80.010, NL-3508 TA Utrecht, The Netherlands. E-mail: `sleijpen@math.uu.nl`, `vorst@math.uu.nl`.

‡ Department of Mathematics, University of Kentucky, Lexington, KY, E-mail: `bai@ms.uky.edu`.

Hermitian Eigenproblem

4.7 Jacobi-Davidson methods

G. Sleijpen and H. van der Vorst

The Lanczos method (see [2, Section 4.4]) is quite effective to compute eigenvalues in the ends of the spectrum of A if these eigenvalue are well separated from the remaining spectrum, or when it can be applied to $(A - \sigma I)^{-1}$, for some reasonable guess σ for an eigenvalue.

If none of these conditions is fulfilled, for instance if the computation of a vector $(A - \sigma I)^{-1} y$ for given y is not feasible with a direct solver, then variants of the Jacobi-Davidson method [32] offer an attractive alternative.

4.7.1 Basic Theory

The Jacobi-Davidson method is based on a combination of two basic principles. The first one is to apply a Ritz-Galerkin approach for the eigenproblem $Ax = \lambda x$, with respect to some given subspace spanned by an orthonormal basis $v_1, \ldots, v_m$. The usage of other than Krylov subspaces was suggested by Davidson [7] who also suggested specific choices, different from the ones that we will take, for the construction of orthonormal basis vectors v_j. The Ritz-Galerkin condition is

$$AV_m s - \theta V_m s \perp \{v_1, \ldots, v_m\},$$

which leads to the reduced system

$$V_m^* A V_m s - \theta s = 0, \tag{4.1}$$

where V_m denotes the matrix with columns v_1 to v_m. This equation has m solutions $(\theta_j^{(m)}, s_j^{(m)})$. The m pairs $(\theta_j^{(m)}, u_j^{(m)} \equiv V_m s_j^{(m)})$ are called the Ritz values and Ritz vectors of A with respect to the subspace spanned by the columns of V_m. These Ritz pairs are approximations for eigenpairs of A, and our goal is to get better approximations by a well chosen expansion of the subspace.

At this point the other principle behind the Jacobi-Davidson approach comes into the play. The idea goes back to Jacobi [17]. Suppose that we have an eigenvector approximation $u_j^{(m)}$ for an eigenvector x corresponding to a given eigenvalue λ. Then Jacobi suggested (in the original paper for strongly diagonally dominant matrices) to compute the orthogonal correction t for $u_j^{(m)}$ so that

$$A(u_j^{(m)} + t) = \lambda(u_j^{(m)} + t).$$

Since $t \perp u_j^{(m)}$, we can restrict ourselves to the subspace orthogonal to $u_j^{(m)}$. The operator A restricted to that subspace is given by

$$\left(I - u_j^{(m)} u_j^{(m)^*}\right) A \left(I - u_j^{(m)} u_j^{(m)^*}\right),$$

and we find that t satisfies the equation

$$\left(I - u_j^{(m)} u_j^{(m)^*}\right) (A - \lambda I) \left(I - u_j^{(m)} u_j^{(m)^*}\right) t = -(A - \theta_j^{(m)} I) u_j^{(m)}.$$

In practical situations we do not know λ and the obvious solution to this is to replace it by its approximation $\theta_j^{(m)}$, which leads to the *Jacobi-Davidson correction equation* for an update $t_j^{(m)} \perp u_j^{(m)}$:

$$\begin{aligned} \left(I - u_j^{(m)} u_j^{(m)^*}\right) (A - \theta_j^{(m)} I) \left(I - u_j^{(m)} u_j^{(m)^*}\right) t_j^{(m)} \\ = -r_j^{(m)} \equiv -(A - \theta_j^{(m)} I) u_j^{(m)}. \end{aligned} \tag{4.2}$$

This correction equation is solved only approximately and its approximate solution $\widetilde{t}^{(m)}$ is taken for the expansion of the subspace. This is a fundamental difference with the Krylov subspace methods; instead of selecting a subspace as powers of an operator acting on a given starting vector, we select some subspace without Krylov structure and we project the given matrix onto this subspace.

From (4.1) we conclude that $r_j^{(m)} \perp \{v_1, \ldots, v_m\}$, and in particular that $r_j^{(m)} \perp u_j^{(m)}$, so that the Jacobi-Davidson correction equation represents a consistent linear system.

It can be shown that the exact solution of (4.2) leads to cubic convergence of the largest $\theta_j^{(m)}$ towards $\lambda_{\max}(A)$, for increasing m (similar statements can be made for the convergence towards other eigenvalues of A, provided that the Ritz values are selected appropriately in each step).

In [32] it is suggested to solve equation (4.2) only approximately, for instance by some steps of MINRES [25], with an appropriate preconditioner K for $A - \theta_j^{(m)} I$, if available, but in fact any approximation technique for $t_j^{(m)}$ is formally allowed, provided that the projectors $\left(I - u_j^{(m)} u_j^{(m)^*}\right)$ are taken into account.
In our templates we will present ways to approximate $t_j^{(m)}$ with Krylov subspace methods.

We will now discuss how to use preconditioning for an iterative solver like GMRES or CGS, applied with equation (4.2). Suppose that we have a left preconditioner K available for the operator $A - \theta_j^{(m)} I$, for which in some sense $K^{-1}(A - \theta_j^{(m)} I) \approx I$. Since $\theta_j^{(m)}$ varies with the iteration count m, so may the preconditioner, although it is often practical to work with the same K for different values of θ. Of course, the preconditioner K has to be restricted to the subspace orthogonal to $u_j^{(m)}$ as well, which means that we have to work effectively with

$$\widetilde{K} \equiv \left(I - u_j^{(m)} u_j^{(m)^*}\right) K \left(I - u_j^{(m)} u_j^{(m)^*}\right).$$

This may seem unnecessarily complicated, and at first sight it seems also to involve a lot of overhead because of the projections $\left(I - u_j^{(m)} u_j^{(m)^*}\right)$ that encircle the matrix vector operations. But it all amounts to a handful of rather simple operations, as we will show now.

We will from now on assume that we use a Krylov solver with initial guess $t_0 = 0$ and with left preconditioning for the approximate solution of the correction equation (4.2). Since the starting vector is in the subspace orthogonal to $u_j^{(m)}$, all iteration vectors for the Krylov solver will be in that space. In that subspace we have to compute the vector $z \equiv \widetilde{K}^{-1} \widetilde{A} v$, for a vector v supplied by the Krylov solver, and

$$\widetilde{A} \equiv \left(I - u_j^{(m)} u_j^{(m)^*}\right) (A - \theta_j^{(m)} I) \left(I - u_j^{(m)} u_j^{(m)^*}\right).$$

We will do this in two steps and first we have to compute

$$
\begin{aligned}
\widetilde{A}v &= \left(I - u_j^{(m)}u_j^{(m)^*}\right)(A - \theta_j^{(m)}I)\left(I - u_j^{(m)}u_j^{(m)^*}\right)v \\
&= \left(I - u_j^{(m)}u_j^{(m)^*}\right)y \\
\text{with } y &\equiv (A - \theta_j^{(m)}I)v \quad \text{since} \quad u_j^{(m)^*}v = 0\ .
\end{aligned}
$$

Then, with left-preconditioning we have to solve $z \perp u_j^{(m)}$ from

$$\widetilde{K}z = \left(I - u_j^{(m)}u_j^{(m)^*}\right)y.$$

Since $z \perp u_j^{(m)}$, it follows that z satisfies $Kz = y - \alpha u_j^{(m)}$ or $z = K^{-1}y - \alpha K^{-1}u_j^{(m)}$. The condition $z \perp u_j^{(m)}$, leads to

$$\alpha = \frac{u_j^{(m)^*}K^{-1}y}{u_j^{(m)^*}K^{-1}u_j^{(m)}}.$$

The vector $\widehat{y} \equiv K^{-1}y$ is solved from $K\widehat{y} = y$ and, likewise, $\widehat{u} \equiv K^{-1}u_j^{(m)}$ is solved from $K\widehat{u} = u_j^{(m)}$. Note that the last equation has to be solved only once in an iteration process for equation (4.2), so that effectively $i_S + 1$ operations with the preconditioner are required for i_S iterations of the linear solver. Note also that one matrix vector multiplication with the left-preconditioned operator, in an iteration of the Krylov solver, requires only one inner product and one vector update instead of the four actions of the projector operator $\left(I - u_j^{(m)}u_j^{(m)^*}\right)$. This has been worked out in the solution template, given in Alg. 4.2. Along similar lines one can obtain an efficient template, although slightly more expensive than the left-preconditioned case, for the right-preconditioned operator. This template is described in Alg. 4.3.
For more details on preconditioning of the correction equation, see [33].

If we form an approximation for $t_j^{(m)}$ in (4.2) as $\widetilde{t}^{(m)} \equiv K^{-1}r - \alpha K^{-1}u_j^{(m)}$, with α such that $\widetilde{t}^{(m)} \perp u_j^{(m)}$, and without acceleration by an iterative solver, then we obtain a process which was suggested by Olsen et al [23].

4.7.2 The Basic Algorithm

The basic form of the Jacobi-Davidson algorithm is given in Alg. 4.1. Later on we will describe more sophisticated variants with restart and other strategies.

In each iteration of this algorithm an approximated eigenpair (θ, u) for the eigenpair of the Hermitian matrix A, corresponding to the largest eigenvalue of A, is computed. The iteration process is terminated as soon as the norm of the residual $Au - \theta u$ is below a given threshold ϵ.

ALGORITHM 4.1: **Jacobi-Davidson Method for** $\lambda_{\max}(A)$

(1) *Start with* $t = v_0$*, starting guess*
for $m = 1, ...$
(2) **for** $i = 1, .., m-1$
$t = t - (t^* v_i) v_i$
$v_m = t/||t||_2$, $v_m^A = A v_m$
(3) **for** $i = 1, .., m$
$M_{i,m} = v_i^* v_m^A$
(4) *Compute the largest eigenpair* $Ms = \theta s$
of the m *by* m *Hermitian matrix* M*,* $(||s||_2 = 1)$
$u = Vs$
$u^A = V^A s$
$r = u^A - \theta\, u$
(5) **if** $(||r||_2 \leq \varepsilon)$, $\widetilde{\lambda} = \theta$, $\widetilde{x} = u$, **then** *STOP.*
(8) *Solve (approximately) a* $t \perp u$ *from*
$(I - uu^*)(A - \theta I)(I - uu^*) t = -r$

To apply this algorithm we need to specify a starting vector v_0, and a tolerance ϵ.

On completion an approximation for the largest eigenvalue $\lambda = \lambda_{\max}(A)$ and its corresponding eigenvector $x = x_{\max}$ is delivered. The computed eigenpair $(\widetilde{\lambda}, \widetilde{x})$ satisfies $||A\widetilde{x} - \widetilde{\lambda}\widetilde{x}|| \leq \epsilon$.

We will now describe some implementation details, referring to the respective phases in Alg. 4.1.

(1) This is the initialization phase of the process. The search subspace is each time expanded by a vector t, and we start this process by a given vector. Ideally, this vector should have a significant component in the direction of the wanted eigenvector. Unless one has some idea of the wanted eigenvector, it may be a good idea to start with a random vector as a starting vector. This gives some confidence that the wanted eigenvector has a non-zero component in the starting vector, which is necessary for detection of the eigenvector.

(2) The first two lines represent the modified Gram-Schmidt process for the orthogonalization of the new vector t with respect to the set $v_1, \ldots, v_{m-1}$. If $m = 1$ then this is an empty loop. Let t_{in} represent the vector before the start of modified Gram-Schmidt, and t_{out} the vector that results at completion of phase (2). It is advisable (see [6]) to repeat the Gram-Schmidt process one time if $||t_{out}||_2/||t_{in}||_2 \leq \kappa$, where κ is a modest constant, say $\kappa = .25$. This guarantees that the loss of orthogonality is restricted to $1/\kappa$ times machine precision, in a relative sense. The template for this modified Gram-Schmidt orthogonalization with iterative refinement is given in Alg. 4.4.

(3) Computation of the mth column of the upper triangular part of the matrix $M \equiv V^*AV$. The matrix V denotes the n by m matrix with columns v_j, V^A likewise.

(4) The largest eigenpair of the $m \times m$ Hermitian matrix M, with elements $M_{i,j}$ in its upper triangular part, can be done with the appropriate routines from LAPACK (see [2, Section 4.2]).

The vector u^A may be alternatively computed as $u^A = Au$, depending on which way is cheapest. The choice is between an m-fold update and another multiplication with A; if A has less than m nonzero elements on average per row, then the computation via Au is preferable. If u^A is computed as Au then it is not necessary to store the vectors v_j^A.

(5) The algorithm is terminated if $||Au - \theta u||_2 \leq \epsilon$. In that case A has an eigenvalue λ for which $|\lambda - \theta| \leq \epsilon$. For the corresponding normalized eigenvector there is a similar bound on the angle, provided that λ is simple and well separated from the other eigenvalues of A, see [2, Eqn. (4.4)]. That case leads also to a sharper bound for λ (see [2, Eqn. (4.5)]).

Convergence to a $\lambda \neq \lambda_{\max}(A)$ may take place, but is in general unlikely. It happens, for instance, if $v_0 \perp x_{\max}$, or if the selected θ is very close to an eigenvalue $\lambda \neq \lambda_{\max}(A)$. This may happen for any iterative solver, in particular if ϵ is taken not small enough (say larger than $\sqrt{\mathbf{u}}$).

(8) The approximate solution for the expansion vector t can be computed with a Krylov solver, for instance, MINRES, SYMMLQ, or CGS. With left- or right-preconditioning one has to select a Krylov solver for unsymmetric systems (like GMRES, CGS, or Bi-CGSTAB), since the preconditioned operator is in general not symmetric. A template for the approximate solution, with a left-preconditioned Krylov subspace method of choice, is given in Alg. 4.2. The right-preconditioned case, which is slightly more expensive, is covered by the template in Alg. 4.3. For iterative Krylov subspace solvers see [4]. The approximate solution has to be orthogonal to u, but that is automatically the case with Krylov solvers if one starts with an initial guess orthogonal to u, for instance $t_0 = 0$. In most cases it is not necessary to solve the correction equation to high precision; a relative precision of 2^{-m} in the mth iteration seems to suffice. It is advisable to put a limit to the number of iteration steps for the iterative solver.
Davidson [7] suggested to take $t = (\mathrm{diag}(A) - \theta I)^{-1} r$, but in this case t is not orthogonal with respect to u. Moreover, for diagonal matrices this choice leads to stagnation which is an illustration of the problems in this approach.

In order to restrict storage, the algorithm can be terminated at some appropriate value $m = m_{\max}$, and restarted with for v_0 the latest value of u. We will describe a variant of the Jacobi-Davidson algorithm with a more sophisticated restart strategy in Section 4.7.3.

Note that most of the computationally intensive operations, i.e., those operations of which the cost is proportional to n, can be done easily in parallel computation mode. Also, the multiple vector updates can be done by the appropriate BLAS2 routines (see [2, Section 10.4]).

ALGORITHM 4.2: **Approximate solution of the Jacobi-Davidson correction equation with left-preconditioning**

"Solve" with left preconditioner $\widetilde{K} \equiv (I - uu^*)K(I - uu^*)$,
for $\widetilde{A} \equiv (I - uu^*)(A - \theta I)(I - uu^*)$:

(8a) *Solve* $\widehat{u}$ *from* $K\widehat{u} = u$, $\mu \equiv u^*\widehat{u}$
Compute $\widetilde{r} \equiv \widetilde{K}^{-1} r$ *as:*
(b′) *solve* $\widehat{r}$ *from* $K\widehat{r} = r$
(c′) $\widetilde{r} = \widehat{r} - \frac{u^*\widehat{r}}{\mu}\,\widehat{u}$
Apply Krylov subspace method with start $t_0 = 0$,
with operator $\widetilde{K}^{-1}\widetilde{A}$, *and right-hand side* $-\widetilde{r}$,
$z = \widetilde{K}^{-1}\widetilde{A}v$ *for given* v *is computed as:*
(a) $y = (A - \theta I)v$
(b) *solve* $\widehat{y}$ *from* $K\widehat{y} = y$
(c) $z = \widehat{y} - \frac{u^*\widehat{y}}{\mu}\,\widehat{u}$

Algorithm 4.3: **Approximate solution of the Jacobi-Davidson correction equation with right-preconditioning**

"Solve" with right preconditioner $\widetilde{K} \equiv (I - uu^*)K(I - uu^*)$,
for $\widetilde{A} \equiv (I - uu^*)(A - \theta I)(I - uu^*)$:

(8a) *Solve* $\widehat{u}$ *from* $K\widehat{u} = u$, $\mu \equiv u^*\widehat{u}$
Apply Krylov subspace method with start $t_0 = 0$,
with operator $\widetilde{A}\,\widetilde{K}^{-1}$, *and right-hand side* $-r$,
$z = \widetilde{A}\,\widetilde{K}^{-1}v$ *for given* v *is computed as:*
(a) *solve* $\widehat{v}$ *from* $K\widehat{v} = v$
(b) $y = \widehat{v} - \frac{u^*\widehat{v}}{\mu}\,\widehat{u}$
(c) $z = (I - uu^*)(A - \theta I)y$
The approximate solution t *should be back-transformed as:*
(a″) *solve* $\widehat{t}$ *from* $K\widehat{t} = t$
(b″) $t = \widehat{t} - \frac{u^*\widehat{t}}{\mu}\,\widehat{u}$

Algorithm 4.4: **Modified Gram-Schmidt orthogonalization with refinement**

(2a) *Select a value for* κ *less than* 1, *say* $\kappa = .25$
$\tau_{in} = ||t||_2$
for $i = 1, .., m-1$
$t = t - (t^*v_i)v_i$
if $||t||_2/\tau_{in} \le \kappa$
for $i = 1, .., m-1$
$t = t - (t^*v_i)v_i$

In the coming subsections we will describe more sophisticated variants of the Jacobi-Davidson algorithm. In Section 4.7.3 we will introduce a variant that allows for restarts, which is very convenient if one wants to keep the dimensions of the involved subspaces limited. The variant is also suitable for a restart after an eigenpair has been discovered, in order to locate a next eigenpair. The technique is based on deflation. The resulting algorithm is designed for the computation of a few of the largest or smallest eigenvalues of a given matrix.

In Section 4.7.4 we will describe a variant of Jacobi-Davidson that is suitable for the computation of interior eigenvalues of A.

Storage and computational costs We have collected the dominant costs of the simple Jacobi-Davidson approach, in terms of storage and floating point operations, in two Tables. The costs are given for iteration m of the algorithm.

item	*storage*
search space	$2m$ n-vectors
residual	2 n-vector
approx. eigenvector.	1 n-vector
projected system	.5 matrix of order m
eigenvectors of proj. system	1 matrix of order m
correction equation	depends on selected solver

action	*work*
search basis	$m+1$ dots, m updates in iteration m
projected system	m dots
eigensystem projected system	$\mathcal{O}(m^3)$
residual	1 Matrix vector product, 1 update or m-fold update
approx. eigenvector	m-fold update
correction equation	depends on choice solver

4.7.3 Jacobi-Davidson with restart and deflation

4.7.3.1 Restart strategy

The increasing storage or the computational overhead, for increasing dimension m of the subspace, may make it necessary to restart. An obvious way to restart is to take the most recent approximation for the desired eigenvector. However, this may not be the most efficient strategy for restarting. With any restart by a single vector we discard possibly valuable information that is contained in the remaining part of the subspace. Unless we have an invariant subspace, all vectors in the subspace contain information for the wanted eigenvector. After restart with one single vector we see the effect of the lost information by a slowdown in the speed of convergence. Therefore it is often better to restart with a set of vectors representing a subspace that contains more information for the wanted eigenpair and a good strategy is to restart with the subspace spanned by the Ritz vectors of a small number of the Ritz values closest to a specified target value.

4.7.3.2 Deflation

When a Ritz value is close enough to an eigenvalue, the remaining part of the current subspace will already have rich components in nearby eigenpairs, since we have selected in all steps the Ritz vectors for Ritz values close to the desired eigenvalue. We can use this information as the basis for a subspace for the computation of a next eigenvector. In order to avoid that the old eigenvector will re-enter the computational process, we make the new search vectors in the Jacobi-Davidson algorithm explicitly orthogonal to the computed eigenvectors. This technique is called explicit *deflation*. We will discuss this in slightly more detail.

Let $\widetilde{x}_1, \ldots, \widetilde{x}_{k-1}$ denote the accepted eigenvector approximations and let us assume that these vectors are orthonormal. The matrix $\widetilde{X}_{k-1}$ has the vectors $\widetilde{x}_j$ as its columns. In order to find the next eigenvector $\widetilde{x}_k$, we apply the Jacobi-Davidson algorithm to the deflated matrix[1]

$$(I - \widetilde{X}_{k-1}\widetilde{X}_{k-1}^*)\, A\, (I - \widetilde{X}_{k-1}\widetilde{X}_{k-1}^*),$$

and this leads to a correction equation of the form

$$P_m(I - \widetilde{X}_{k-1}\widetilde{X}_{k-1}^*)\,(A - \theta_j^{(m)} I)\,(I - \widetilde{X}_{k-1}\widetilde{X}_{k-1}^*)P_m t_j^{(m)} = -r_j^{(m)}, \tag{4.3}$$

with $P_m \equiv \left(I - u_j^{(m)} u_j^{(m)^*}\right)$, that has to be solved for the correction $t_j^{(m)}$ to each new eigenvector approximation $u_j^{(m)}$, with corresponding Ritz value $\theta_j^{(m)}$. In [12] it is shown, by numerical evidence, that the explicit deflation against the vectors represented by $\widetilde{X}_{k-1}$ is highly recommended for the correction equation, but is is not necessary to include this deflation in the computation of the projected matrix (the projection of A onto the subspace spanned by the successive approximations v_j in the search for the kth eigenvector). The projected matrix can be computed as $V_m^* A V_m$, without significant loss of accuracy.

[1] Deflation with approximate eigenvectors may introduce an error of order ϵ^2 on the eigenvalues, provided that the computed eigenvalues are well separated from the remaining ones [26, Section 5.1].

4.7.3.3 Preconditioning

Preconditioning for an iterative solver like GMRES or CGS, applied with equation (4.3) is only slightly more complicated than in the single vector case (cf. Sec. 4.7.1). Suppose that we have a left preconditioner K available for the operator $A - \theta_j^{(m)} I$. Let $\widetilde{Q}$ denote the matrix $\widetilde{X}_{k-1}$ expanded with $u_j^{(m)}$ as its kth column. In this case, the preconditioner K has to be restricted to the subspace orthogonal to $\widetilde{Q}$ as well, which means that we have to work effectively with

$$\widetilde{K} \equiv (I - \widetilde{Q}\widetilde{Q}^*)K(I - \widetilde{Q}\widetilde{Q}^*).$$

Similar to the single vector case this can be realized in a surprisingly efficient way.

Assume that we use a Krylov solver with initial guess $t_0 = 0$ and with left preconditioning for the approximate solution of the correction equation (4.3). Since the starting vector is in the subspace orthogonal to $\widetilde{Q}$, all iteration vectors for the Krylov solver will be in that space. In that subspace we have to compute the vector $z \equiv \widetilde{K}^{-1}\widetilde{A}v$, for a vector v supplied by the Krylov solver, and

$$\widetilde{A} \equiv (I - \widetilde{Q}\widetilde{Q}^*)(A - \theta_j^{(m)} I)(I - \widetilde{Q}\widetilde{Q}^*).$$

This is done in two steps and first we compute

$$\begin{aligned} \widetilde{A}v &= (I - \widetilde{Q}\widetilde{Q}^*)(A - \theta_j^{(m)} I)(I - \widetilde{Q}\widetilde{Q}^*)v \\ &= (I - \widetilde{Q}\widetilde{Q}^*)y \\ \text{with } y &\equiv (A - \theta_j^{(m)} I)v \ \text{ since } \ \widetilde{Q}^* v = 0\ . \end{aligned}$$

Then, with left-preconditioning we have to solve $z \perp \widetilde{Q}$ from

$$\widetilde{K}z = (I - \widetilde{Q}\widetilde{Q}^*)y.$$

Since $\widetilde{Q}^* z = 0$, it follows that z satisfies $Kz = y - \widetilde{Q}\vec{\alpha}$ or $z = K^{-1}y - K^{-1}\widetilde{Q}\vec{\alpha}$. The condition $\widetilde{Q}^* z = 0$ leads to

$$\vec{\alpha} = (\widetilde{Q}^* K^{-1}\widetilde{Q})^{-1}\widetilde{Q}^* K^{-1} y.$$

The vector $\widehat{y} \equiv K^{-1}y$ is solved from $K\widehat{y} = y$ and, likewise, $\widehat{Q} \equiv K^{-1}\widetilde{Q}$ is solved from $K\widehat{Q} = \widetilde{Q}$. Note that the last set of equations has to be solved only once in an iteration process for equation (4.3), so that effectively $i_S + k$ operations with the preconditioner are required for i_S iterations of the linear solver. Note also that one matrix vector multiplication with the left-preconditioned operator, in an iteration of the Krylov solver, requires only one operation with $\widetilde{Q}^*$ and $K^{-1}\widetilde{Q}$, instead of the four actions of the projector operator $(I - \widetilde{Q}\widetilde{Q}^*)$. This has been worked out in the solution template, given in Alg. 4.6. Note that obvious savings can be realized if the operator K is kept the same for a number of successive eigenvalue computations (for details, see [33]).

4.7.3.4 Jacobi-Davidson template with restart and deflation for exterior eigenvalues

The complete algorithm for the Jacobi-Davidson method, that includes restart with a number of Ritz vectors, and deflation for the computation of a number of eigenpairs is called JDQR [12], since it can be interpreted as an iterative approach for the QR algorithm. The template for this algorithm is given in Alg. 4.5.

ALGORITHM 4.5: **Jacobi-Davidson for $k_{\max}$ exterior eigenvalues**

Start with v_0 starting vector and τ target value.
(1) $t = v_0$, $k = 0$, $m = 0$; $\widetilde{X} = [\,]$,
while $k < k_{\max}$
(2) **for** $i = 1, .., m$
    $t = t - (v_i^* t) v_i$
  $m = m + 1$; $v_m = t/\|t\|_2$, $v_m^A = Av_m$
(3) **for** $i = 1, .., m$
    $M_{i,m} = v_i^* v_m^A$
(4) *Make eigendecomposition $MS = S\Theta$*
  of the m by m Hermitian matrix M.
  Sort the pairs: $|\theta_i - \tau| \geq |\theta_{i-1} - \tau|$
  $u = Vs_1$, $u^A = V^A s_1$, $r = u^A - \theta_1 u$
(5) **while** $\|r\|_2 \leq \epsilon$
    $\widetilde{\lambda}_{k+1} = \theta_1$, $\widetilde{X} = [\widetilde{X}, u]$, $k = k + 1$
    if $k = k_{\max}$ **then** *STOP*
(6) $m = m - 1$, $M = 0$
    for $i = 1, .., m$
      $v_i = Vs_{i+1}$, $v_i^A = V^A s_{i+1}$,
      $M_{i,i} = \theta_{i+1}$, $s_i = e_i$, $\theta_i = \theta_{i+1}$
    $u = v_1$, $r = v_1^A - \theta_1 u$
(7) **if** $m \geq m_{\max}$ **then**
    $M = 0$
    for $i = 2, .., m_{\min}$
      $v_i = Vs_i$, $v_i^A = V^A s_i$, $M_{i,i} = \theta_i$
    $v_1 = u$, $v_1^A = u^A$, $M_{1,1} = \theta_1$, $m = m_{\min}$
  $\theta = \theta_1$, $\widetilde{Q} = [\widetilde{X}, u]$
(8) *Solve t ($\perp \widetilde{Q}$) (approximately) from:*
    $(I - \widetilde{Q}\widetilde{Q}^*)(A - \theta I)(I - \widetilde{Q}\widetilde{Q}^*)t = -r$

To apply this algorithm we need to specify a starting vector v_0, a tolerance ϵ, a target value τ, and a number $k_{\max}$ that specifies how many eigenpairs near τ should be computed. The value of $m_{\max}$ denotes the maximum dimension of the search subspace. If it is exceeded then a restart takes place with a subspace of specified dimension $m_{\min}$.

On completion typically the $k_{\max}$ largest eigenvalues are delivered when τ is chosen larger than $\lambda_{\max}(A)$; the $k_{\max}$ smallest eigenvalues are delivered if τ is chosen smaller than $\lambda_{\min}(A)$. The computed eigenpairs $(\widetilde{\lambda}_j, \widetilde{x}_j)$, $\|\widetilde{x}_j\|_2 = 1$, satisfy $\|A\widetilde{x}_j - \widetilde{\lambda}_j \widetilde{x}_j\|_2 \leq j\epsilon$, where $\widetilde{x}_j$ denotes the jth column of $\widetilde{X}$.

In principle, this algorithm computes the $k_{\max}$ eigenvalues closest to a specified target value τ. This is only reliable if the $k_{\max}$ largest or $k_{\max}$ smallest eigenvalues are wanted. For interior sets of eigenvalues we will describe safer techniques in Section 4.7.4. We will now comment on some parts of the algorithm in view of our discussions in previous subsections.

(1) Initialization phase. Search subspace is initialized with t.

(2) The new expansion vector for the search subspace is made orthogonal with respect to the current search subspace by means of modified Gram-Schmidt. This can be replaced, for improved numerical stability, by the template given in Alg. 4.4.

  If $m = 0$ then this is an empty loop.

(3) We compute only the upper triangular part of the Hermitian matrix $M \equiv V^*AV$ (of order m).

(4) The eigenproblem for the m by m matrix M can be solved by a standard eigensolver for dense Hermitian eigenproblems from LAPACK. We have chosen to compute the standard Ritz values, which makes the algorithm suitable for computing the largest or smallest $k_{\max}$ eigenvalues of A. If one wishes to compute $k_{\max}$ eigenvalues somewhere in the interior of the spectrum then the usage of harmonic Ritz values is advocated, see Section 4.7.4.

The matrix V denotes the n by m matrix with columns v_j, $V^A \equiv AV$ likewise; S is the m by m matrix with columns s_j and $\Theta = \mathrm{diag}\,(\theta_1, \ldots, \theta_m)$.

(5) The stopping criterion is to accept an eigenvector approximation as soon as the norm of the residual (for the normalized eigenvector approximation) is below ϵ. This means that we accept inaccuracies in the order of ϵ^2 in the computed eigenvalues, and inaccuracies (in angle) in the eigenvectors in the order of ϵ, provided that the associated eigenvalue is simple and well separated from the other eigenvalues, see [2, Eqn. (4.4)].

Occasionally one of the wanted eigenvectors of A may be undetected, for instance if v_0 has no component in the corresponding eigenvector direction. For a random start vector this is rather unlikely. (See also note (5) for Alg. 4.1).

(6) After acceptance of a Ritz pair, we continue the search for a next eigenpair, with the remaining Ritz nvectors as a basis for the initial search space.

(7) We restart as soon as the dimension of the search space for the current eigenvector exceeds $m_{\max}$. The process is restarted with the subspace spanned by the $m_{\min}$ Ritz vectors corresponding to the Ritz values closest to the target value τ.

(8) We have collected the locked (computed) eigenvectors in $\widetilde{X}$, and the matrix $\widetilde{Q}$ is $\widetilde{X}$ expanded with the current eigenvector approximation u. This is done in order to obtain a more compact formulation; the correction equation is equivalent to the one in (4.3). The new correction t has to be orthogonal to the columns of $\widetilde{X}$ as well as to u.
Of course, the correction equation can be solved by any suitable process, for instance a preconditioned Krylov subspace method. Because of the occurrence of $\widetilde{Q}$ one has to be careful with the usage of preconditioners for the matrix $A - \theta I$. The inclusion of preconditioners can be done following the same principles as for the single vector Jacobi-Davidson algorithm, see Alg. 4.6 for a Template. Make sure that the starting vector t_0 for an iterative solver satisfies the orthogonality constraints $\widetilde{Q}^* t_0 = 0$. Note that significant savings per step can be made in Alg. 4.6 if K is kept the same for a (few) Jacobi-Davidson iterations. In that case columns of $\widehat{Q}$ can be saved from previous steps. Also the matrix $\mathcal{M}$ in Alg. 4.6 can be updated from previous steps, as well as its $\mathcal{LU}$ decomposition.

Algorithm 4.6: **Approximate solution of the deflated Jacobi-Davidson correction equation**

"Solve" with left preconditioner $\widetilde{K} \equiv (I - \widetilde{Q}\widetilde{Q}^*)K(I - \widetilde{Q}\widetilde{Q}^*)$,
for $\widetilde{A} \equiv (I - \widetilde{Q}\widetilde{Q}^*)(A - \theta I)(I - \widetilde{Q}\widetilde{Q}^*)$:

(8a) *Solve* $\widehat{Q}$ *from* $K\widehat{Q} = \widetilde{Q}$
Compute $\mathcal{M} = \widetilde{Q}^*\widehat{Q}$
Decompose $\mathcal{M} = \mathcal{L}\mathcal{U}$
Compute $\widetilde{r} \equiv \widetilde{K}^{-1}r$ *as:*
(b') *solve* $\widehat{r}$ *from* $K\widehat{r} = r$
(c') $\vec{\gamma} = \widetilde{Q}^*\widehat{r}$
solve $\vec{\beta}$ *from* $\mathcal{L}\vec{\beta} = \vec{\gamma}$
solve $\vec{\alpha}$ *from* $\mathcal{U}\vec{\alpha} = \vec{\beta}$
(d') $\widetilde{r} = \widehat{r} - \widehat{Q}\vec{\alpha}$
Apply Krylov subspace method with start $t_0 = 0$,
with operator $\widetilde{K}^{-1}\widetilde{A}$, *and right-hand side* $-\widetilde{r}$,
$z = \widetilde{K}^{-1}\widetilde{A}v$ *for given* v *is computed as:*
(a) $y = (A - \theta I)v$
(b) *solve* $\widehat{y}$ *from* $K\widehat{y} = y$
(c) $\vec{\gamma} = \widetilde{Q}^*\widehat{y}$
solve $\vec{\beta}$ *from* $\mathcal{L}\vec{\beta} = \vec{\gamma}$
solve $\vec{\alpha}$ *from* $\mathcal{U}\vec{\alpha} = \vec{\beta}$
(d) $z = \widehat{y} - \widehat{Q}\vec{\alpha}$

4.7.4 Jacobi-Davidson for interior Eigenvalues

4.7.4.1 The usage of harmonic Ritz values for interior eigenvalues

If one is heading for the eigenpair with the smallest or largest eigenvalue only, then the obvious restart approach works quite well, but often it does not do very well if one is interested in an interior eigenvalue. The problem is that the Ritz values converge monotonically towards exterior eigenvalues, and a Ritz value that is close to a target value in the interior of the spectrum may be well on its way to some other exterior eigenvalue. It may be even the case that the corresponding Ritz vector has only a small component in the direction of the desired eigenvector. It will be clear that such a Ritz vector represents a poor candidate for restart and the question is: what is the optimal vector for restart? The answer is given by the so-called *harmonic Ritz* vectors [21, 24, 32].

As we have seen, the Jacobi-Davidson methods generate basis vectors $v_1, \ldots, v_m$, for a subspace $\mathcal{V}_m$. For the projection of A onto this subspace we have to compute the vectors $w_j \equiv Av_j$. The inverses of the Ritz values of A^{-1}, with respect to the subspace spanned by the w_j, are called the *harmonic Ritz values*. The harmonic Ritz values can be computed without inverting A, since a harmonic Ritz pair $(\widetilde{\theta}_j^{(m)}, \widetilde{u}_j^{(m)})$ satisfies

$$A\widetilde{u}_j^{(m)} - \widetilde{\theta}_j^{(m)}\widetilde{u}_j^{(m)} \perp \mathcal{W}_m \equiv \mathrm{span}(w_1, \ldots, w_m), \tag{4.4}$$

for $\widetilde{u}_j^{(m)} \in \mathcal{V}_m$ and $\widetilde{u}_j^{(m)} \neq 0$. This implies that the harmonic Ritz values are the eigenvalues of the pencil $(W_m^*AV_m, W_m^*V_m)$, or since $W_m = AV_m$:

$$W_m^*W_m s_j^{(m)} - \widetilde{\theta}_j^{(m)} W_m^* V_m s_j^{(m)} = 0\,.$$

For stability reasons we orthonormalize the columns of W_m, and transform the columns of V_m accordingly. This also further simplifies the equation: we see that the harmonic Ritz values are the inverses of the eigenvalues of the symmetric matrix $W_m^*V_m$.

In [24] it is shown that for Hermitian A the harmonic Ritz values converge monotonically towards the smallest non-zero eigenvalues in absolute value. Note that the harmonic Ritz values are unable to identify a zero eigenvalue of A, since that would correspond to an infinite eigenvalue of A^{-1}. Likewise, the harmonic Ritz values for the shifted matrix $A - \tau I$ converge monotonically towards eigenvalues $\lambda \neq \tau$ closest to the target value τ. Fortunately, the search subspace $\mathcal{V}_m$ for the shifted matrix and the unshifted matrix coincide, which facilitates the computation of harmonic Ritz pairs. The harmonic Ritz vector, for the shifted matrix, corresponding to the harmonic Ritz value closest to τ can be interpreted as to maximize a Rayleigh quotient for $(A - \tau I)^{-1}$. It represents asymptotically the best information that is available for the wanted eigenvalue, and hence it represents asymptotically the best candidate as a starting vector after restart, provided that $\tau \neq \lambda$.

4.7.4.2 A template for Jacobi-Davidson for interior eigenvalues

An algorithm for Jacobi-Davidson based on harmonic Ritz values and vectors, combined with restart and deflation, is given in Alg. 4.7. The algorithm can be used for the computation of a number of successive eigenvalues immediately to the right of the target value τ.

ALGORITHM 4.7: **Jacobi-Davidson for k_{max} interior eigenvalues at the right side nearest to τ**

(1) $t = v_0$, $k = 0$; $m = 0$; $\widetilde{X} = [\,]$;
  while $k < k_{\max}$
(2) $w = (A - \tau I)t$
   for $i = 1, .., m$
    $\gamma = w_i^* w$, $w = w - \gamma w_i$, $t = t - \gamma v_i$
   $m = m + 1$, $w_m = w/\|w\|_2$, $v_m = t/\|w\|_2$
(3)   **for** $i = 1, .., m$
    $M_{i,m} = w_i^* v_m$
(4)   *Compute eigendecomposition* $MS = S\widetilde{\Theta}$
   of the m *by* m *Hermitian matrix* M.
   Sort the pairs so that: $\widetilde{\theta}_1 \geq \widetilde{\theta}_2 \geq \ldots$
   $\widetilde{u} = Vs_1$, $\mu = \|\widetilde{u}\|_2$, $u = \widetilde{u}/\mu$, $\vartheta = \widetilde{\theta}_1/\mu^2$
   $\widetilde{w} = Ws_1$, $r = \widetilde{w}/\mu - \vartheta u$
(5)   **while** $\|r\|_2 \leq \epsilon$
    $k = k + 1$, $\widetilde{X} = [\widetilde{X}, u]$, $\widetilde{\lambda}_k = \vartheta + \tau$
    if $k = k_{\max}$ **then** *STOP*
(6) $m = m - 1$, $M = 0$
    for $i = 1, .., m$
     $v_i = Vs_{i+1}$, $w_i = Ws_{i+1}$
     $M_{i,i} = \widetilde{\theta}_{i+1}$, $s_i = e_i$, $\widetilde{\theta}_{i+1} = \widetilde{\theta}_i$
    $\mu = \|v_1\|_2$, $\vartheta = \widetilde{\theta}_2/\mu^2$, $u = v_1/\mu$, $r = w_1/\mu - \vartheta u$
(7)   **if** $m \geq m_{\max}$ **then**
    $M = 0$
    for $i = 2, .., m_{\min}$
     $v_i = Vs_i$, $w_i = Ws_i$, $M_{i,i} = \widetilde{\theta}_i$
    $w_1 = \widetilde{w}$, $v_1 = \widetilde{u}$, $M_{1,1} = \widetilde{\theta}_1$, $m = m_{\min}$
(8) $\theta = \vartheta + \tau$, $\widetilde{Q} = [\widetilde{X}, u]$
   Solve $t \, (\perp \widetilde{Q})$ *(approximately) from:*
    $(I - \widetilde{Q}\widetilde{Q}^*)(A - \theta I)(I - \widetilde{Q}\widetilde{Q}^*)t = -r$

To apply this algorithm we need to specify a starting vector v_0, a tolerance ϵ, a target value τ, and a number $k_{\max}$ that specifies how many eigenpairs near τ should be computed. The value of $m_{\max}$ denotes the maximum dimension of the search subspace. If it is exceeded then a restart takes place with a subspace of specified dimension $m_{\min}$.

On completion the $k_{\max}$ eigenvalues at the right side nearest to τ are delivered. The computed eigenpairs $(\widetilde{\lambda}_j, \widetilde{x}_j)$, $\|\widetilde{x}_j\|_2 = 1$, satisfy $\|A\widetilde{x}_j - \widetilde{\lambda}_j\widetilde{x}_j\|_2 \le j\epsilon$, where $\widetilde{x}_j$ denotes the jth column of $\widetilde{X}$.

For exterior eigenvalues a simpler algorithm has been described in Section 4.7.3. We will now comment on some parts of the algorithm in view of our discussions in previous subsections.

(1) Initialization phase.

(2) The vector $(A - \tau I)t$ is made orthogonal with respect to the current test subspace $\mathcal{W}_m$ by means of modified Gram-Schmidt. This can be replaced, for improved numerical stability, by the template given in Alg. 4.4.

(3) The values $M_{i,j}$ represent elements of the square m by m matrix $M \equiv W^*V$, where V denotes the n by m matrix with columns v_j, and likewise W. Because M is Hermitian, only the upper triangular part of this matrix is computed.

(4) At this point the eigenpairs for the problem $Ms = \widetilde{\theta}s$ should be computed. This can be done with a suitable routine for Hermitian dense matrices from LAPACK. Note that the harmonic Ritz values are just the inverses of the eigenvalues of M.
Then we have to compute the Rayleigh quotient ϑ for Vs_1, and we have to normalize Vs_1, in order to compute a proper residual $r \perp Vs_1$.
We have used that $s_1^*V^*(A - \tau I)Vs_1 = s_1^*M^*s_1 = \widetilde{\theta}_1$.

The vectors s_j are the columns of m by m matrix S and $\widetilde{\Theta} = \mathrm{diag}\,(\widetilde{\theta}_1, \ldots, \widetilde{\theta}_m)$.

(5) The stopping criterion is to accept an eigenvector approximation as soon as the norm of the residual (for the normalized eigenvector approximation) is below ϵ. This means that we accept inaccuracies in the order of ϵ^2 in the computed eigenvalues, and inaccuracies (in angle) in the eigenvectors of $\mathcal{O}(\epsilon)$, provided that the associated eigenvalue is simple and well separated from the others, see [2, Eqn. (4.4)].

Detection of all wanted eigenvalues cannot be guaranteed, see note (5) for Alg. 4.1 and for Alg. 4.5.

(6) This is a restart after acceptance of an approximate eigenpair. The restart is slightly more complicated since two subspaces are involved.

(7) At this point we have a restart when the dimension of the subspace exceeds $m_{\max}$. After a restart the Jacobi-Davidson iterations are resumed with a subspace of dimension $m_{\min}$. The selection of this subspace is based on the harmonic Ritz values nearest to the target τ.

(8) The deflation with computed eigenvectors is represented by the factors with $\widetilde{X}$. The matrix $\widetilde{X}$ has the computed eigenvectors as its columns. If a left preconditioner K is available for the operator $A - \theta I$, then with a Krylov solver similar reductions are realizable as in the situation for exterior eigenvalues. A template for the efficient handling of the left-preconditioned operator is given in Alg. 4.6.

4.7.4.3 Numerical example

We include a numerical example for testing purposes, so that potential users of the Jacobi-Davidson algorithms can verify and compare their results.

The symmetric matrix A is of dimension $n = 1000$. The diagonal entries are $a_{i,i} \equiv i$, the codiagonal entries are $a_{i,i-1} \equiv a_{i,i+1} \equiv 0.5$, and furthermore, $a_{1000,1} \equiv a_{1,1000} \equiv 0.5$. All other entries are zero. This example has been taken from [5], and is discussed, in the context of the Jacobi-Davidson algorithm in [32, page 410].

We use Alg. 4.5, for the computation of the $k_{\max} = 10$ largest eigenvalues. The input parameters have been chosen as follows. The starting vector $v_0 = (0.01, \ldots, 0.01, 1)^T$. The tolerance is

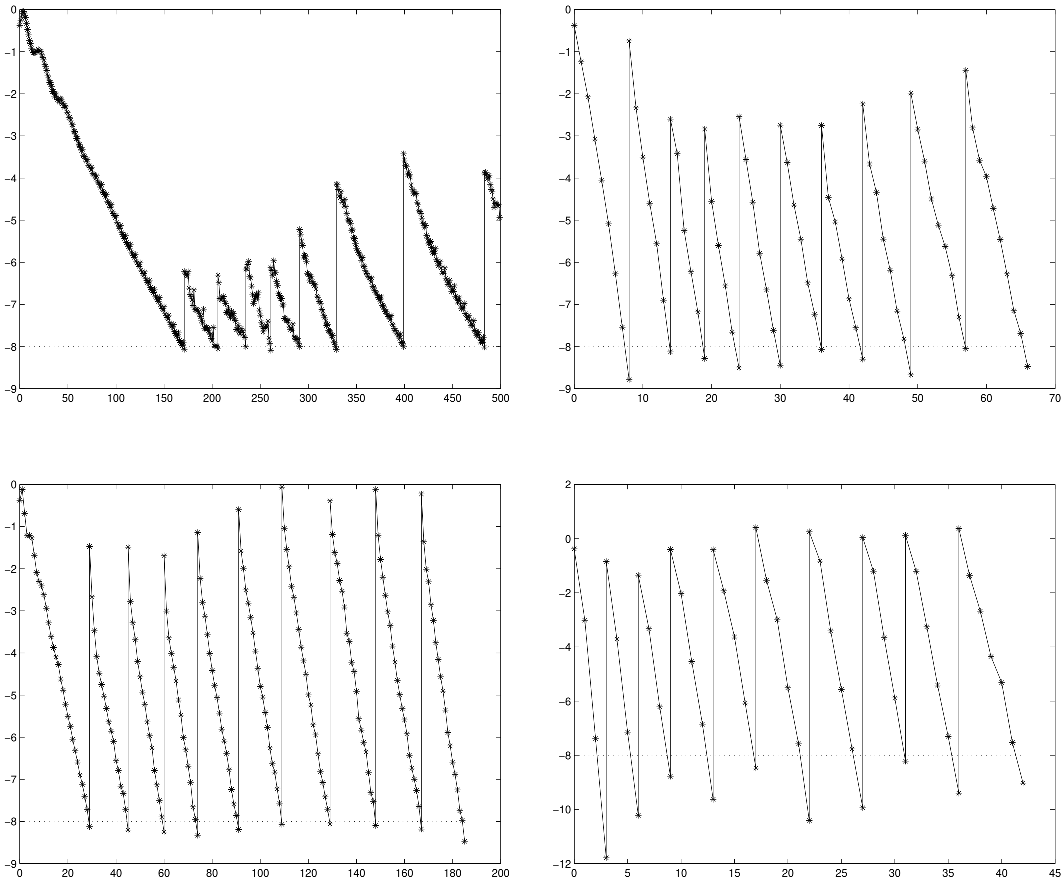

Figure 4.1: Jacobi-Davidson for exterior eigenvalues with several strategies for solving the correction equation.

$\epsilon = 10^{-8}$. The subspace dimension parameters are $m_{\min} = 10$, $m_{\max} = 15$, and the target value $\tau = 1001$.

We show graphically, the norm of the residual vector as a function of the iteration number in Fig. 4.1. Every time when the norm is less than ϵ, then we have determined an eigenvalue within this precision, and the iteration is continued with deflation for the next eigenvalue. The four pictures represent, lexicographically, the following different situations:

1. Top left: this shows what happens when t, in line (8) of Alg. 4.5, is simply taken as $t = -r$. In exact arithmetic, this should deliver the same Ritz values as Arnoldi's algorithm (assuming for Arnoldi a similar restart strategy as in Alg. 4.5).

2. Top right: here we have applied a simple preconditioner $K = \mathrm{diag}(A)$, as in Alg. 4.6, without further subspace acceleration, that is we stopped after step (d$'$). This is equivalent to the method published in [23].

3. Bottom left: this gives the iteration results, for the case where the correction equation has been solved with 5 steps of MINRES, without preconditioning (Alg. 4.5 with $K = I$).

4. Bottom right: here we have used preconditioning as in Alg. 4.5, with $K = \mathrm{diag}(A)$ and 5 steps of GMRES (note that it would have been more efficient to use MINRES, but this requires two-sided preconditioning, for which we did not supply the algorithm).

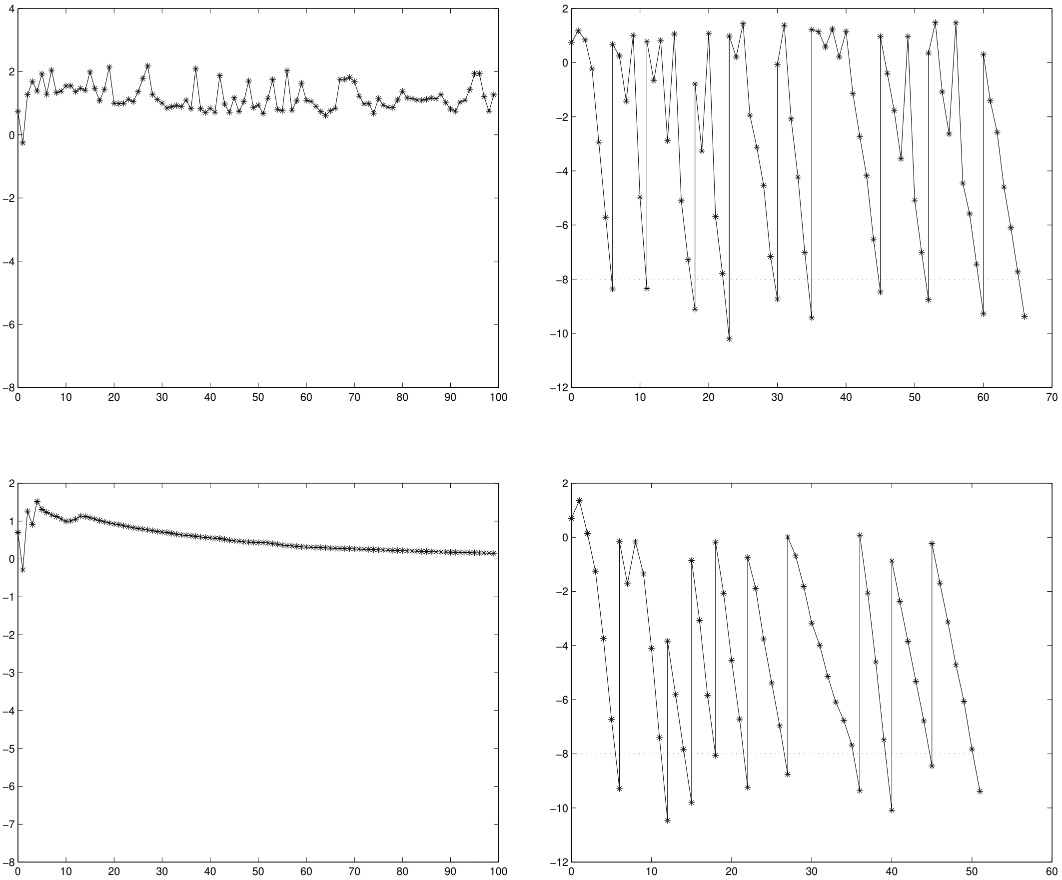

Figure 4.2: Jacobi-Davidson for exterior eigenvalues (top figures) and interior eigenvalues (bottom figures). The correction equations have been solved with 5 steps of plain GMRES (left figures) and with 5 steps of preconditioned GMRES (right figures).

In Fig. 4.2, we give the convergence history for interior eigenvalues, as obtained with Alg. 4.5 (top figures) and with Alg. 4.7 (bottom figures), with the following input specifications:
$v_0 = (0.01, \ldots, 0.01, 1)^T$, $\tau = 900.5$, $\epsilon = 10^{-8}$, $k_{\max} = 10$, $m_{\min} = 5$, and $m_{\max} = 10$. Again, every time when the curve gets below ϵ, then this indicates convergence of an approximated eigenvalue within that tolerance. For all figures, we used 5 steps of GMRES to solve the correction equation in (8). For the left figures, we did not use preconditioning. For the right figures, we preconditioned GMRES with $K = \mathrm{diag}(A) - \tau I$, as in Alg. 4.6

4.7.5 Software Availability

Matlab versions of the algorithms are available from either
`http://www.math.uu.nl/people/sleijpen`
or
`http://www.math.uu.nl/people/vorst`

Fortran implementations of the Jacobi-Davidson method are available via
`http://www.math.uu.nl/people/bomhof`
The Fortran software is of experimental type and does not reflect all insights and possibilities described in this Section.

CHAPTER 5

Generalized Hermitian Eigenproblem

5.6 Jacobi-Davidson methods

G. Sleijpen and H. van der Vorst

Basic Theory

We consider the application of the Jacobi-Davidson approach to the generalized eigenproblem

$$Ax = \lambda Bx, \tag{5.1}$$

with A and B Hermitian and B positive definite. We can, similarly as for the Lanczos method [38], apply the Jacobi-Davidson method to (5.1), using a B-orthogonal basis $v_1, v_2, \ldots, v_m$ for the search subspace, that is

$$V_m^* B V_m = I_m,$$

if we let V_m denote the matrix with columns v_j.

The Ritz-Galerkin condition for vectors $u \equiv V_m s$ in this subspace leads to

$$V_m^* A V_m s - \theta V_m^* B V_m s = 0, \tag{5.2}$$

or, because of the B-orthogonality of V_m:

$$V_m^* A V_m s - \theta s = 0.$$

This leads to Ritz vectors $u_j^{(m)} \equiv V_m s_j$ and Ritz values $\theta_j^{(m)}$. We will assume that these Ritz vectors are normalized with respect to the B-inner product.

The correction equation for the eigenvector component $t_j^{(m)} \perp B u_j^{(m)}$, for the generalized eigenproblem, can be written as

$$\begin{aligned} \left(I - B u_j^{(m)} u_j^{(m)^*}\right) (A - \theta_j^{(m)} B) \left(I - u_j^{(m)} u_j^{(m)^*} B\right) t_j^{(m)} \\ = -r_j^{(m)} \equiv -(A - \theta_j^{(m)} B) u_j^{(m)}. \end{aligned} \tag{5.3}$$

If linear systems with the matrix $A - \theta_j^{(m)} B$ are efficiently solvable, then we can compute the exact solution $t_j^{(m)}$, in other cases we can, as is usual for the Jacobi-Davidson methods, compute approximate solutions to $t_j^{(m)}$ with a Krylov solver applied to (5.3). Note that the operator

$$\left(I - B u_j^{(m)} u_j^{(m)^*}\right) (A - \theta_j^{(m)} B) \left(I - u_j^{(m)} u_j^{(m)^*} B\right)$$

maps the space $(Bu_j^{(m)})^\perp$ onto the space ${u_j^{(m)}}^\perp$, so that preconditioning is required anyhow if we use a Krylov solver in order to get a mapping between $(Bu_j^{(m)})^\perp$ and $(Bu_j^{(m)})^\perp$ (see Remark (8) for Alg. 5.1).

As for the standard Hermitian case, the resulting scheme can be combined with restart and deflation. If we want to work with orthogonal operators in the deflation, then we have to work with B-orthogonal matrices that reduce the given generalized system to Schur form:

$$AQ_k = Z_k D_k,$$

in which $Z_k = BQ_k$, and Q_k is B-orthogonal. The matrix D_k is a diagonal matrix with the k computed eigenvalues values on its diagonal, the columns of Q_k are eigenvectors of A. This leads to skew projections for the deflation with the first k eigenvectors:

$$(I - Z_k Q_k^*)\,(A - \lambda B)\,(I - Q_k Z_k^*)\ .$$

It is easy to verify that the deflated operator B is still symmetric positive definite with respect to the space $(Bu_j^{(m)})^\perp$. We can simply use the B-inner product in that space, since B and the deflated B coincide over that space.
If B is not well-conditioned, that means if x^*By leads to a highly distorted inner product, then we suggest to use the QZ-approach with Jacobi-Davidson (see Section 8.8), which does not exploit symmetry of the involved matrices.
A template for a left-preconditioned Krylov solver is given in Alg. 5.2.

Jacobi-Davidson template with restart and deflation for exterior eigenvalues

ALGORITHM 5.1: **Jacobi-Davidson for $k_{\max}$ exterior eigenvalues**

(1) $\quad t = v_0,\ k = 0,\ m = 0;\ Q = [\,],\ Z = [\,]$
$\quad$ **while** $k < k_{\max}$
(2) $\qquad$ **for** $i = 1, .., m$
$\qquad\quad t = t - (v_i^{B*} t) v_i$
$\qquad m = m + 1; \widetilde{t} = Bt$, norm $= \sqrt{t^* \widetilde{t}}$,
$\qquad v_m = t/\text{norm},\ v_m^A = A v_m,\ v_m^B = \widetilde{t}/\text{norm}$
(3) $\qquad$ **for** $i = 1, .., m$
$\qquad\quad M_{i,m} = v_i^* v_m^A$
(4) $\qquad$ *Compute all eigenpairs* (θ_i, s_i)
$\qquad$ *of the Hermitian matrix* M, $(\|s_i\|_2 = 1)$
$\qquad$ *Sort the pairs:* $|\theta_i - \tau| \geq |\theta_{i-1} - \tau|$
$\qquad u = V s_1,\ p = V^B s_1,\ u^A = V^A s_1,\ r = u^A - \theta_1 p$
(5) $\qquad$ **while** $\|r\|_2 / \|u\|_2 \leq \epsilon$
$\qquad\quad Q = [Q, u],\ Z = [Z, p],\ \widetilde{\lambda}_{k+1} = \theta_1,\ k = k + 1$
$\qquad\quad$ **if** $(k = k_{\max})$, $\widetilde{X} = Q$, **then** *STOP*
$\qquad\quad m = m - 1,\ M = 0$
(6) $\qquad\quad$ **for** $i = 1, .., m$
$\qquad\qquad v_i = V s_{i+1},\ v_i^A = V^A s_{i+1},\ v_i^B = V^B s_{i+1}$
$\qquad\qquad M_{i,i} = \theta_{i+1},\ s_i = e_i,\ \theta_i = \theta_{i+1}$
$\qquad\quad u = v_1,\ p = v_1^B,\ r = v_1^A - \theta_1 p$
(7) $\qquad$ **if** $m > m_{\max}$ **then**
$\qquad\quad M = 0$
$\qquad\quad$ **for** $i = 2, .., m_{\min}$
$\qquad\qquad v_i = V s_i,\ v_i^A = V^A s_i,\ v_i^B = V^B s_i,\ M_{i,i} = \theta_i$
$\qquad\quad v_1 = u,\ v_1^B = p,\ M_{1,1} = \theta_1,\ m = m_{\min}$
$\qquad \theta = \theta_1,\ \widetilde{Q} = [Q, u],\ \widetilde{Z} = [Z, p]$
(8) $\qquad$ *Solve* t $(\perp \widetilde{Z})$ *(approximately) from:*
$\qquad\quad (I - \widetilde{Z}\widetilde{Q}^*)(A - \theta B)(I - \widetilde{Q}\widetilde{Z}^*) t = -r$

To apply this algorithm we need to specify a tolerance ϵ, a target value τ, and a number $k_{\max}$ that specifies how many eigenpairs near τ should be computed. The value of $m_{\max}$ denotes the maximum dimension of the search subspace. If it is exceeded then a restart takes place with a subspace of specified dimension $m_{\min}$. We also need to give a starting vector v_0.

On completion the $k_{\max}$ largest eigenvalues are delivered when τ is chosen larger than $\lambda_{\max}(A)$; the $k_{\max}$ smallest eigenvalues are delivered if τ is chosen smaller than $\lambda_{\min}$. The computed eigenpairs $(\widetilde{\lambda}_j, \widetilde{x}_j)$, $\|\widetilde{x}_j\|_B = 1$, satisfy $\|A\widetilde{x}_j - \widetilde{\lambda}_j B \widetilde{x}_j\|_2 \leq j\epsilon$, where $\widetilde{x}_j$ denotes the jth column of $\widetilde{X}$. The eigenvectors are B-orthogonal: $\widetilde{x}_i^* B \widetilde{x}_j = 0$ for $i \neq j$.

Let us now discuss the different steps of Algorithm 5.1.

(1) Initialization phase.

(2) The new vector t is made B-orthogonal with respect to the current search subspace by means of modified Gram-Schmidt. This can be replaced, for improved numerical stability, by a template as in Alg. 4.4, in which all inner products and norms should be interpreted as B-inner products, B-norms, respectively.

If $m = 0$ then this is an empty loop.

We expand the n by m matrices V, $V^A \equiv AV$, and $V^B \equiv BV$, V denotes the matrix with the current basis vectors v_i for the search subspace as its columns, likewise V^A and V^B.

(3) The mth column of the symmetric matrix $M \equiv V^*AV = V^*V^A$ (of order m) is computed.

(4) The eigenproblem for the m by m matrix M can be solved with a suitable routine for Hermitian dense matrices from LAPACK. We have chosen to compute the standard Ritz values, which makes the algorithm suitable for computing $k_{\max}$ exterior eigenvalues of $A-\lambda B$ close to a specified τ. If eigenvalues in the interior part of the spectrum have to be computed, then the computation of *harmonic Petrov values* is advocated, see Section 8.8.

(5) The stopping criterion is to accept an eigenvector approximation as soon as the norm of the residual (for the normalized vector approximation) is below ϵ. This means that we accept inaccuracies in the order of ϵ^2 in the computed eigenvalues, and inaccuracies (in angle) in the eigenvectors of $\mathcal{O}(\epsilon)$ (provided that the concerned eigenvalue is simple and well separated from the others and B is not ill-conditioned; use [2, Eqn. (4.4)]).

Detection of all wanted eigenvalues cannot be guaranteed, see note (5) for Alg. 4.5.

(6) After acceptance of a Ritz pair, we continue the search for a next pair, with the remaining Ritz vectors as a basis for the initial search space.

(7) We restart as soon as the dimension of the search space for the current eigenvector exceeds $m_{\max}$. The process is restarted with the subspace spanned by the $m_{\min}$ Ritz vectors corresponding to the Ritz values closest to the target value τ.

(8) We have collected the locked (computed) eigenvectors in Q, and the matrix $\widetilde{Q}$ is Q expanded with the current eigenvector approximation u. This is done in order to obtain a more compact formulation; the correction equation in step (8) of Alg. 5.1 is equivalent to the one in equation (5.3). The new correction t has to be orthogonal to the columns of $Z = BQ$ as well as to $p = Bu$.
Of course, the correction equation can be solved by any suitable process, for instance a preconditioned Krylov subspace method that is designed to solve unsymmetric systems. However, because of the skew projections, we always need a preconditioner (which may be the identity operator if nothing else is available) that is deflated by the same skew projections so that we obtain a mapping between $\widetilde{Z}^\perp$ and itself. Because of the occurrence of $\widetilde{Q}$ and $\widetilde{Z}$, one has to be careful with the usage of preconditioners for the matrix $A - \theta B$. The inclusion of preconditioners can be done as in Alg. 5.2. Make sure that the starting vector t_0 for an iterative solver satisfies the orthogonality constraints $\widetilde{Z}^* t_0 = 0$. Note that significant savings per step can be made in Alg. 5.2 if K is kept the same for a (few) Jacobi-Davidson iterations. In that case, columns of $\widehat{Z}$ can be saved from previous steps. Also the matrix $\mathcal{M}$ can be updated from previous steps, as well as its $\mathcal{LU}$ decomposition.
It is not necessary to solve the correction equation very accurately. A strategy, often used for inexact Newton methods [9], here also works well: increase the accuracy with the Jacobi-Davidson iteration step, for instance, solve the correction equation with a residual reduction of $2^{-\ell}$ in the ℓth Jacobi-Davidson iteration (ℓ is reset to 0 when an eigenvector is detected).

For a full theoretical background of this method see [12]. For details on the deflation technique with eigenvectors see also Section 4.7.3.2.

ALGORITHM 5.2: **Approximate solution of the deflated Jacobi-Davidson correction equation, for the generalized Hermitian eigenproblem**

"Solve" with left preconditioner $\widetilde{K} \equiv (I - \widetilde{Z}\widetilde{Q}^*)K(I - \widetilde{Q}\widetilde{Z}^*)$,
for $\widetilde{A} \equiv (I - \widetilde{Z}\widetilde{Q}^*)(A - \theta B)(I - \widetilde{Q}\widetilde{Z}^*)$:

(8a) *Solve* $\widehat{Z}$ *from* $K\widehat{Z} = \widetilde{Z}$
Compute $\mathcal{M} = \widetilde{Z}^*\widehat{Z}$
Decompose $\mathcal{M} = \mathcal{L}\mathcal{U}$
Compute $\widetilde{r} \equiv \widetilde{K}_j^{-1} r$ *as:*
(b′) *solve* $\widehat{r}$ *from* $K\widehat{r} = r$
(c′) $\vec{\gamma} = \widetilde{Z}^*\widehat{r}$
solve $\vec{\beta}$ *from* $\mathcal{L}\vec{\beta} = \vec{\gamma}$
solve $\vec{\alpha}$ *from* $\mathcal{U}\vec{\alpha} = \vec{\beta}$
(d′) $\widetilde{r} = \widehat{r} - \widehat{Z}\vec{\alpha}$
Apply Krylov subspace method with start $t_0 = 0$,
with operator $\widetilde{K}_j^{-1}\widetilde{A}$, *and right-hand side* $-\widetilde{r}$,
$z = \widetilde{K}_j^{-1}\widetilde{A}v$ *for arbitrary* v *is computed as:*
(a) $y = (A - \theta B)v$
(b) *solve* $\widehat{y}$ *from* $K\widehat{y} = y$
(c) $\vec{\gamma} = \widetilde{Z}^*y$
solve $\vec{\beta}$ *from* $\mathcal{L}\vec{\beta} = \vec{\gamma}$
solve $\vec{\alpha}$ *from* $\mathcal{U}\vec{\alpha} = \vec{\beta}$
(d) $z = \widehat{y} - \widehat{Z}\vec{\alpha}$

CHAPTER 7

Non-Hermitian Eigenproblem

7.11 Jacobi-Davidson methods

G. Sleijpen and H. van der Vorst

7.11.1 Generalization of Hermitian Case

Similar to the Lanczos methods and the Arnoldi method, the Jacobi-Davidson method constructs a subspace onto which the given eigenproblem is projected. The subspace is constructed with approximate shift and invert steps, instead of forming a Krylov subspace. In §4.7 we have explained the method in detail, and the generalization to the non-Hermitian case for the basic algorithm, described in Alg. 4.1, is quite straight forward. In fact, the changes are:

1. The construction of the matrix M has to take into account that M is non-Hermitian, hence the corresponding action in part (3) has to be replaced by:

   for $i = 1, .., m-1$
   $\quad M_{i,m} = v_i^* v_m^A$, $M_{m,i} = v_m^* v_i^A$
   $M_{m,m} = v_m^* v_m^A$

2. In (4) a routine has to be selected for the non-Hermitian dense matrix M

If the correction equation (in step (7) of the algorithm) is solved exactly, then the approximate eigenvalues have quadratic convergence towards the eigenvalues of A.

7.11.2 Schur Form and Restart

If we want to include restarts and deflation then matters become more complicated since non-Hermitian matrices do not have orthonormal eigensystems in general. Since we prefer to work with an orthonormal basis for at least the test-subspace, we compute Schur forms of the reduced matrices. Instead of eigenvectors of the matrix A, we compute a partial Schur form $AQ_k = Q_kR_k$, where Q_k is an n by k orthonormal matrix and R_k is k by k upper tridiagonal. A scheme that accommodates for Schur forms is given in Alg. 7.1. This scheme includes the possibility for restart when the dimension of the current subspace exceeds $m_{\max}$.
The scheme computes $k_{\max}$ eigenvalues close to a target τ in the complex plane. Here we have to be necessarily unprecise, since the eigenvalues of a general non-Hermitian matrix are not ordered in the complex plane. Which Ritz values are delivered as close eigenvalues depends amongst others on the angles of the corresponding eigenvectors. Usually, the selection from Ritz values is

appropriate if τ is chosen somewhere at the exterior of the distribution of eigenvalues. If we want eigenvalues of M close to some interior point then the scheme may be much less satisfactory, and we propose to work in such cases with harmonic Ritz values. A scheme for interior eigenvalues will be presented in §7.11.3.

ALGORITHM 7.1: **Jacobi-Davidson for $k_{\max}$ exterior eigenvalues**

(1) $t = v_0,\ k = 0,\ m = 0;\ Q = [\,],\ R = [\,]$
    while $k < k_{\max}$
(2)   **for** $i = 1, .., m$
        $t = t - (v_i^* t) v_i$
      $m = m + 1,\ v_m = t/\|t\|_2,\ v_m^A = A v_m$
(3)   **for** $i = 1, .., m - 1$
        $M_{i,m} = v_i^* v_m^A,\ M_{m,i} = v_m^* v_i^A$
      $M_{m,m} = v_m^* v_m^A$
(4)   *Make a Schur decomposition* $M = STS^*$,
      S *unitary and* T *upper triangular,*
      such that: $|T_{i,i} - \tau| \le |T_{i+1,i+1} - \tau|$
      $u = V s_1,\ u^A = V^A s_1,\ \theta = T_{1,1},\ r = u^A - \theta u,\ \widetilde{a} = Q^* r,\ \widetilde{r} = r - Q\widetilde{a}$
(5)   **while** $\|\widetilde{r}\|_2 \le \epsilon$
        $R = \begin{pmatrix} R & \widetilde{a} \\ 0 & \theta \end{pmatrix},\ Q = [Q, u],\ k = k + 1$
        if $k = k_{\max}$ **then** *STOP*
        $m = m - 1$
(6)   **for** $i = 1, .., m$
          $v_i = V s_{i+1},\ v_i^A = V^A s_{i+1},\ s_i = e_i$
        $M =$ *lower* m *by* m *block of* T
        $u = v_1,\ \theta = M_{1,1},\ r = v_1^A - \theta u,\ \widetilde{a} = Q^* r,\ \widetilde{r} = r - Q\widetilde{a}$
(7)   **if** $m \ge m_{\max}$ **then**
        for $i = 2, .., m_{\min}$
          $v_i = V s_i,\ v_i^A = V^A s_i$
        $M =$ *leading* $m_{\min}$ *by* $m_{\min}$ *block of* T
        $v_1 = u,\ v_1^A = u^A,\ m = m_{\min}$
(8) $\widetilde{Q} = [Q, u]$, *Solve* $t\ (\perp \widetilde{Q})$ *(approximately) from:*
        $(I - \widetilde{Q}\widetilde{Q}^*)(A - \theta I)(I - \widetilde{Q}\widetilde{Q}^*) t = -\widetilde{r}$

To apply this algorithm we need to specify a starting vector v_0, a tolerance ϵ, a target value τ, and a number $k_{\max}$ that specifies how many eigenpairs near τ should be computed. The value of $m_{\max}$ denotes the maximum dimension of the search subspace. If it is exceeded then a restart takes place with a subspace of specified dimension $m_{\min}$.

On completion the $k_{\max}$ eigenvalues close to τ are delivered, and the corresponding reduced Schur form $AQ = QR$, where Q is n by $k_{\max}$ orthogonal and R is $k_{\max}$ by $k_{\max}$ upper triangular. The eigenvalues are on the diagonal of R. The computed Schur form satisfies $\|Aq_j - QRe_j\|_2 \le j\epsilon$, where q_j is the jth column of Q.

We will now discuss the components of the algorithm.

(1) Initialization phase.

(2) The new vector t is made orthogonal with respect to the current search subspace by means of modified Gram-Schmidt. This can be replaced, for improved numerical stability, by the template, for iterated modified Gram-Schmidt, given in Alg. 4.4.

If $m = 0$ then this is an empty loop.

If the angle between the new vector t and the search subspace is very small, then the resulting vector v_j has little significance and the method practically stagnates. A random vector t helps to overcome this stagnation. A more sophisticated strategy can be found in [14].

(3) We compute the last column and row of the dense matrix $M \equiv V^*AV = V^*V^A$ (of order m); $V^A \equiv AV$. The matrix V denotes the n by m matrix with columns v_j, V^A likewise.

(4) The Schur reduction for the m by m matrix M can be solved by a standard solver for dense eigenproblems. We have chosen to compute the standard Ritz values, which makes the algorithm suitable for computing $k_{\max}$ exterior eigenvalues of A close to a specified τ. If eigenvalues in the interior part of the spectrum have to be computed, then the usage of harmonic Ritz values is advocated, see §7.11.3.

In each step, the Schur form has to be sorted such that $T_{1,1}$ is closest to τ. Only if $m \geq m_{\max}$ the sorting of the Schur form has to be be such that all of the $m_{\max}$ leading diagonal elements of T are closest to τ. For ease of presentation we sorted all diagonal elements here.

For a template of an algorithm for the sorting of a Schur form, see [39, 40, 3] and [11, Chap. 6B].

S is the m by m matrix with columns s_j.

(5) The stopping criterion is to accept a Schur vector approximation as soon as the norm of the residual (for the normalized Schur vector approximation) is below ϵ. This means that we accept inaccuracies in the order of ϵ in the computed eigenvalues, and inaccuracies (in angle) in the Schur vectors of $\mathcal{O}(\epsilon)$ (provided that the concerned eigenvalue is simple and well separated from the others).

Detection of all wanted eigenvalues cannot be guaranteed, see note (5) for Alg. 4.5.

If the matrix is real, then all eigenpairs are either real or appear in complex conjugate pairs. If a complex eigenpair has been detected, then its conjugate is known and can be deflated as well. This makes the algorithm more efficient.

(6) After acceptance of a Ritz pair, we continue the search for a next pair, with the remaining Ritz vectors as a basis for the initial search space.

(7) We restart as soon as the dimension of the search space for the current Schur vector exceeds $m_{\max}$. The process is restarted with the subspace spanned by the $m_{\min}$ Ritz vectors corresponding to the Ritz values closest to the target value τ.

(8) We have collected the locked (computed) Schur vectors in Q, and the matrix $\widetilde{Q}$ is Q expanded with the current Schur vector approximation u. This is done in order to obtain a more compact formulation; the correction equation is equivalent to the one in (4.3). The new correction t has to be orthogonal to the columns of Q as well as to u.

Of course, the correction equation can be solved by any suitable process, for instance a preconditioned Krylov subspace method that is designed to solve unsymmetric systems. Because of the occurrence of $\widetilde{Q}$ one has to be careful with the usage of preconditioners for the matrix $A - \theta I$. The inclusion of preconditioners can be done following the same principles as for the single vector Jacobi-Davidson algorithm, see Alg. 4.6 for a Template. Make sure that the starting vector t_0 for an iterative solver satisfies the orthogonality constraints $\widetilde{Q}^* t_0 = 0$. Note that significant savings per step can be made in Alg. 4.6 if K is kept the same for a (few) Jacobi-Davidson iterations. In that case columns of $\widehat{Q}$ can be saved from previous steps. Also the matrix $\mathcal{M}$ can be updated from previous steps, as well as its $\mathcal{LU}$ decomposition.

It is not necessary to solve the correction equation very accurately. A strategy, often used for inexact Newton methods [9], here also works well: increase the accuracy with the Jacobi-Davidson iteration step, and, for instance, solve the correction equation with a residual reduction of $2^{-\ell}$ in the ℓ-th Jacobi-Davidson iteration (ℓ is reset to 0 when a Schur vector is detected).

In particular, in the first few initial steps, the approximate eigenvalue θ may be very inaccurate and then it does not make sense to solve the correction equation accurately. In this stage it can be more effective to temporarily replace θ by τ or to take $t = -r$ for the expansion of the search subspace [22, 12].

For deflation see 8.8.2.1, with Z_j replaced by Q_j and B by I. For a full theoretical background of this method, as well as details on the deflation technique with Schur vectors, see [12].

7.11.3 Interior Eigenvalues

7.11.3.1 The usage of harmonic Ritz values for interior eigenvalues

For restart purposes, specially if one is heading for interior eigenvalues, the harmonic Ritz vectors have been advocated for symmetric matrices [21], see also §4.7.4.

The concept of harmonic Ritz values [24] is easily generalized for unsymmetric matrices [32]. As we have seen, the Jacobi-Davidson methods generate basis vectors $v_1, \ldots, v_m$, for a subspace $\mathcal{V}_m$. For the projection of A onto this subspace we have to compute the vectors $w_j \equiv Av_j$. The inverses of the Ritz values of A^{-1}, with respect to the subspace spanned by the w_j, are called the *harmonic Ritz values*. The harmonic Ritz values can be computed without inverting A, since a harmonic Ritz pair $(\widetilde{\theta}_j^{(m)}, \widetilde{u}_j^{(m)})$ satisfies

$$A\widetilde{u}_j^{(m)} - \widetilde{\theta}_j^{(m)}\widetilde{u}_j^{(m)} \perp \mathcal{W}_m \equiv \operatorname{span}(w_1, \ldots, w_m), \tag{7.1}$$

for $\widetilde{u}_j^{(m)} \in V_m$ and $\widetilde{u}_j^{(m)} \neq 0$. This implies that the harmonic Ritz values are the eigenvalues of the pencil $(W_m^* A V_m, W_m^* V_m)$:

$$W_m^* A V_m s_j^{(m)} - \widetilde{\theta}_j^{(m)} W_m^* V_m s_j^{(m)} = 0$$

The exterior standard Ritz values usually converge to exterior eigenvalues of A. Likewise, the interior harmonic Ritz values for the shifted matrix $A - \tau I$ usually converge towards eigenvalues $\lambda \neq \tau$ closest to the shift τ. Fortunately, the search subspaces $\mathcal{V}_m$ for the shifted matrix and the unshifted matrix coincide, which facilitates the computation of harmonic Ritz pairs. For reasons of stability we construct both V_m and W_m to orthonormal: W_m is such that $(A - \tau I)V_m = W_m M_m^A$ where M_m^A is m by m upper triangular.

In the resulting scheme we compute a (partial) Schur decomposition rather than a partial eigendecomposition. That is, we wish to compute vectors $q_1, \ldots, q_k$, such that $AQ_k = Q_k R_k$, with $Q_k^* Q_k = I_k$ and R_k is a k by k upper triangular matrix. The diagonal elements of R_k represent eigenvalues of A, and the corresponding eigenvectors of A can be computed from Q_k and R_k.

7.11.3.2 A template for Jacobi-Davidson for interior eigenvalues

An algorithm for Jacobi-Davidson, with partial reduction to Schur form, and based on harmonic Ritz values and vectors, is given in Alg. 7.2. The algorithm includes restart and deflation techniques. It can be used for the computation of a number of eigenvalues close to τ.

Algorithm 7.2: **Jacobi-Davidson for $k_{\max}$ interior eigenvalues close to τ**

(1) $t = v_0,\ k = 0,\ m = 0;\ Q = [\,],\ R = [\,]$
  while $k < k_{\max}$
(2)   **for** $i = 1, .., m$
      $t = t - (v_i^* t) v_i$
    $m = m + 1,\ v_m = t/\|t\|_2,\ v_m^A = A v_m - \tau v_m,\ w = v_m^A$
    for $i = 1, .., k$
      $w = w - (q_i^* w) q_i$
    for $i = 1, .., m-1$
      $M_{i,m}^A = w_i^* w,\ w = w - M_{i,m}^A w_i$
    $M_{m,m}^A = \|w\|_2,\ w_m = w / M_{m,m}^A$
(3)   **for** $i = 1, .., m-1$
      $M_{i,m} = w_i^* v_m,\ M_{m,i} = w_m^* v_i$
    $M_{m,m} = w_m^* v_m$
(4)   *Make a QZ decomposition* $M^A S^R = S^L T^A,\ M S^R = S^L T$,
    $S^R,\ S^L$ *unitary and* $T^A,\ T$ *upper triangular,*
    such that: $|T_{i,i}^A / T_{i,i}| \le |T_{i+1,i+1}^A / T_{i+1,i+1}|$
    $u = V s_1^R,\ u^A = V^A s_1^R,\ \vartheta = \overline{T_{1,1}} \cdot T_{1,1}^A,$
    $r = u^A - \vartheta u,\ \widetilde{a} = Q^* r,\ \widetilde{r} = r - Q\widetilde{a}$
(5)   **while** $\|\widetilde{r}\|_2 \le \epsilon$
      $R = \begin{pmatrix} R & \widetilde{a} \\ 0 & \vartheta + \tau \end{pmatrix},\ Q = [Q, u],\ k = k + 1$
      if $k = k_{\max}$ **then** *STOP*
(6) $m = m - 1$
      for $i = 1, .., m$
        $v_i = V s_{i+1}^R,\ v_i^A = V^A s_{i+1}^R,\ w_i = W s_{i+1}^L,\ s_i^R = s_i^L = e_i$
      $M^A,\ M$ *is the lower m by m block of* $T^A,\ T$, *resp.*
      $u = v_1,\ u^A = v_1^A,\ \vartheta = \overline{M_{1,1}} \cdot M_{1,1}^A$
      $r = u^A - \vartheta u,\ \widetilde{a} = Q^* r,\ \widetilde{r} = r - Q\widetilde{a}$
(7)   **if** $m \ge m_{\max}$ **then**
      for $i = 2, .., m_{\min}$
        $v_i = V s_i^R,\ v_i^A = V^A s_i^R,\ w_i = W s_i^L$
      $M^A,\ M$ *is the leading $m_{\min}$ by $m_{\min}$ block of* $T^A,\ T$, *resp.*
      $v_1 = u,\ v_1^A = u^A,\ w_1 = W s_1^L,\ m = m_{\min}$
(8) $\theta = \vartheta + \tau,\ \widetilde{Q} = [Q, u]$
    Solve $t\ (\perp \widetilde{Q})$ *(approximately) from:*
      $(I - \widetilde{Q}\widetilde{Q}^*)(A - \theta I)(I - \widetilde{Q}\widetilde{Q}^*) t = -\widetilde{r}$

To apply this algorithm we need to specify a starting vector v_0, a tolerance ϵ, a target value τ, and a number $k_{\max}$ that specifies how many eigenpairs near τ should be computed. The value of $m_{\max}$ denotes the maximum dimension of the search subspace. If it is exceeded then a restart takes place with a subspace of specified dimension $m_{\min}$.

On completion the $k_{\max}$ eigenvalues close to τ are delivered, and the corresponding reduced Schur form $AQ = QR$, where Q is n by $k_{\max}$ orthogonal and R is $k_{\max}$ by $k_{\max}$ upper triangular. The eigenvalues are on the diagonal of R. The computed Schur form satisfies $\|Aq_j - QRe_j\|_2 \le j\epsilon$, where q_j is the jth column of Q.

We will comment on some parts of the algorithm in view of our discussions in previous subsections.

(1) Initialization phase.

(2) The newly computed correction is made orthogonal with respect to the current search subspace by means of modified Gram-Schmidt. We have chosen to store also the matrix $V^A \equiv AV - \tau V$; v_m^A is the expansion vector for this matrix. The expansion vector for W is obtained by making v_m^A orthogonal with respect to the space of detected Schur vectors and with respect to the current test subspace by means of modified Gram-Schmidt. The Gram-Schmidt steps can be replaced, for improved numerical stability, by the template given in Alg. 4.4.

V denotes the n by m matrix with columns v_j, likewise W and V^A.

(3) The values $M_{i,j}$ represent elements of the square m by m matrix $M \equiv W^*V$. The values $M_{i,j}^A$ represent elements of the m by m upper triangular $M^A \equiv W^*V^A$.

(Note that $V^A = WM^A + QF$, if $F \equiv Q^*V^A$. Therefore, V^A can be reconstructed from W, M^A, Q and F. In particular, r can be computed from these quantities. Instead of storing the n-dimensional matrix V^A, it suffices to store the k by m matrix F (of elements q_i^*w, computed in (2)). This approach saves memory space. However, for avoiding instabilities, the deflation procedure needs special attention.)

(4) At this point the QZ decomposition (generalized Schur form) of the matrix pencil (M^A, M) has to be be computed: $M^AS^R = S^LT^A$ and $MS^R = S^LT$, where S^R and S^L are unitary and T^A and T are upper triangular. This can be done with a suitable routine for dense matrix pencils dense matrices from LAPACK.

In each step, the Schur form has to be sorted such that $|T_{1,1}^A/T_{1,1}|$ is smallest. Only if $m \geq m_{\max}$ the sorting of the Schur form has to be be such that all of the $m_{\max}$ leading diagonal elements of T^A and T represent the $m_{\max}$ harmonic Ritz values closest to 0. For ease of presentation we sorted all diagonal elements here.

For an algorithm of the sorting of a generalized Schur form, see [39, 40, 3] and [11, Chap. 6B].

The value of θ needs some attention. We have chosen to compute the Rayleigh quotient ϑ (instead of the harmonic Ritz value) corresponding to the harmonic Ritz vector u (see [33]). The Rayleigh quotient follows from the requirement that $(A - \tau I)u - \vartheta u \perp u$ instead of $\perp W$: then $r \perp u$.

$\overline{T_{1,1}}$ denotes the complex conjugate of $T_{1,1}$. The matrix S^R is m by m with columns s_j^R, likewise S^L.

(5) The stopping criterion is to accept a Schur vector approximation as soon as the norm of the residual (for the normalized Schur vector approximation) is below ϵ. This means that we accept inaccuracies in the order of ϵ in the computed eigenvalues, and inaccuracies (in angle) in the eigenvectors of $\mathcal{O}(\epsilon)$ (provided that the concerned eigenvalue is simple and well separated from the others and the left and right eigenvector have a small angle).

Detection of all wanted eigenvalues cannot be guaranteed, see note (5) for Alg. 4.5.

(6) This is a restart after acceptance of an approximate eigenpair.

(7) At this point we have a restart when the dimension of the subspace exceeds $m_{\max}$. After a restart the Jacobi-Davidson iterations are resumed with a subspace of dimension $m_{\min}$.

(8) The deflation with computed eigenvectors is represented by the factors with Q. The matrix Q has the converged eigenvectors as its columns. If a left preconditioner K is available for the operator $A - \theta I$, then with a Krylov solver similar reductions are realizable as in the situation for exterior eigenvalues. A template for the efficient handling of the left-preconditioned operator is given in Alg. 4.6.

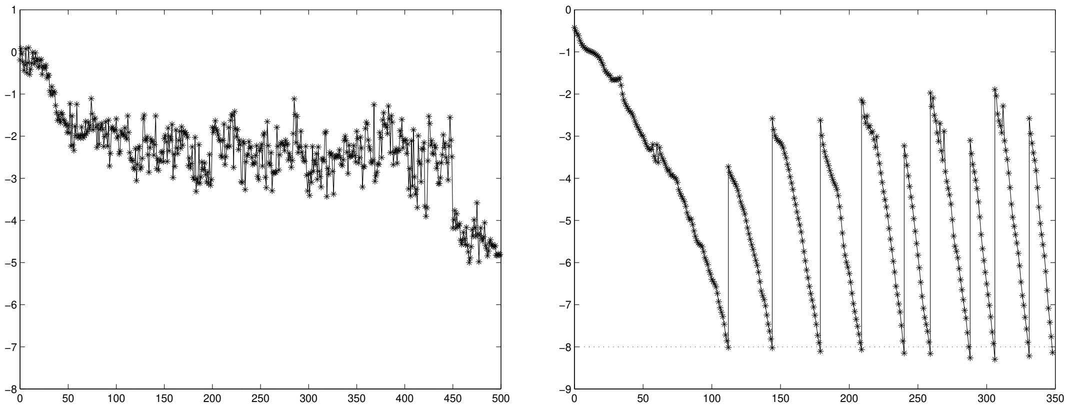

Figure 7.1: Jacobi-Davidson for exterior eigenvalues (left figure) and interior eigenvalues (right figure).

7.11.3.3 Numerical example

We discuss the results for a small example that can be easily repeated.

The unsymmetric matrix A is of dimension $n = 100$ and it is tridiagonal. The diagonal entries are $a_{i,i} \equiv -2$, the codiagonal entries are $a_{i,i-1} \equiv 1$, $a_{i,i+1} \equiv 1.2$.

We use Alg. 7.1 and Alg. 7.2, for the computation of the $k_{\max} = 10$ eigenvalues closest to the target value $\tau = -2 + 0.1i$. Since these eigenvalues are located in the interior of the spectrum, we expect some advantage of Alg. 7.2, which is designed for interior eigenvalues. Indeed, as we will see, the usage of harmonic Ritz values leads to an advantage here.

We have carried out the experiments in Matlab. The input parameters have been chosen as follows. The starting vector v has been chosen with random entries (with seed = 0 in Matlab). The tolerance is $\epsilon = 10^{-8}$. The subspace dimension parameters are $m_{\min} = 10$, $m_{\max} = 15$. The correction equations have been solved approximately with 5 steps of GMRES.

We show graphically in Fig. 7.1 the norm of the residual vector as a function of the iteration number. Each time when the norm is less than ϵ, then we have determined an eigenvalue in acceptable approximation, and the iteration is continued with deflation for the next eigenvalue. The left picture represents the results obtained with Alg. 7.1, and we see that there is no convergence detected within 500 Jacobi-Davidson steps. In the right picture we see the results for Alg. 7.2, and now 10 eigenvalues have been discovered within 350 iterations.

7.11.4 Software Availability

Matlab versions of the algorithms are available from either
`http://www.math.uu.nl/people/sleijpen`
or
`http://www.math.uu.nl/people/vorst`

Fortran implementations of the Jacobi-Davidson method are available via
`http://www.math.uu.nl/people/bomhof`
The Fortran software is of experimental type and does not reflect all insights and possibilities described in this section.

CHAPTER 8

Generalized Non-Hermitian Eigenproblem

8.8 Jacobi-Davidson method

G. Sleijpen and H. van der Vorst

8.8.1 Basic Theory

Similar as for the generalized Hermitian eigenproblem, we want to avoid transformation of $Ax = \lambda Bx$ to a standard eigenproblem. This would require the solution of some linear system, involving B and/or A, per iteration step. Furthermore, for stability reasons we want to work with orthonormal transformations and for this reason our goal is to compute Schur vectors for the pencil $A - \lambda B$, rather than eigenvectors. This leads to a generalization of the Jacobi-Davidson algorithm, in which we compute a partial Schur form for the pencil. This algorithm can be interpreted as a subspace iteration variant of the QZ-algorithm. A consequence of this approach is that we have to work with search and test spaces that are different.

With $\lambda = \alpha/\beta$, the generalized eigenproblem $Ax = \lambda Bx$ is equivalent to the eigenproblem

$$(\beta A - \alpha B)x = 0, \tag{8.1}$$

and we denote a generalized eigenvalue of the matrix pair (A, B) as a pair (α, β). This approach is preferred, because underflow or overflow for $\lambda = \alpha/\beta$ in finite precision arithmetic may occur when α and/or β are zero or close to zero, in which case the pair is still well-defined [20], [35, Ch.VI], [28].

A *partial generalized Schur form* of dimension k for a matrix pair (A, B) is the decomposition

$$AQ_k = Z_k R_k^A, \quad BQ_k = Z_k R_k^B, \tag{8.2}$$

where Q_k and Z_k are orthogonal n by k matrices, and R_k^A and R_k^B are upper triangular k by k matrices. A column q_i of Q_k is referred to as a *generalized Schur vector*, and we refer to a pair $((\alpha_i, \beta_i), q_i)$, with $(\alpha_i, \beta_i) = (R_k^A(i,i), R_k^B(i,i))$ as a *generalized Schur pair.*
It follows that if $((\alpha, \beta), y)$ is a generalized eigenpair of (R_k^A, R_k^B) then $((\alpha, \beta), Q_k y)$ is a generalized eigenpair of (A, B).

We will now describe Jacobi-Davidson for the generalized eigenproblem (8.1).
From the relations (8.2) we conclude that

$$\beta_i A q_i - \alpha_i B q_i \perp z_i,$$

which suggests that we should follow a Petrov Galerkin condition for the construction of reduced systems. In each step the approximate eigenvector u is selected from a *search subspace* span(V).

We require that the residual $\eta Au - \zeta Bu$ is orthogonal to some other well-chosen *test subspace* span(W):

$$\eta\, Au - \zeta\, Bu \perp \mathrm{span}(W). \tag{8.3}$$

Both subspaces are of the same dimension, say m. Equation (8.3) leads to the *projected eigenproblem*

$$(\eta\, W^*AV - \zeta\, W^*BV)\, s = 0. \tag{8.4}$$

The pencil $\eta\, W^*AV - \zeta\, W^*BV$ can be reduced by the QZ algorithm [20] to generalized Schur form (note that this is a m-dimensional problem). This leads to orthogonal m by m matrices S^R and S^L, and upper triangular m by m matrices T^A and T^B, such that

$$(S^L)^*(W^*AV)S^R = T^A \quad \text{and} \quad (S^L)^*(W^*BV)S^R = T^B. \tag{8.5}$$

The decomposition can be reordered such that the first column of S^R and the $(1,1)$-entries of T^A and T^B represent the wanted Petrov solution [12].
With $s \equiv s_1^R \equiv S^R e_1$ and $\zeta \equiv T^A_{1,1}$, $\eta \equiv T^B_{1,1}$, the *Petrov vector* is defined as $u \equiv Vs$, for the associated *generalized Petrov value* (ζ, η). In our algorithm we will construct orthogonal matrices V and W, so that also $\|u\|_2 = 1$.
With the decomposition in (8.5), we construct an approximate partial generalized Schur form (cf. (8.2)): VS^R approximates a Q_k, and WS^L approximates the associated Z_k. Since span$(Z_k) =$ span$(AQ_k) =$ span(BQ_k) (cf. (8.2)), this suggests to choose W such that span(W) coincides with span$(\nu_0 AV + \mu_0 BV)$. With the weights ν_0 and μ_0 we can influence the convergence of the Petrov values. If we want eigenpair approximations for eigenvalues λ close to a target τ, then the choice

$$\nu_0 = 1/\sqrt{1+|\tau|^2} \quad \mu_0 = -\tau\nu_0$$

is very effective [12], especially if we want to compute eigenvalues in the interior of the spectrum of $A - \lambda B$. We will call the Petrov approximations for this choice the *harmonic Petrov* eigenpairs.
The Jacobi-Davidson correction equation for the component $t \perp u$, for the pencil $\eta A - \zeta B$ becomes:

$$\left(I - \frac{pp^*}{p^*p}\right)(\eta A - \zeta B)\,(I - uu^*)\, t = -r, \tag{8.6}$$

with $r \equiv \eta Au - \zeta Bu$, and $p \equiv \nu_0 Au + \mu_0 Bu$. It can be shown that if (8.6) is solved exactly, the convergence to the generalized eigenvalue will be quadratic, see [30, Th. 3.2].
Usually, this correction equation is solved only approximately, for instance, with a (preconditioned) iterative solver. The obtained vector t is used for the expansion v of V and $\nu_0 Av + \mu_0 Bv$ is used for the expansion of W. For both spaces we work with orthonormal bases. Therefore, the new columns are orthonormalized with respect to the current basis by modified Gram-Schmidt (see Section 4.7.1).

It can be shown that, with the above choices for p and W,

$$p = W s_1^L = W S^L e_1. \tag{8.7}$$

The relation between the partial generalized Schur form for the given large problem and the complete generalized Schur form for the reduced problem (8.4) via right vectors ($u = V s_1^R$) is similar to the relation via left vectors ($p = W s_1^L$). This can also be exploited for restarts.

8.8.2 Deflation and Restart

Similar to the situation for the standard eigenproblem, see Chapters 4.7.3 and 7.11.2, the Jacobi-Davidson process can be enhanced with restart possibilities in order to restrict the dimension of the search subspace. The process can also be combined with deflation in order to find a number of different generalized Schur pairs.

8.8.2.1 Deflation

The partial generalized Schur form can be obtained in a number of successive steps. Suppose that we have the partial generalized Schur form $AQ_{k-1} = Z_{k-1}R^A_{k-1}$ and $BQ_{k-1} = Z_{k-1}R^B_{k-1}$. We want to expand this partial generalized Schur form with the new right Schur vector q and the left Schur vector z to

$$\begin{aligned} A\,[Q_{k-1}, q] &= [Z_{k-1}, z] \begin{pmatrix} R^A_{k-1} & a \\ 0 & \alpha \end{pmatrix} \\ \text{and} \quad B\,[Q_{k-1}, q] &= [Z_{k-1}, z] \begin{pmatrix} R^B_{k-1} & b \\ 0 & \beta \end{pmatrix}. \end{aligned} \tag{8.8}$$

The new generalized Schur pair $((\alpha, \beta), q)$ satisfies

$$Q^*_{k-1} q = 0 \quad \text{and} \quad (\beta A - \alpha B) q - Z_{k-1}(\beta a - \alpha b) = 0, \tag{8.9}$$

or, since $\beta a - \alpha b = Z^*_{k-1}(\beta A - \alpha B) q$,

$$Q^*_{k-1} q = 0 \quad \text{and} \quad \left(I - Z_{k-1} Z^*_{k-1}\right)(\beta A - \alpha B)\left(I - Q_{k-1} Q^*_{k-1}\right) q = 0. \tag{8.10}$$

The vectors a and b can be computed from

$$a = Z^*_{k-1} A q \quad \text{and} \quad b = Z^*_{k-1} B q. \tag{8.11}$$

Hence, the generalized Schur pair $((\alpha, \beta), q)$ is an eigenpair of the deflated matrix pair

$$\begin{aligned} \big(\left(I - Z_{k-1} Z^*_{k-1}\right) A \left(I - Q_{k-1} Q^*_{k-1}\right)&, \\ \left(I - Z_{k-1} Z^*_{k-1}\right) B \left(I - Q_{k-1} Q^*_{k-1}\right)&\big) \end{aligned} \tag{8.12}$$

This eigenproblem can be solved again with the Jacobi-Davidson process that we have outlined in Section 8.8.1. In that process we construct vectors v_i that are orthogonal to Q_{k-1} and vectors w_i that are orthogonal to Z_{k-1}. This simplifies the computation of the interaction matrices M^A and M^B, associated with the deflated operators:

$$\left\{ \begin{aligned} M^A &\equiv W^* \left(I - Z_{k-1} Z^*_{k-1}\right) A \left(I - Q_{k-1} Q^*_{k-1}\right) V = W^* A V \\ M^B &\equiv W^* \left(I - Z_{k-1} Z^*_{k-1}\right) B \left(I - Q_{k-1} Q^*_{k-1}\right) V = W^* B V \end{aligned} \right. \tag{8.13}$$

and M^A and M^B can be simply computed as W^*AV and W^*BV, respectively.

8.8.2.2 Restart

Suppose that the generalized Schur form (8.5) is ordered with respect to τ such that

$$|T^A_{1,1}/T^B_{1,1} - \tau| \leq |T^A_{2,2}/T^B_{2,2} - \tau| \leq \ldots \leq |T^A_{m,m}/T^B_{m,m} - \tau|, \tag{8.14}$$

where m is the dimension of span(V). Then, for $i < m$, the space span$(Vs^R_1, \ldots, Vs^R_i)$ spanned by the first i columns of VS^R contains the i most promising Petrov vectors. The corresponding test subspace is given by span$(Ws^L, \ldots, Ws^L_i)$.
Therefore, in order to reduce the dimension of the subspaces ("implicit restart") to $m_{\min}$, $m_{\min} < m$, the columns $v_{m_{\min}+1}$ through v_m, and $w_{m_{\min}+1}$ through w_m can simply be discarded and the Jacobi-Davidson algorithm can be continued with

$$V = [Vs^R_1, \ldots, Vs^R_{m_{\min}}] \quad \text{and} \quad W = [Ws^L_1, \ldots, Ws^L_{m_{\min}}]. \tag{8.15}$$

8.8.3 A template for Jacobi-Davidson QZ for interior eigenvalues

The Jacobi-Davidson algorithm is given in Alg. 8.1. This algorithm attempts to compute the generalized Schur pairs $((\alpha, \beta), q)$, for which the ratio β/α is closest to a specified target value τ in the complex plane. The algorithm includes restart in order to limit the dimension of the search space, and deflation with already converged left and right Schur vectors.

ALGORITHM 8.1: **Jacobi-Davidson QZ for $k_{\max}$ interior eigenvalues close to τ**

(1) $t = v_0$, $k = 0$, $\nu_0 = 1/\sqrt{1 + |\tau|^2}$, $\mu_0 = -\tau\nu_0$, $m = 0$;
  $Q = [\,]$, $Z = [\,]$, $S = [\,]$, $T = [\,]$
  while $k < k_{\max}$
(2)   **for** $i = 1, .., m$
      $t = t - (v_i^* t) v_i$
    $m = m + 1$, $v_m = t/\|t\|_2$, $v_m^A = A v_m$, $v_m^B = B v_m$, $w = \nu_0 v_m^A + \mu_0 v_m^B$
    for $i = 1, .., k$
      $w = w - (z_i^* w) z_i$
    for $i = 1, .., m - 1$
      $w = w - (w_i^* w) w_i$
    $w_m = w/\|w\|_2$
(3)   **for** $i = 1, .., m - 1$
      $M_{i,m}^A = w_i^* v_m^A$, $M_{m,i}^A = w_m^* v_i^A$, $M_{i,m}^B = w_i^* v_m^B$, $M_{m,i}^B = w_m^* v_i^B$
    $M_{m,m}^A = w_m^* v_m^A$, $M_{m,m}^B = w_m^* v_m^B$
(4)   *Make a QZ decomposition:* $M^A S^R = S^L T^A$, $M^B S^R = S^L T^B$
    S^R, S^L *unitary and* T^A, T^B *upper triangular,*
    such that: $|T_{i,i}^A / T_{i,i}^B - \tau| \leq |T_{i+1,i+1}^A / T_{i+1,i+1}^B - \tau|$
    $u = V s_1^R$, $p = W s_1^L$, $u^A = V^A s_1^R$, $u^B = V^B s_1^R$, $\zeta = T_{1,1}^A$, $\eta = T_{1,1}^B$
    $r = \eta u^A - \zeta u^B$, $\widetilde{a} = Z^* u^A$, $\widetilde{b} = Z^* u^B$, $\widetilde{r} = r - Z(\eta \widetilde{a} - \zeta \widetilde{b})$
(5)   **while** $\|\widetilde{r}\|_2 \leq \epsilon$
      $R^A = \begin{pmatrix} R^A & \widetilde{a} \\ 0 & \zeta \end{pmatrix}$, $R^B = \begin{pmatrix} R^B & \widetilde{b} \\ 0 & \eta \end{pmatrix}$
      $Q = [Q, u]$, $Z = [Z, p]$, $k = k + 1$
      if $k = k_{\max}$ **then** *STOP*
(6) $m = m - 1$
      for $i = 1, .., m$
        $v_i = V s_{i+1}^R$, $v_i^A = V^A s_{i+1}^R$, $v_i^B = V^B s_{i+1}^R$,
        $w_i = W s_{i+1}^L$, $s_i^R = s_i^L = e_i$
      M^A, M^B *is the lower* m *by* m *block of* T^A, T^B, *resp.*
      $u = v_1$, $p = w_1$, $u^A = v_1^A$, $u^B = v_1^B$, $\zeta = T_{1,1}^A$, $\eta = T_{1,1}^B$
      $r = \eta u^A - \zeta u^B$, $\widetilde{a} = Z^* u^A$, $\widetilde{b} = Z^* u^B$, $\widetilde{r} = r - Z(\eta \widetilde{s} - \zeta \widetilde{t})$
(7)   **if** $m \geq m_{\max}$ **then**
      for $i = 2, .., m_{\min}$
        $v_i = V s_i^R$, $v_i^A = V^A s_i^R$, $v_i^B = V^B s_i^R$, $w_i = W s_i^L$
      M^A, M^B *is the leading* $m_{\min}$ *by* $m_{\min}$ *block of* T^A, T^B, *resp.*
      $v_1 = u$, $v_1^A = u^A$, $v_1^B = u^B$, $w_1 = p$, $m = m_{\min}$
(8) $\widetilde{Q} = [Q, u]$, $\widetilde{Z} = [Z, p]$
    Solve t $(\perp \widetilde{Q})$ *(approximately) from:*
      $(I - \widetilde{Z}\widetilde{Z}^*)(\eta A - \zeta B)(I - \widetilde{Q}\widetilde{Q}^*) t = -\widetilde{r}$

To apply this algorithm we need to specify a starting vector v_0, a tolerance ϵ, a target value τ, and a number $k_{\max}$ that specifies how many eigenpairs near τ should be computed. The value of $m_{\max}$ specifies the maximum dimension of the search subspace. If it is exceeded then a restart takes place with a subspace of dimension $m_{\min}$.

On completion the $k_{\max}$ generalized eigenvalues close to τ are delivered, and the corresponding reduced Schur form $AQ = ZR^A$, $BQ = ZR^B$, where Q and Z are n by $k_{\max}$ orthogonal and R^A, R^B are $k_{\max}$ by $k_{\max}$ upper triangular. The generalized eigenvalues are the on diagonals of R^A and R^B. The computed form satisfies $\|Aq_j - ZR^A e_j\|_2 = \mathcal{O}(\epsilon)$, $\|Bq_j - ZR^B e_j\|_2 = \mathcal{O}(\epsilon)$, where $q_j$is the jth column of Q.

The accuracy of the computed reduced Schur form depends on the distance between the target value τ and the eigenvalue $(\alpha_j, \beta_j) \equiv (R^A_{j,j}, R^B_{j,j})$. If we neglect terms of order machine precision and of order ϵ^2, then we have that $\|Aq_j - ZR^A e_j\|_2 \leq j\gamma_A \epsilon$, $\|Bq_j - ZR^B e_j\|_2 \leq j\gamma_B \epsilon$, where the constants γ_A and γ_B are given by

$$\gamma_A \equiv \frac{|\mu_0|}{|\nu_0 \alpha_j + \mu_0 \beta_j|} \quad \text{and} \quad \gamma_B \equiv \frac{|\nu_0|}{|\nu_0 \alpha_j + \mu_0 \beta_j|}$$

If $\mu_0/\nu_0 = -\tau$, as in step (1) of the algorithm, then $\gamma_A = |\tau|/|\alpha_j - \tau\beta_j|$, $\gamma_B = 1/|\alpha_j - \tau\beta_j|$. These values can be large if $\tau \approx \alpha_j/\beta_j$. In practise an accuracy of order ϵ is achieved also if τ is close to detected eigenvalues. The ϵ-accuracy can be guaranteed when an additional refinement step is performed with values for (μ_0, ν_0) as $(\mu_0, \nu_0) = (1, \bar{\tau})$.

We will now explain the successive main phases of the algorithm.

(1) The initialization phase.

The choice for the scalars ν_0 and μ_0 is in particular effective if τ is in the interior of the spectrum. The choice causes a break-down if τ is an eigenvalue (which can easily be tested).

(2) The new vector t is made orthogonal with respect to the current search subspace V by means of modified Gram-Schmidt. Likewise, the vector $w = (\nu_0 A + \mu_0 B)t$ is made orthogonal with respect to the current test subspace W. The two orthogonalization processes can be replaced, for improved numerical stability, by a templates as in Alg. 4.4.

We expand the subspaces V, $V^A \equiv AV$, $V^B \equiv BV$, and W. V denotes the matrix with the current basis vectors v_i for the search subspace as its columns. The other matrices are defined in a similar obvious way.

(3) The m-th row and column of the matrices $M^A \equiv W^*AV$ and $M^B \equiv W^*BV$ are computed.

Note that the scalars $M^B_{i,m}$ can also be computed from the scalars $M^A_{i,m}$ and the orthogonalisation constants of $w_i^* w$ in step (2).

(4) The QZ decomposition for the pair (M^A, M^B) of m by m matrices can be computed by a suitable routine for dense matrix pencils dense matrices from LAPACK.

We have chosen to compute the generalized Petrov pairs, which makes the algorithm suitable for computing $k_{\max}$ interior generalized eigenvalues of $\beta A - \alpha B$, for which α/β is close to a specified τ.

For algorithms for reordering the generalized Schur form, see [39],[40],[11].

(5) The stopping criterion is to accept a generalized eigenpair approximation as soon as the norm of the residual (for the normalized right Schur vector approximation) is below ϵ. This means that we accept inaccuracies in the order of ϵ in the computed generalized eigenvalues, and inaccuracies (in angle) in the Schur vectors of $\mathcal{O}(\epsilon)$ (provided that the concerned eigenvalue is simple and well separated from the others).

Detection of all wanted eigenvalues cannot be guaranteed, see note (5) for Alg. 4.5.

(6) After acceptance of a Petrov pair, we continue the search for a next pair, with the remaining Petrov vectors as a basis for the initial search space.

(7) We restart when the dimension of the search space for the current eigenvector exceeds $m_{\max}$. The process is restarted with the subspaces spanned by the $m_{\min}$ left and right Ritz vectors corresponding to the generalized Ritz pairs closest to the target value τ.

(8) We have collected the locked (computed) right Schur vectors in Q, and the matrix $\widetilde{Q}$ is Q expanded with the current right Schur eigenvector approximation u. Likewise, the converged left Schur vectors have been collected in Z, and this matrix is expanded with p. This is done in order to obtain a more compact formulation; the correction equation in step (8) of Alg. 8.1 is equivalent to the one in equation (8.6) for the deflated pair in (8.12). The new correction t has to be orthogonal to the columns of Q as well as to u.

Of course, the correction equation can be solved by any suitable process, for instance a preconditioned Krylov subspace method that is designed to solve unsymmetric systems. However, because of the different projections, we always need a preconditioner (which may be the identity operator if nothing else is available) that is deflated by the same skew projections so that we obtain a mapping between $\widetilde{Q}^\perp$ and itself. Because of the occurrence of $\widetilde{Q}$ and $\widetilde{Z}$ one has to be careful with the usage of preconditioners for the matrix $\eta A - \zeta B$. The inclusion of preconditioners can be done as in Alg. 8.2. Make sure that the starting vector t_0 for an iterative solver satisfies the orthogonality constraints $\widetilde{Q}^* t_0 = 0$. Note that significant savings per step can be made in Alg. 8.2 if K is kept the same for a (few) Jacobi-Davidson iterations. In that case columns of $\widehat{Z}$ can be saved from previous steps. Also the matrix $\mathcal{M}$ can be updated from previous steps, as well as its $\mathcal{LU}$ decomposition.

It is not necessary to solve the correction equation very accurately. A strategy, often used for inexact Newton methods [9], here also works well: increase the accuracy with the Jacobi-Davidson iteration step, for instance, solve the correction equation with a residual reduction of $2^{-\ell}$ in the ℓ-th Jacobi-Davidson iteration (ℓ is reset to 0 when a Schur vector is detected).

In particular, in the first few initial steps, the approximate eigenvalue θ may be very inaccurate and then it does not make sense to solve the correction equation accurately. In this stage it can be more effective to temporarily replace θ by τ or to take $t = -r$ for the expansion of the search subspace [22, 12].

For a full theoretical background of this method, as well as for details on the deflation technique with Schur vectors, see [12].

ALGORITHM 8.2: **Approximate solution of the deflated Jacobi-Davidson correction equation, for the generalized eigenproblem**

"Solve" with left preconditioner $\widetilde{K} \equiv (I - \widetilde{Z}\widetilde{Z}^*)K(I - \widetilde{Q}\widetilde{Q}^*)$,
for $\widetilde{A} \equiv (I - \widetilde{Z}\widetilde{Z}^*)(\eta A - \zeta B)(I - \widetilde{Q}\widetilde{Q}^*)$:

(8a) *Solve* $\widehat{Z}$ *from* $K\widehat{Z} = \widetilde{Z}$,
Compute $\mathcal{M} = \widetilde{Q}^*\widehat{Z}$
Decompose $\mathcal{M} = \mathcal{L}\mathcal{U}$
Compute $\widetilde{r} \equiv \widetilde{K}^{-1}r$ *as:*
(b′) *solve* $\widehat{r}$ *from* $K\widehat{r} = r$
(c′) $\vec{\gamma} = \widetilde{Q}^*\widehat{r}$
solve $\vec{\beta}$ *from* $\mathcal{L}\vec{\beta} = \vec{\gamma}$
solve $\vec{\alpha}$ *from* $\mathcal{U}\vec{\alpha} = \vec{\beta}$
(d′) $\widetilde{r} = \widehat{r} - \widehat{Z}\vec{\alpha}$
Apply Krylov subspace method with start $t_0 = 0$,
with operator $\widetilde{K}^{-1}\widetilde{A}$, *and right-hand side* $-\widetilde{r}$,
$z = \widetilde{K}^{-1}\widetilde{A}v$ *for given* v *is computed as:*
(a) $y = (\eta A - \zeta B)v$
(b) *solve* $\widehat{y}$ *from* $K\widehat{y} = y$
(c) $\vec{\gamma} = \widetilde{Q}^*\widehat{y}$
solve $\vec{\beta}$ *from* $\mathcal{L}\vec{\beta} = \vec{\gamma}$
solve $\vec{\alpha}$ *from* $\mathcal{U}\vec{\alpha} = \vec{\beta}$
(d) $z = \widehat{y} - \widehat{Z}\vec{\alpha}$

8.8.4 Numerical Example

We present the results for a small example that can be easily repeated. We took the example from the collection of test matrices in [1].

We consider the bounded fineline dielectric waveguide generalized eigenproblem BFW782 [1] of order 782. This problem stems from a finite element discretization of the Maxwell equation for propagating modes and magnetic field profiles of a rectangular waveguide filled with dielectric and PEC structures. The resulting matrix A is non-symmetric and the matrix B is positive definite. Of special interest are the generalized eigenvalues (α, β) with positive real part (i.e., $\mathrm{Re}(\alpha/\beta) \geq 0$) and their corresponding eigenvectors.

For this problem, the parameters were set to: $\tau = 2750.0$, $k_{\max} = 5$ and $\epsilon = 10^{-9}$. In the first few steps, until the size of the first residual was smaller than 10^{-6}, we replaced (ζ, η) in the correction equation by $(1, \tau)$ (as explained in Comment (8)).

The computed generalized eigenvalues, represented as α/β, are given in Table 8.1. With Alg. 8.1 we discovered all 4 positive generalized eigenvalues.

The convergence history is plotted in Fig. 8.1. We solved the correction equation by (1) simply taken t as $-\widehat{r}$, denoted by GMRES_1, (2) with full GMRES [29] with a maximum of 10 steps, denoted by GMRES_{10}, and (3) with BiCGstab(2) [31] with a maximum of 100 matrix multiplications. We did not use preconditioning ($K = I$). As stopping criterion for the iterative methods for the correction equation, we used a residual reduction of $2^{-\ell}$ in the ℓth Jacobi-Davidson iteration or on the maximum number of iterations permitted. A summary of the results is given in Table 8.2. We see that the Jacobi-Davidson QZ method converges quite nicely for GMRES_{10} and BiCGstab(2). It should be noted that although it seems that with BiCGstab(2) only 4 generalized eigenvalues are computed, in fact 5 generalized eigenvalues are computed: the 2 rightmost generalized eigenvalues, that are relatively close, are found in same Jacobi-Davidson iteration.

$-1.1373e+03$
$5.6467e+02$
$1.2634e+03$
$2.4843e+03$
$2.5233e+03$

Table 8.1: 5 generalized eigenvalues of BFW782, computed by Jacobi-Davidson QZ.

Method for the correction equation	JD iterations	MVs	flops $\times 10^6$
GMRES_1	143	143	$6.70e+01$
GMRES_{10}	37	233	$3.17e+01$
BiCGstab(2)	32	429	$3.88e+01$

Table 8.2: Summary of results for BFW782.

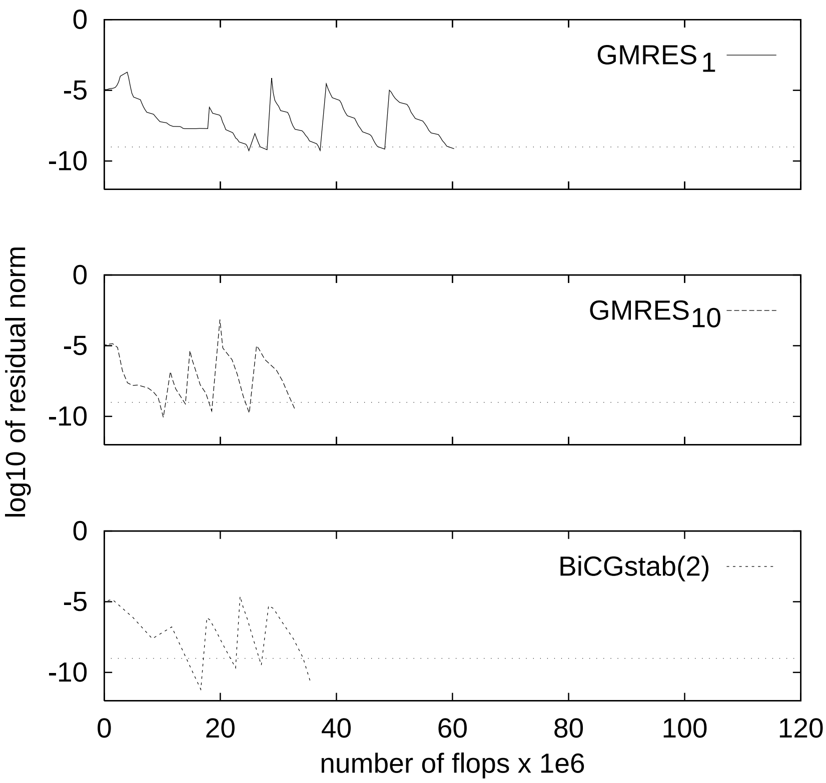

Figure 8.1: Convergence history for BFW782.

Software Availability

Matlab versions of the algorithms are available from either
`http://www.math.uu.nl/people/sleijpen`
or
`http://www.math.uu.nl/people/vorst`

Fortran implementations of the Jacobi-Davidson method are available via
`http://www.math.uu.nl/people/bomhof`
The Fortran software is of experimental type and does not reflect all insights and possibilities described in this Section.

CHAPTER 9

Quadratic and Nonlinear Eigenproblems

9.2 Quadratic Eigenvalue Problems

Z. Bai, G. Sleijpen and H. van der Vorst

9.2.1 Introduction

In this section, we consider the quadratic eigenvalue problem (QEP) of the form

$$(\lambda^2 M + \lambda C + K)x = 0 \quad \text{and} \quad y^*(\lambda^2 M + \lambda C + K) = 0, \tag{9.1}$$

where M, K and C are given matrices of size $n \times n$. The nontrivial n-vectors x, y, and the corresponding scalars λ are the right, left eigenvectors, and eigenvalues, respectively.

The matrix-function $L(\lambda) \equiv \lambda^2 M + \lambda C + K$ is a special case of a matrix polynomial, or a λ-matrix, see, for example, [13, 19]. In this case, it is a λ-matrix of degree 2. The matrix-function $L(\lambda)$ is said to be *regular* if $\det(L(\lambda))$ is not identical to zero for all λ. Otherwise, it is called *singular*.

An important special case of the quadratic eigenvalue problem is when

$$M^* = M > 0, \quad C^* = C \quad \text{and} \quad K^* = K > 0. \tag{9.2}$$

These matrices are sometimes called mass, damping and stiffness matrices, respectively, referring to their origin in mechanical engineering models, see, for instance, [10]. In some problems, the stiffness matrix K is only semi-positive definite. In this case, we may consider a shifted QEP to be discussed in §9.2.3.

One of factors makes the QEP different from standard eigenproblems $Ax = \lambda x$, or generalized eigenproblems $Ax = \lambda Bx$, is that there are $2n$ eigenvalues for QEP, with at most $2n$ right (and left) eigenvectors. Of course, in an n-dimensional space the right (and left) eigenvectors do no longer form an independent set. This is illustrated by the following simple example, in which we have taken some of the matrices as unsymmetric, in order to make some phenomena more clearly visible. The triplet

$$M = \begin{bmatrix} 5 & 2 \\ 1 & 4 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 \\ 0 & 7 \end{bmatrix}, K = \begin{bmatrix} 2 & 0 \\ 0 & 12 \end{bmatrix}.$$

has 4 different (but pairwise conjugate) eigenvalues (rounded to 5 decimals):

$$\begin{aligned} \lambda_1 &= -0.9396 + 1.5749i, \quad \lambda_2 = -0.9396 - 1.5749i, \\ \lambda_3 &= -0.0049 + 0.6296i, \quad \lambda_4 = -0.0049 - 0.6296i. \end{aligned}$$

The associated eigenvectors (normalized so that the first coordinate is equal to one) are:

$$x_1 = (1, -2.4756 - 0.9779i)^T, \quad x_2 = (1, -2.4756 + 0.9779i)^T,$$
$$x_3 = (1,\ 0.0326 - 0.0132i)^T, \quad x_4 = (1,\ 0.0326 + 0.0132i)^T.$$

The four eigenvectors are obviously dependent, but, in actual problems, each of them may represent a relevant state of the system.

One has to be careful with Rayleigh quotients for quadratic eigenproblems. Indeed, given x as a right eigenvector for the QEP (9.1), i.e.,

$$(\lambda^2 M + \lambda C + K)x = 0,$$

one can form a *quadratic Rayleigh quotient*:

$$\lambda^2(x^* M x) + \lambda(x^* C x) + (x^* K x) = 0. \tag{9.3}$$

However, this equation has two roots, and one of the roots is an eigenvalue, the other root may be a spurious one. For instance, if we compute the quadratic Rayleigh quotient for our example, with (λ_1, x_1), then clearly, the pair (λ_1, x_1) satisfies equation (9.3). If we solve equation (9.3), then we find the two roots $\mu_1 = -0.9396 + 1.5749i$, $\mu_2 = -0.8776 - 1.6057i$. We see that λ_1 is recovered (by μ_1), the other root has no meaning for the given QEP.

In an effort to decide which of the two is the desired one and which is the spurious one, one could compute the residual vector

$$r_\mu \equiv (\mu^2 M + \mu C + K)x_1,$$

and this leads to $\|r_{\mu_1}\|_2 \approx 8.4 \times 10^{-14}$, $\|r_{\mu_2}\|_2 \approx 12.5$, which, in this case, clearly points that μ_2 is not an eigenvalue. We can not exclude that in contrived examples, one might make a wrong choice, which may lead to a delay in a specific iterative solution method.

For more general matrices, we can have defectiveness, as for the standard eigenproblems, which means that there is not necessarily a complete set of eigenvectors. In the next section, we will relate the QEP to a generalized standard problem, which helps to shed more light on this matter.

9.2.2 Reduction to Linear Form

It is easy to see that the QEP in (9.1) is equivalent to the following generalized "linear" eigenvalue problem [2]:

$$Az = \lambda B z \quad \text{and} \quad w^* A = \lambda w^* B \tag{9.4}$$

where

$$A = \begin{bmatrix} 0 & I \\ -K & -C \end{bmatrix}, \quad B = \begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix} \tag{9.5}$$

and

$$z = \begin{bmatrix} x \\ \lambda x \end{bmatrix}, \quad w = \begin{bmatrix} (\lambda M + C)^* y \\ y \end{bmatrix}. \tag{9.6}$$

The generalized eigenvalue problem (9.4) is commonly called the linearization of the QEP (9.1). It can be shown that for any matrices A and B of the above forms, the right and left eigenvectors z and w have the structures described in (9.6).

Note that from the factorization

$$A - \lambda B = \begin{bmatrix} 0 & I \\ -I & -\lambda M - C \end{bmatrix} \begin{bmatrix} \lambda^2 M + \lambda C + K & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ -\lambda I & I \end{bmatrix}, \tag{9.7}$$

[2] The term "linear" is in the quotation mark, because λ appears in the eigenvectors as well.

we can conclude that the pencil $A - \lambda B$ is *equivalent*[3] to the matrix

$$\begin{bmatrix} \lambda^2 M + \lambda C + K & 0 \\ 0 & I \end{bmatrix} \tag{9.8}$$

and

$$\det(A - \lambda B) = \det(\lambda^2 M + \lambda C + K),$$

This means that the eigenvalues of the original QEP (9.1) coincide with the eigenvalues of the generalized eigenvalue problem (9.4). Furthermore, we have

- $L(\lambda)$ is regular if and only if $A - \lambda B$ is regular.
- If M (hence B) is nonsingular, then $L(\lambda)$ is regular.
- If K (hence A) is nonsingular, then $L(\lambda)$ is regular.

For theory on regular pencils (A, B), see for instance [35, Chapter VI]. We will assume that at least M is nonsingular throughout this section.

A disadvantage of the above reduction to generalized form is that if the matrices M, C and K are all Hermitian, then this is not reflected in the reduced form (9.5), where A is non-Hermitian. This can be repaired as follows.

In fact, the matrix pair (A, B) in (9.4) can be chosen in a more general form

$$A = \begin{bmatrix} 0 & W \\ -K & -C \end{bmatrix}, \quad B = \begin{bmatrix} W & 0 \\ 0 & M \end{bmatrix},$$

where W can be any arbitrary nonsingular matrix. Note that now the matrix pencil $A - \lambda B$ is equivalent to the matrix polynomial (9.8) if and only if W is nonsingular, and because of (9.7)

$$\det(A - \lambda B) = \det(W) \cdot \det(\lambda^2 M + \lambda C + K).$$

For example, if the matrices M, K and C are all symmetric, as in the special case (9.2), and K is nonsingular, then we may choose $W = -K$, which leads to the following symmetric generalized "linear" eigenvalue problem

$$Az = \lambda Bz \quad \text{and} \quad w^* A = \lambda w^* B \tag{9.9}$$

where

$$A = \begin{bmatrix} 0 & -K \\ -K & -C \end{bmatrix}, \quad B = \begin{bmatrix} -K & 0 \\ 0 & M \end{bmatrix}, \tag{9.10}$$

and

$$z = \begin{bmatrix} x \\ \lambda x \end{bmatrix}, \quad w = \begin{bmatrix} y \\ \bar{\lambda} y \end{bmatrix}. \tag{9.11}$$

Both A and B are symmetric, but may be indefinite.

9.2.3 Spectral Transformations for QEP

9.2.3.1 Invert QEP

For most iterative methods for solving a generalized eigenvalue problem, the formulation (9.4), with either (9.5) or with (9.10), is suitable if one wants to determine a few of the exterior eigenvalues and eigenvectors. If one wants to compute some of the smallest (in modulus) eigenvalues and eigenvectors, then the obvious transformation is $\mu = 1/\lambda$, and, after multiplying the QEP (9.1) with μ^2, we obtain the *invert QEP*:

$$\left(M + \mu C + \mu^2 K\right) x = 0. \tag{9.12}$$

[3]Two matrix polynomials $M_1(\lambda)$ and $M_2(\lambda)$ of size $n \times n$ are called equivalent if $M_1(\lambda) = E(\lambda) M_2(\lambda) F(\lambda)$ for some $n \times n$ matrix polynomials $E(\lambda)$ and $F(\lambda)$ with constant nonzero determinants (unimodular).

Here it is assumed that that 0 is not an eigenvalues of the original QEP (9.1), i.e., K is nonsingular.

The QEP for the triplet $\{K, C, M\}$ can be linearized as discussed in §9.2.2, for instance as (9.4) with (9.5), where M interchanged with K. We can reformulate this generalized linearized eigenproblem in terms of λ, instead of μ, which leads to

$$Az = \frac{1}{\lambda} Bz \tag{9.13}$$

where

$$A = \begin{bmatrix} -C & -M \\ I & 0 \end{bmatrix}, \quad B = \begin{bmatrix} K & 0 \\ 0 & I \end{bmatrix}, \quad z = \begin{bmatrix} x \\ \lambda x \end{bmatrix}. \tag{9.14}$$

Note that from the factorization

$$B - \lambda A = \begin{bmatrix} I & \lambda M \\ 0 & I \end{bmatrix} \begin{bmatrix} \lambda^2 M + \lambda C + K & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ -\lambda I & I \end{bmatrix}$$

we know that the pencil $B - \lambda A$ is equivalent to

$$\begin{bmatrix} \lambda^2 M + \lambda C + K & 0 \\ 0 & I \end{bmatrix}.$$

Since $\det(B - \lambda A) = \det(\lambda^2 M + \lambda C + K)$, we conclude that the matrix pencil $B - \lambda A$ is regular if and only if the quadratic matrix polynomial $\lambda^2 M + \lambda C + K$ is regular and the eigenvalues of the original QEP (9.1) coincide with the eigenvalues of the matrix pencil $B - \lambda A$.

For the special case (9.2), we may formulate the generalized eigenvalue problem $Az = \frac{1}{\lambda} Bz$, with

$$A = \begin{bmatrix} C & M \\ M & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -K & 0 \\ 0 & M \end{bmatrix}. \tag{9.15}$$

In this case, both matrices are Hermitian, but indefinite. Linearization with (9.15) results after left multiplication of (9.14) with a block diagonal matrix $\mathrm{diag}(-I, -M)$. Therefore, if $\det(M) \neq 0$, then the pencil $B - \lambda A$ is regular if and only if the quadratic matrix polynomial $\lambda^2 M + \lambda C + K$ is regular.

9.2.3.2 Shifted QEP

With a shift $\lambda = \mu + \sigma$, the *shifted QEP* is

$$\left(\mu^2 \widehat{M} + \mu \widehat{C} + \widehat{K}\right) x = 0, \tag{9.16}$$

where $\widehat{M} = M$, $\widehat{C} = C + 2\sigma M$ and $\widehat{K} = K + \sigma C + \sigma^2 M$. The shift transforms eigenvalues λ of (9.1) close to σ become eigenvalues μ close to 0.

The corresponding generalized "linear" eigenvalue problem is (again in terms of λ, rather than μ):

$$\begin{bmatrix} 0 & I \\ -\widehat{K} & -\widehat{C} \end{bmatrix} \begin{bmatrix} x \\ (\lambda - \sigma)x \end{bmatrix} = (\lambda - \sigma) \begin{bmatrix} I & 0 \\ 0 & \widehat{M} \end{bmatrix} \begin{bmatrix} x \\ (\lambda - \sigma)x \end{bmatrix},$$

or

$$\begin{bmatrix} 0 & \widehat{K} \\ \widehat{K} & \widehat{C} \end{bmatrix} \begin{bmatrix} x \\ (\lambda - \sigma)x \end{bmatrix} = (\lambda - \sigma) \begin{bmatrix} \widehat{K} & 0 \\ 0 & -\widehat{M} \end{bmatrix} \begin{bmatrix} x \\ (\lambda - \sigma)x \end{bmatrix}.$$

if Hermitian of the matrix triplet $\{M, K, C\}$ wants to be preserved.

9.2.3.3 Shift-and-Invert QEP

Combining the above invert and shift spectral transformations, the so-called *shift-and-invert QEP* becomes

$$\left(\mu^2 \widehat{M} + \mu \widehat{C} + \widehat{K}\right) x = 0, \tag{9.17}$$

where

$$\mu = \frac{1}{\lambda - \sigma},$$

and $\widehat{M} = \sigma^2 M + \sigma C + K$, $\widehat{C} = C + 2\sigma M$, and $\widehat{K} = M$. The exterior eigenvalues μ of the QEP (9.17) approximate the eigenvalues λ of the original QEP (9.1) closest to the shift σ. These eigenvalues λ are given by

$$\sigma + \frac{1}{\mu}.$$

Again, the corresponding generalized "linear" eigenvalue problem in terms of λ, rather than μ, is

$$\begin{bmatrix} -\widehat{C} & -\widehat{K} \\ I & 0 \end{bmatrix} \begin{bmatrix} x \\ (\lambda - \sigma)x \end{bmatrix} = \frac{1}{\lambda - \sigma} \begin{bmatrix} \widehat{M} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ (\lambda - \sigma)x \end{bmatrix},$$

or

$$\begin{bmatrix} \widehat{C} & \widehat{K} \\ \widehat{K} & 0 \end{bmatrix} \begin{bmatrix} x \\ (\lambda - \sigma)x \end{bmatrix} = \frac{1}{\lambda - \sigma} \begin{bmatrix} -\widehat{M} & 0 \\ 0 & \widehat{K} \end{bmatrix} \begin{bmatrix} x \\ (\lambda - \sigma)x \end{bmatrix}.$$

if Hermitian of the matrix triplet $\{M, K, C\}$ wants to be preserved.

9.2.3.4 QEP with Cayley Transform

With the so-called Cayley transform,

$$\mu = \frac{\alpha\lambda - \beta}{\lambda - \tau},$$

where the parameters α, β and τ are chosen such that $\alpha\tau - \beta \neq 1$, the original QEP (9.1) becomes

$$\left(\mu^2 \widehat{M} + \mu \widehat{C} + \widehat{K}\right) x = 0, \tag{9.18}$$

where $\widehat{M} = \tau^2 M + \tau C + K$, $\widehat{C} = -2\tau\beta M - (\alpha\tau + \beta)C - 2\alpha K$, and $\widehat{K} = \beta^2 M + \alpha\beta C + \alpha^2 K$. Eigenvalues λ of the original QEP (9.1) close to the *anti-shift* τ are transformed into large (in modulus) eigenvalues μ of the QEP (9.18). Eigenvalues λ close to the *shift* β/α correspond to eigenvalues μ of (9.17) close to 0.

Note that the triple $\{\widehat{M}, \widehat{C}, \widehat{K}\}$ is symmetric if that is the case for the real triple $\{M, C, K\}$ and if α, β, and τ are real.

9.2.4 Higher Order Polynomial Eigenvalue Problems

Some applications lead to higher order polynomial problems

$$\Psi(\lambda)x = 0 \qquad y^* \Psi(\lambda) = 0, \tag{9.19}$$

where

$$\Psi(\lambda) \equiv \lambda^\ell C_\ell + \lambda^{\ell-1} C_{\ell-1} + \ldots + \lambda C_1 + C_0, .$$

in which the C_j are square n by n matrices. In order to make this problem well defined, these matrices have to satisfy certain properties, in particular C_ℓ should be nonsingular. Similar to the quadratic problem, these problems can also be linearized to

$$Az = \lambda Bz,$$

where

$$A \equiv \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & I \\ -C_0 & -C_1 & -C_2 & \cdots & -C_{\ell-1} \end{bmatrix}, \qquad B \equiv \begin{bmatrix} I & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \\ & & & & C_\ell \end{bmatrix}.$$

The relation between x and z is given by $z = (x^T, \lambda x^T, \ldots, \lambda^{\ell-1} x^T)^T$. The generalized eigenproblem $Az = \lambda Bz$ can be solved with one of the methods discussed in Chapter 8. A disadvantage of this approach is that one has to work with larger matrices of order $n \times \ell$, and these matrices have, of course, also $n \times \ell$ eigenpairs. This implies that one has to check which of the computed eigenpairs satisfies the original polynomial equation. Ruhe [27] (see also Davis [8]) discusses methods that handle directly the problem (9.19), for instance, with Newton's method. For larger values of n one may expect all sorts of problems with the convergence of these techniques (Ruhe's examples are pretty small). In section 9.2.6, we will discuss a method that can be used to attack problems with large n. In that approach, one first projects the given problem (9.19) onto a low dimensional subspace, and obtains a similar problem of low dimension. This low dimensional polynomial eigenproblem can then be solved with one of the approaches mentioned above. In [16] a fourth-order polynomial problem has been solved successfully, using this reduction technique.

9.2.5 Numerical Methods for Solving Linearized Problems

As discussed in §9.2.2, one can use either the generalized eigenvalue problem (9.4) with (9.5) or the generalized eigenvalue problem (9.9) with (9.10), for solving the corresponding quadratic eigenvalue problem (9.1).

If all matrices M, C and K are Hermitian and M is positive definite, as in the special case (9.2), then the decision comes down to choosing either intrinsically non-Hermitian generalized eigenvalue problem (9.4) and (9.5), with a Hermitian positive definite B matrix, or a generalized eigenvalue problem (9.9) and (9.10), where both A and B matrices are Hermitian but neither of them will be positive definite.

Numerical methods discussed in [2, Ch. 8]can be used for solving these generalized "linear" eigenvalue problems. The indefinite symmetric Lanczos method discussed in [2, Section 8.7]is specifically targeted for Hermitian indefinite generalized eigenvalue problem (9.9). The symmetric Lanczos method, see [2, Section 4.4], is formally extended to solve such symmetric indefinite generalized eigenvalue problem. The trouble is that the basis vectors are orthogonal with respect to an indefinite inner product. Therefore, these basis vectors may be linearly independent and the algorithm may be breakdown and numerical instable. Nevertheless this is an attractive way to solve the original QEP because of potential savings in memory requirements and floating point operations. See [2, Section 8.7]for further details.

9.2.6 The Jacobi-Davidson method

The possible disadvantage of the linearized approach is the doubling of the dimension of the problems, that is a problem with n-dimensional matrices M, C, and K, is transformed to a generalized problem with $2n$-dimensional matrices A and B. This is avoided in the Jacobi-Davidson method to be discussed in this section. In this method, the QEP is first projected onto a low-dimensional subspace, which leads to a QEP of modest dimension. This low-dimensional projected QEP can be solved with any method of choice. Expansion of the subspace is realized by a Jacobi-Davidson correction equation. For polynomial eigenproblems this technique was first suggested and discussed in [30, sec.8].

As we will see below, this method can also be applied directly to polynomial eigenproblems

$$(\lambda^\ell C_\ell + \ldots + \lambda C_1 + C_0)x = 0, \tag{9.20}$$

and no transformation to a generalized "linear" eigenvalue problem will be required, where C_i for $i = 0, 1, \ldots, \ell$, are given $n \times n$ matrices. For simplicity, we will only present the case for $\ell = 2$.

In the first part of a Jacobi-Davidson iteration step, for solving the polynomial eigenproblem

$$\Psi(\lambda)x = 0 \tag{9.21}$$

where

$$\Psi(\lambda) \equiv \lambda^2 C_2 + \lambda C_1 + C_0, \tag{9.22}$$

the projected polynomial problem

$$(\theta^2 V^* C_2 V + \theta V^* C_1 V + V^* C_0 V)s = 0 \tag{9.23}$$

is solved. The columns v_i of the $n \times m$ matrix V form a basis for the search subspace. For stability reasons, the columns are constructed to be orthonormal. The projected problem has the same order as the original one, but is of much smaller dimension, typically $m \ll n$. We will assume that the solution vectors s are normalized, $||s||_2 = 1$. First, a Ritz value θ with the desired properties, such as the largest real part or closet to some target τ, is selected and for the associated eigenvector s. Then the Ritz vector $u \equiv Vs$ and the residual $r \equiv \Psi(\theta)u$ is computed. For expansion of the search space the vector p,

$$p \equiv \Psi'(\theta)u$$

with

$$\Psi'(\theta) = 2\theta C_2 + C_1,$$

is also computed.

In the second part of the Jacobi-Davidson iteration step, the search subspace span(V) is expanded by a vector $t \perp u$ that solves (approximately) the correction equation

$$\left(I - \frac{p\,u^*}{u^* p}\right) \Psi(\theta)\,(I - u\,u^*)\,t = -r. \tag{9.24}$$

The next column of V is obtained by orthonormalizing the approximate solution against the previously computed columns $v_1, \ldots, v_m$.

This process is repeated until an eigenpair (λ, x) has been detected, i.e., until the residual vector r is sufficiently small. The basic form the algorithm is presented in Algorithm 9.1. We refer to [34], for an example of a quadratic eigenvalue problem arising from an acoustic problem, that has been solved with this reduction technique, for n up to 250,000.

ALGORITHM 9.1: **Jacobi-Davidson Method**

(1) *Choose an $n \times m$ orthonormal matrix V.*
(2) **For** $i = 0, 1, 2$
  compute $W_i = C_i V$ and $M_i = V^* W_i$.
(3) *Iterate until convergence*
(4) *Compute the eigenpairs (θ, s) of*
  $\left(\theta^2 M_2 + \theta M_1 + M_0\right) s = 0.$
(5) *Select the desired eigenpair (θ, s) with $||s||_2 = 1$.*
(6) *Compute $u = Vs$, $p = \Psi'(\theta)u$, $r = \Psi(\theta)u$.*
(7) **If** $(||r||_2 < \epsilon)$, $\lambda = \theta$, $x = u$, **then** *STOP.*
(8) *Solve (approximately) a $t \perp u$ from*
  $\left(I - \frac{p\,u^*}{u^* p}\right) \Psi(\theta)\,(I - u\,u^*)\,t = -r$
(9) *Orthogonalize t against V, $v = t/||t||_2$.*
  For $i = 0, 1, 2$
  compute $w_i = C_i v$
  $M_i = \begin{bmatrix} M_i & V^* w_i \\ v^* W_i & v^* w_i \end{bmatrix}$, $W_i = [W_i, w_i]$
(10) *Expand $V = [V, v]$.*

If the dimension of the search subspace becomes too large, if the number of columns of V is equal to, say, $m = m_{\max}$, then the process can be continued with a search subspace of smaller dimension. Take for the new search subspace the space spanned by the best $m_{\min}$ Ritz pairs from the last step, that is, take $V = [u_1, \ldots, u_{m_{\min}}]$, where $u_1, \ldots, u_{m_{\min}}$ are the Ritz vectors associated with the best available $m_{\min}$ Ritz values $\theta_1, \ldots, \theta_{m_{\min}}$. Then apply modified Gram-Schmidt to orthonormalize V and restart with this matrix. Note that the new search matrix $V = V_{m_{\min}}$ can be expressed in terms of the old search matrix $V = V_{m_{\max}}$ as $V_{m_{\min}} = V_{m_{\max}} T$ for some $m_{\max} \times m_{\min}$ matrix T. The transformation matrix T can explicitly be computed, and can be used to update the auxiliary matrices W_i $(= W_i T)$ and M_i $(= T^* M_i T)$.

Eigenvectors that already have been detected can be kept in the search subspace (explicit deflation), if more eigenvectors are wanted. Keeping the eigenvectors in the search subspace prevents the process from reconstruction known eigenvectors.

In the §4.7.3.2 and 8.8.2.1, we suggested to use "implicit" deflation, since that approach is based on Schur forms and this is more stable, with better conditioned correction equations. However, in this general polynomial setting, it is not clear how to incorporate implicit deflation: the Schur form and the generalized Schur form can not easily be generalized.

9.2.7 Notes and References

Numerical algorithm design and analysis for the solution of quadratic eigenvalue problem are still an active researcher subject. Besides those methods discussed in this section, some alternative methods are available in literature, see for example [19, 18, 27, 8, 15]. Most of these methods are variants of Newton's method. They generally have good local convergence properties, and find one eigenpair at time. In [15], a proper deflation technique is presented for finding more than one eigenpair.

Recently, a backward error analysis of the quadratic eigenvalue problems and more generally, the polynomial eigenvalue problems is presented in [37] is studied. In [36], a perturbation analysis of the quadratic eigenvalue problem (9.2) is presented.

References

[1] Z. Bai, D. Day, J. Demmel, and J. Dongarra, *A test matrix collection for non-Hermitian eigenvalue problems*, LAPACK Working Note #123, Technical Report CS-97-355, University of Tennessee, 1997.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. Van der Vorst, eds., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, PA, to appear in 2000.

[3] Z. J. Bai and J. W. Demmel, *On swapping diagonal blocks in real Schur form*, Linear Algebra Appl., 186 (1993), pp. 73–95.

[4] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the solution of linear systems : building blocks for iterative methods*, SIAM, Philadelphia, PA, USA, 1994.

[5] M. Crouzeix, B. Philippe, and M. Sadkane, *The Davidson method*, SIAM J. Sci. Comput., 15 (1994), pp. 62–76.

[6] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, *Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 772–795.

[7] E. R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real symmetric matrices*, J. Comput. Phys., 17 (1975), pp. 87–94.

[8] G. J. Davis, *Numerical solution of a quadratic matrix equation*, SIAM J. Sci. Comput., 2 (1981), pp. 164–175.

[9] R. S. Dembo, E. S. C., and S. T., *Inexact newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[10] R. J. Duffin, *A minimax theory for overdamped networks*, J. of Rational Mechanics and Analysis, 4 (1955), pp. 221–233.

[11] D. R. Fokkema, *Subspace methods for linear, nonlinear, and eigen problems*, PhD thesis, Utrecht University, Utrecht, 1996.

[12] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Jacobi-Davidson style QR and QZ algorithms for the partial reduction of matrix pencils*, SIAM J. Sci. Comput., 20 (1998), pp. 94–125.

[13] F. Gantmacher, *The Theory of Matrices, vol. I and II (transl.)*, Chelsea, New York, 1959.

[14] M. Genseberger and G. L. Sleijpen, *Alternative corection equations in the Jacobi-Davidson method*, Preprint 1073, Department of Mathematics, Utrecht University, Utrecht, the Netherlands, 1998.

[15] J.-S. Guo, W.-W. Lin, and C.-S. Wang, *Numerical solutions for large sparse quadratic eigenvalue problems*, Linear Alg. Appl., 225 (1995), pp. 57–89.

[16] R. Heeg, *Stability and transistion of attachment-line flow*, PhD thesis, Universiteit Twente, Enschede, The Netherlands, 1998.

[17] C. G. J. Jacobi, *Ueber ein leichtes Verfahren, die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen*, Journal für die Reine und Angewandte Mathematik, 30 (1846), pp. 51–94.

[18] V. N. Kublanovskaja, *On an application to the solution of the generalized latent value problem for λ-matrices*, SIAM J. Numer. Anal., 7 (1970), pp. 532–537.

[19] P. Lancaster, *Lambda-matrices and vibrating systems*, Pergamon Press, Oxford, England, 1966.

[20] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, sinum, 10 (1973), pp. 241–256.

[21] R. B. Morgan, *Computing interior eigenvalues of large matrices*, Linear Alg. Appl., 154–156 (1991), pp. 289–309.

[22] R. B. Morgan and D. S. Scott, *Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 817–825.

[23] J. Olsen, P. Jørgensen, and J. Simons, *Passing the one-billion limit in full configuration-interaction (FCI) calculations*, Chemical Physics Letters, 169 (1990), pp. 463–472.

[24] C. C. Paige, B. N. Parlett, and H. A. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Num. Lin. Alg. Appl., 2 (1995), pp. 115–133.

[25] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[26] B. N. Parlett, *The symmetric eigenvalue problem*, Prentice Hall series in computational mathematics, Prentice-Hall, 1980. Reprinted by SIAM Press, 1997.

[27] A. Ruhe, *Algorithms for the nonlinear eigenvalue problem*, SIAM J. Numer. Anal., 10 (1973), pp. 674–689.

[28] A. Ruhe, *The Rational Krylov algorithm for nonsymmetric eigenvalue problems, III: Complex shifts for real matrices*, BIT, 34 (1994), pp. 165–176.

[29] Y. Saad and M. H. Schultz, *GMRES : a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[30] G. L. G. Sleijpen, G. L. Booten, D. R. Fokkema, and H. A. van der Vorst, *Jacobi Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT, 36 (1996), pp. 595–633.

[31] G. L. G. Sleijpen and D. R. Fokkema, *Bi-CGSTABℓ methods for linear equations involving matrices with complex spectrum*, ETNA, 1 (1993), pp. 11–32.

[32] G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Applic., 17 (1996), pp. 401–425.

[33] G. L. G. Sleijpen, H. A. van der Vorst, and E. Meijerink, *Efficient expansion of subspaces in the Jacobi-Davidson method for standard and generalized eigenproblems*, ETNA, 7 (1998), pp. 75–89.

[34] G. L. G. Sleijpen, H. A. van der Vorst, and M. B. van Gijzen, *Quadratic eigenproblems are no problem*, SIAM News, 29 (1996), pp. 8–9.

[35] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990.

[36] J. G. Sun, *Perturbation analysis of quadratic eigenvalue problems*, BIT, (1999). submitted.

[37] Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Tech. Report Numerical Analysis Report No. 332, The Univ. of Manchester, Machester, England, 1998.

[38] H. A. van der Vorst, *A generalized Lanczos scheme*, Math. Comp., 39 (1982), pp. 559–561.

[39] P. Van Dooren, *A generalized eigenvalue approach for solving Ricatti equations*, SIAM J. Sci. Comput., 2 (1981), pp. 121–135.

[40] ——, *Algorithm 590, DUSBSP and EXCHQZ: FORTRAN subroutines for computing deflating subspaces with specified spectrum*, ACM Trans. Math. Softw., 8 (1982), pp. 376–382.