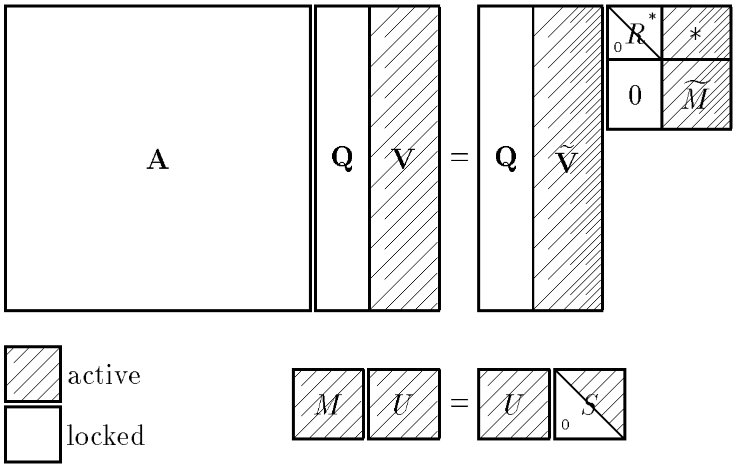
Universiteit Utrecht

Department
of Mathematics



Jacobi-Davidson style QR and QZ algorithms for the partial reduction of matrix pencils

by

Diederik R. Fokkema, Gerard L.G. Sleijpen, and Henk A. Van der Vorst

Preprint

nr. 941

January, 1996

JACOBI-DAVIDSON STYLE QR AND QZ ALGORITHMS FOR THE PARTIAL REDUCTION OF MATRIX PENCILS

DIEDERIK R. FOKKEMA[*‡] GERARD L.G. SLEIJPEN[*] AND HENK A. VAN DER VORST[*]

Abstract. The Jacobi-Davidson subspace iteration method offers possibilities for solving a variety of eigenproblems. In practice one has to apply restarts because of memory limitations, in order to restrict computational overhead, and also if one wants to compute several eigenvalues. In general, restarting has negative effects on the convergence of subspace methods. We will show how effective restarts can be incorporated in the Jacobi-Davidson methods, very similar to the implicit restart procedure for the Arnoldi process. We will present two algorithms, JDQR for the standard eigenproblem, and JDQZ for the generalized eigenproblem, that are based on the iterative construction of the (generalized) partial Schur form with the Jacobi-Davidson approach. The algorithms are suitable for the efficient computation of several (even multiple) eigenvalues, and the corresponding eigenvectors, near a user-specified target value in the complex plane.

Key words. linear eigenproblems, generalized eigenproblems, Schur form, generalized Schur form, QR-algorithm, QZ-algorithm, Jacobi-Davidson, iterative methods.

AMS subject classification. 65F15, 65N25.

1 Introduction. In this paper we propose two iterative methods, one for computing solutions of the standard eigenproblem

$$(\mathbf{A} - \lambda\,\mathbf{I})\,\mathbf{q} = \mathbf{0}, \tag{1}$$

and the other for computing solutions for the generalized eigenproblem[1]

$$(\beta\,\mathbf{A} - \alpha\,\mathbf{B})\,\mathbf{q} = \mathbf{0}, \tag{2}$$

where $\mathbf{A}$ and $\mathbf{B}$ are large and sparse $(n\times n)$-matrices, which may be complex and/or non-normal. Of course, with $\mathbf{B} = \mathbf{I}$ the generalized eigenproblem reduces to a standard eigenproblem, and we could have restricted ourselves to the generalized eigenproblem case. However, simplifications are possible when $\mathbf{B} = \mathbf{I}$, that help reduce the memory requirements and the computational complexity. For this reason, and also since the standard problem allows for a less complicated description, we have chosen to consider both situations in detail.

Our algorithms are based on the Jacobi-Davidson method described in [20], and adapted for generalized eigenproblems (and other polynomial eigenproblems) in [18]. We have chosen the Jacobi-Davidson approach for the computation of a partial Schur form for the standard eigenproblem, and for a partial generalized Schur form for the generalized eigenproblem. The partial Schur forms have been chosen mainly for numerical stability, since they involve orthogonal bases. These bases are also useful for deflation, another ingredient of our algorithms.

In the Jacobi-Davidson approach a search subspace is generated onto which the given eigenproblem is projected. The much smaller projected eigenproblem is solved and this leads to approximations for the wanted eigenvectors and eigenvalues of the given larger problem. This is the 'Davidson' part of the method. Then, a correction equation for a selected eigenpair is considered. The solution of the correction equation defines an orthogonal correction for the current eigenvector approximation (in fact if the exact value for the eigenvalue would have been known, then the correction equation defines the exact eigenvector); this is the 'Jacobi' part of

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

[‡]Present address: ISE Integrated Systems Engineering AG, Technopark Zürich, Technoparkstrasse 1, CH-8005 Zürich, Switzerland. E-mail: `fokkema@ise.ch`.

[1]The family $\mathbf{A} - \lambda\mathbf{B}$ is called a *matrix pencil* and the generalized eigenvalues $\langle\alpha,\beta\rangle$, solutions of (2), are also called eigenvalues of the matrix pencil (cf. e.g., [24]).

the algorithm. The correction is then used for the expansion of the search subspace and the process is repeated.
The correction equation may be solved by any method of choice, and for large problems it is often more efficient to solve this equation only approximately by some iterative method. The speed of convergence of this iterative method may be improved by preconditioning, and this is where in the Jacobi-Davidson method preconditioning takes place. It should be noted that the preconditioning does not affect the given eigenproblem. By including shifts in the Jacobi-Davidson method, and by a proper selection of the approximate eigenpair for the correction equation, the process can be guided to find eigenpairs close to a given target value. More details will be given in §2.1 and §3.1, and for a complete description of the Jacobi-Davidson method we refer to [20], [18].

A problem in the Jacobi-Davidson method is that convergence towards a specific eigenvalue is favored, and for efficient computation of several eigenvalues, one has to apply the usual restart with a different target. Because of memory limitations one may also be forced to restart, even before an eigenpair is found. Restarts have the disadvantage that a subspace, that may contain very useful information, is replaced by one single vector, so that much valuable information is lost. This problem has been solved elegantly for the Arnoldi method [22], and our approach (cf. §2.2 and §3.2) is related to this (see also [20, §5.3]). For the Jacobi-Davidson method this problem is solved by our new algorithms. In these algorithms the given large system is projected on a suitably filtered subspace, and this leads to a similar, but much smaller, problem. The projected problem is reduced to Schur form by either the QR [6] or the QZ [11] method. Also the construction of the subspace and the projected system may be viewed as iterative inexact forms of QR and QZ. For this reason we have named our new methods JDQR and JDQZ, respectively. JDQR generates a partial Schur form for the standard eigenproblem, a partial "QR-decomposition"; JDQZ produces a partial generalized Schur form for the generalized eigenproblem, a partial "QZ-decomposition".

Our §2 focuses on the standard eigenproblem and is organized as follows. The Jacobi-Davidson method is briefly discussed in §2.1. In §2.2, the original algorithm of [20] is slightly adapted, by incorporating an ordered Schur decomposition of the projected eigenproblem. An ordered Schur form can be used to select the approximate eigenpair and to select a properly limited sized subspace for continuing the process. This culminates in the construction of the JDQR method, in §2.3: by combining the Schur decomposition of the projected problem with the iteratively obtained partial Schur decomposition of the given large problem, a simple and efficient algorithm is obtained. The rate of convergence of Jacobi-Davidson depends on the accuracy by which the correction equation is solved. If it is solved iteratively, then it may be attractive to include preconditioning. This requires extra care because of the projection operations included in the correction equation; attention to this will be paid in §2.4. In our approach, a preconditioner constructed for one Ritz pair appears to be effective also for nearby Ritz pairs (see §2.7). Some observations on the selection of approximate eigenpairs are made in §2.5. In §2.6, it is explained that the search subspace for the obtained partial Schur decomposition provides a suitable limited subspace for continuing the iterative process.

The generalized eigenproblem is central in §3, where JDQR is generalized to JDQZ. The JDQZ approach basically allows for two different choices for constructing the projected problem (cf. §§3.1 and 3.5), one of which is very useful for computing interior eigenvalues. For such eigenvalues, it appears to be more effective to select more carefully the approximate eigenpairs than to strive for optimal expansion of the search subspace (cf. §3.5). The derivation of JDQZ is given in §3.3, while in §3.4 it is explained how preconditioning for the correction equation can be incorporated.

In §4, we illustrate the convergence behavior of JDQR and JDQZ with numerical experiments for a number of eigenproblems. Aspects that are investigated concern, amongst others, the accuracy of the solution of the correction equation (§4.1), the effect of preconditioning (§4.2), multiple eigenvalues (§4.3 and §4.7), interior eigenvalues (§4.4 and §4.8), and different approaches for the construction of the projected problem (§4.4 and §4.8), and implicit versus explicit deflation (§4.6).

§5 contains our conclusions.

REMARK 1 All computations can be done in complex arithmetic if necessary. An alternative for real matrices would be to use quasi Schur forms with 2×2 blocks on the diagonal, which can be done in real arithmetic. It is possible to derive a variant of Jacobi-Davidson based on this blocked form. However, this variant will necessarily also involve a blocked version of the correction equation, which will double the computational work per iteration step. The construction of a suitable block-preconditioner may be a problem as well. Hence, it is not clear whether implementations with quasi Schur forms may be competitive, and we will not discuss this possibility further in this paper.

REMARK 2 With bold face letters we indicate that variables are associated with the large n-dimensional space, and for low dimensional spaces we use italic letters.
We use a tilde to indicate that a quantity approximates the corresponding quantity without tilde: $\widetilde{\mathbf{q}}$ approximates $\mathbf{q}$, etc.. The algorithms are given in MATLAB-style. We use the MATLAB conventions when we refer to entries in matrices and vectors. In particular, where in the algorithms new values overwrite old ones, the tildes are deleted.

2 The standard eigenproblem. We will focus on the standard eigenproblem (1) in this section.

2.1 Jacobi-Davidson. For the standard eigenproblem, Jacobi-Davidson selects the approximate eigenvector from a *search subspace* span$\{\mathbf{V}\}$ that is expanded in each step. Each step consists of two parts. In the first part, the 'Davidson' part, the *projected eigenproblem*

$$(\mathbf{V}^*\mathbf{A}\mathbf{V} - \widetilde{\lambda}\,\mathbf{V}^*\mathbf{V})\,u = 0, \tag{3}$$

is solved and a solution $(u, \widetilde{\lambda})$ is selected. The *Ritz value* $\widetilde{\lambda}$ and the *Ritz vector* $\widetilde{\mathbf{q}} \equiv \mathbf{V}u$ form an approximate eigenvalue and eigenvector, respectively, with *residual* $\mathbf{r} \equiv (\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})\,\widetilde{\mathbf{q}}$ (we will assume that $\|\widetilde{\mathbf{q}}\|_2 = 1$). Observe that $\mathbf{r} \perp \widetilde{\mathbf{q}}$. In the second part, the 'Jacobi' part, the search subspace is expanded by a vector $\mathbf{v} \perp \widetilde{\mathbf{q}}$, that solves (approximately) the *correction equation*

$$\widetilde{\mathbf{q}}^*\mathbf{v} = 0 \quad \text{and} \quad (\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)\,\mathbf{v} = -\mathbf{r}. \tag{4}$$

The expanded search subspace is span$\{\mathbf{V}, \mathbf{v}\}$. In exact arithmetic, $\mathbf{V}$ is an orthogonal matrix, $\mathbf{V}^*\mathbf{V} = I$. We have used modified Gram-Schmidt in our computations for the construction of an orthogonal basis of the search subspace.

As observed in the introduction, if $\widetilde{\lambda}$ is replaced in the correction equation (4) by an eigenvalue λ, then the associated eigenvector is contained in the space spanned by $\mathbf{V}$ and the exact solution of the correction equation. Usually we have no better estimate for λ than the Ritz value $\widetilde{\lambda}$ and then the solution of the correction equation is the best we can do for expansion. If this correction equation (4) is solved exactly, then the speed of convergence for the selected Ritz values is asymptotically quadratical (cf. [20], [18]).

In our applications, the projected problem (3) will be of relatively small dimension, i.e., $\dim(\text{span}\{\mathbf{V}\}) \ll n$, and can be solved efficiently by standard algorithms for dense problems.

2.2 Practical selection and implicit restart. If we reduce the projected eigenproblem (3) to Schur form by the QR algorithm [6], then we can exploit the Schur form for the selection of a Ritz pair $(\tilde{\mathbf{q}}, \tilde{\lambda})$, and for restriction of the dimension of the subspace span$\{\mathbf{V}\}$, in the following way.

Suppose we are interested in the eigenpair(s) with eigenvalue close to some specified *target* value τ, and suppose that the Schur form of the *interaction matrix* $M \equiv \mathbf{V}^*\mathbf{A}\mathbf{V}$, given by

$$MU = US, \quad U^*U = I, \quad \text{and} \quad S \text{ upper triangular,}$$

is ordered such that

$$|S(1,1) - \tau| \leq |S(2,2) - \tau| \leq \cdots \leq |S(j,j) - \tau|, \tag{5}$$

where j is the dimension of span$\{\mathbf{V}\}$. Then

$$(\tilde{\mathbf{q}}, \tilde{\lambda}) \equiv (\mathbf{V}U(:,1), S(1,1)) \tag{6}$$

is the Ritz approximation corresponding to the projected system (3) with Ritz value closest to the target τ. Furthermore, $\mathbf{V}U(:,1:i)$, with $i < j$, spans the subspace that has the best information for the i eigenvalues closest to τ. Therefore, if we want to reduce the dimension of the search subspace from j to $j_{\min}$, $j_{\min} < j$, then we discard the columns $\mathbf{v}_{j_{\min}+1}$ through $\mathbf{v}_j$, and continue the JD algorithm with

$$\mathbf{V} = \mathbf{V}U(:,1\!:\!j_{\min}). \tag{7}$$

It is convenient that $\mathbf{V}U(:,1\!:\!j_{\min})$ is already orthogonal.

We refer to this reduction strategy as *implicit restart.*

REMARK 3 Our restart strategy follows similar ideas as in the Implicitly Restarted Arnoldi (IRA) [22]. However, in [22] implicit shifts are used to delete the *unwanted* part, instead of explicitly selecting the *wanted* portion of the Krylov subspace as we do. The situation for IRA is more complicated because the reduced search subspace has to be a Krylov subspace. For further details, see [22], [9].

A reordering algorithm for the Schur form can be found in, for instance, [23], [7], [14]; a Fortran implementation is available from LAPACK [1]. A simple MATLAB implementation for reordering with respect to a target value τ, is given in Appendix B. For completeness, a theoretical justification is given there as well.

In ALG. 1, an implementation of the JD algorithm for the standard eigenproblem is given in MATLAB-style. This implementation includes the implicit restart option based on ordered Schur forms. It also includes reordering, and is adjusted for one eigenvalue closest to a target τ. This "basic" algorithm will be extended for the computation of a number of eigenvalues in the neighborhood of τ (in §2.3), and preconditioning will be included (§2.4).

REMARK 4 In the implementations of our algorithms, we consider an approximate eigenpair $(\tilde{\mathbf{q}}, \tilde{\lambda})$ converged if the residual $\mathbf{r}$ is sufficiently small ($\|\mathbf{r}\| \leq \epsilon$); then $(\tilde{\mathbf{q}}, \tilde{\lambda})$ is a *"detected"* eigenpair. Weighted residuals (e.g., $\|\mathbf{r}\| \leq \epsilon\|\mathbf{A}\|$) or more sophisticated stopping criteria can be employed as well (cf. [4]).

In TAB. 1 we have listed the main computation-intensive ingredients per iteration of JD.

```
function [q, λ] = JD (A, τ, v_0, ε, j_max, j_min)
V = [ ]; V_A = [ ]; M = [ ];
j = 0; found = 0;
while ~ found,
    if j == 0,
        v = v_0;
    else
        — the correction equation —
        Solve v (approximately) from:
            q*v = 0 and
            (I − q q*)(A − λ I)(I − q q*) v = −r.
    end
    — the projected problem —
    v = mgs (V, v); v = v/||v||_2; v_A = Av;
    M = [ M, V*v_A; v*V_A, v*v_A];
    V = [V, v]; V_A = [V_A, v_A];
    [U, S] = schur (M); [U, S] = qrsort (τ, U, S);
    j = j + 1;
    — Ritz approximation —
    λ = S(1, 1); q = VU(:, 1);
    r = (A − λ I) q;
    found = (||r||_2 < ε),
    — "found and implicit restart" —
    if found,
        break;
    elseif j == j_max,
        — implicit restart —
        j = j_min; J = [1:j];
        V = VU(:, J); V_A = V_A U(:, J);
        S = S(J, J); M = S; U = I;
    end
end
```

JD returns an eigenpair $(\mathbf{q}, \lambda)$ of the matrix $\mathbf{A}$ with λ near the target τ. $\mathbf{v}_0$ is an initial guess, and ϵ is the stopping tolerance. $j_{\max}$ and $j_{\min}$ specify the dimension of the subspace $\mathbf{V}$ before and after implicit restart, respectively. schur is a MATLAB function that computes a Schur decomposition $MU = US$. The functions mgs (modified Gram-Schmidt) and qrsort (Sort Schur form) are given in Appendix B.

ALGORITHM 1. *JD with implicit restart.*

2.3 JDQR. Now we focus on the efficient computation of several eigenpairs. The idea is to use the JD algorithm for the computation of a *partial Schur form*, which is defined as follows (cf. [16]).

DEFINITION 1 A *partial Schur form* of dimension k for a matrix $\mathbf{A}$ is the following decomposition:

$$\mathbf{A}\mathbf{Q}_k = \mathbf{Q}_k R_k$$

where $\mathbf{Q}_k$ is an orthogonal $(n \times k)$-matrix, and R_k is an upper triangular $(k \times k)$-matrix. A column $\mathbf{q}_i$ of the matrix $\mathbf{Q}_k$ is a *Schur vector*, and the pair $(\mathbf{q}_i, \lambda_i)$, with $\lambda_i = R_k(i, i)$, is a *Schur*

Part	DOTS	AXPYS	MVs
The correction equation	variable		
The projected problem	$3j$	$j-1$	1 [a]
Ritz approximation	1	j	1 [b]
Restart	0	$2j_{\mathrm{min}}$	0

[a] If Krylov subspace methods are used to solve the correction equation, then the product $\mathbf{A}\mathbf{v}$ is often already available, as a side-product. No MV is needed in this part then.
[b] Instead of computing the residual $\mathbf{r}$ as $(\mathbf{A}-\widetilde{\lambda}\,\mathbf{I})\widetilde{\mathbf{q}}$, $\mathbf{r}$ may also be computed as $\mathbf{V}_A U(:,1)-\widetilde{\lambda}\widetilde{\mathbf{q}}$, where $\mathbf{V}_A\equiv\mathbf{A}\mathbf{V}$ (cf. ALG. 1); depending on the number of nonzeros in $\mathbf{A}$ and the value of j, this may be more efficient.

TABLE 1. *The computational costs of implicitly restarted JD per iteration. j is the dimension of* $\mathrm{span}\{\mathbf{V}\}$.

pair.

The diagonal entries of the matrix R_k represent eigenvalues of $\mathbf{A}$, and if (x,λ) is an eigenpair of R_k then $(\mathbf{Q}_k x,\lambda)$ is an eigenpair of $\mathbf{A}$.

Steps I, II, and III below represent the JDQR algorithm for computing a partial Schur form. A MATLAB implementation of JDQR is not given until § 2.4, where preconditioning for the correction equations is discussed.
I. For the first Schur pair, we apply the JD algorithm. This leads to a search subspace $\mathbf{V}$. For the interaction matrix $M\equiv\mathbf{V}^*\mathbf{A}\mathbf{V}$ we compute an ordered Schur form $MU=US$. The first Schur pair of the projected problem is taken as the approximation to a Schur pair of the original large eigenproblem. This is used for the correction equation (4), of which the (approximate) solution $\mathbf{v}$ gives the expansion of the subspace ($\mathbf{V}$ is expanded with the orthogonal complement $\mathbf{v}'$ of $\mathbf{v}$ to $\mathbf{V}$). With the expanded subspace $\mathbf{V}=[\mathbf{V},\mathbf{v}']$ we construct again the Schur form for the corresponding interaction matrix, and this process is repeated until a Schur pair has been detected. Upon convergence, when $\mathrm{span}\{\mathbf{V}\}$ is of dimension j, say, the subspace reduced by the detected Schur vector, $\mathbf{V}=\mathbf{V}U(:,2\!:\!j)$, can be used as the starting subspace for a new Schur pair.
II. Now suppose that $k-1$ Schur pairs have been detected, i.e., we already have the partial Schur form $\mathbf{A}\mathbf{Q}_{k-1}=\mathbf{Q}_{k-1}R_{k-1}$. We want to expand $\mathbf{Q}_{k-1}$, with a suitable $\mathbf{q}$, so that

$$\mathbf{A}\begin{bmatrix}\mathbf{Q}_{k-1} & \mathbf{q}\end{bmatrix}=\begin{bmatrix}\mathbf{Q}_{k-1} & \mathbf{q}\end{bmatrix}\begin{bmatrix}R_{k-1} & s\\ 0 & \lambda\end{bmatrix}. \tag{8}$$

The new Schur pair $(\mathbf{q},\lambda)$ should satisfy

$$\mathbf{Q}_{k-1}^*\mathbf{q}=0 \quad\text{and}\quad (\mathbf{A}-\lambda\,\mathbf{I})\mathbf{q}-\mathbf{Q}_{k-1}s=\mathbf{0},$$

or, since $s=\mathbf{Q}_{k-1}^*(\mathbf{A}-\lambda\,\mathbf{I})\,\mathbf{q}$,

$$\mathbf{Q}_{k-1}^*\mathbf{q}=0 \quad\text{and}\quad (\mathbf{I}-\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{A}-\lambda\,\mathbf{I})\,\mathbf{q}=\mathbf{0}.$$

Hence, $(\mathbf{q},\lambda)$ satisfies

$$\mathbf{Q}_{k-1}^*\mathbf{q}=0 \quad\text{and}\quad (\mathbf{I}-\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{A}-\lambda\,\mathbf{I})(\mathbf{I}-\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,\mathbf{q}=\mathbf{0}, \tag{9}$$

and the Schur pair $(\mathbf{q},\lambda)$ is therefore also an eigenpair of the *deflated matrix*

$$(\mathbf{I}-\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\mathbf{A}(\mathbf{I}-\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*). \tag{10}$$

We solve this eigenproblem, by the procedure sketched in I., for the deflated matrix (10). More precisely, the JD algorithm for the deflated matrix (10) constructs a subspace span$\{\mathbf{V}\}$ for finding approximate eigenpairs, and $\mathbf{V}$ is an orthogonal matrix such that $\mathbf{V}^*\mathbf{Q}_{k-1} = 0$. For the *deflated interaction matrix* M we have

$$M \equiv \mathbf{V}^*(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\mathbf{A}(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\mathbf{V} = \mathbf{V}^*\mathbf{A}\mathbf{V}. \tag{11}$$

The ordered Schur form $MU = US$ (see (5)) gives an approximation $(\widetilde{\mathbf{q}}, \widetilde{\lambda}) \equiv (\mathbf{V}U(:,1), S(1,1))$ for a wanted eigenpair of the deflated matrix (10). Then, according to the Jacobi-Davidson approach, the search subspace span$\{\mathbf{V}\}$ is expanded by the orthogonal complement of $\mathbf{v}$ to $\mathbf{V}$, where $\mathbf{v}$ is the (approximate) solution of the *deflated correction equation*

$$\begin{cases} \mathbf{Q}_{k-1}^*\mathbf{v} = 0, \quad \widetilde{\mathbf{q}}^*\mathbf{v} = 0 \quad \text{and} \\ (\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)\,\mathbf{v} = -\mathbf{r}, \end{cases} \tag{12}$$

where $\mathbf{r} \equiv (\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,\widetilde{\mathbf{q}}$ and note that $\mathbf{Q}_{k-1}^*\widetilde{\mathbf{q}} = 0$.

Note that the projections in (12) can be subdivided into two parts; the part $(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)$ associated with Jacobi-Davidson, and the deflation part $(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)$. Observe also that $\mathbf{Q}_{k-1}^*\mathbf{r} = 0$ and $\widetilde{\mathbf{q}}^*\mathbf{r} = 0$.

III. When the Schur pair $(\widetilde{\mathbf{q}}, \widetilde{\lambda})$ is sufficiently close to $(\mathbf{q}, \lambda)$, then we may continue for still another Schur pair. In that case, the matrix $\mathbf{V}$ of dimension $n \times j$, say, is reduced to $\mathbf{V}U(:, 2\!:\!j)$, in order to obtain a subspace orthogonal to span$\{\mathbf{Q}_{k-1}, \widetilde{\mathbf{q}}\}$, and we continue the process.

REMARK 5 Two deflation techniques can be found in literature for subspace methods like Arnoldi. They are referred to as explicit and implicit deflation (cf., e.g., [16, Ch.VI, §2.3]).
In explicit deflation, the computation is continued with a deflated matrix after detection of Schur vectors. For efficiency reasons, it is sometimes suggested to use $\mathbf{A} - \mathbf{Q}\,\Lambda\,\mathbf{Q}^*$ (Schur-Wielandt deflation), rather than our more stable representation $(\mathbf{I} - \mathbf{Q}\,\mathbf{Q}^*)\mathbf{A}(\mathbf{I} - \mathbf{Q}\,\mathbf{Q}^*)$.
In implicit deflation, each new vector for the search subspace is generated with $\mathbf{A}$ itself and is then made orthogonal to the detected Schur vectors, before adding it to the search subspace. Our approach, is a mixture of both techniques. In the correction equation we use the explicitly deflated matrix $(\mathbf{I} - \mathbf{Q}\,\mathbf{Q}^*)\mathbf{A}(\mathbf{I} - \mathbf{Q}\,\mathbf{Q}^*)$. Since the solutions of the deflated correction equations are orthogonal to the detected Schur vectors, there is no need to use the deflated matrix for the deflated interaction matrix M; this is computed as $M = \mathbf{V}^*\mathbf{A}\mathbf{V}$ (which is implicit deflation, cf. (11)).

When we use Krylov subspace methods for the solution of the correction equation, then the work with either of the two representations of the deflated matrix is the same (cf. REMARK 10).

Exclusively implicit deflation is possible as well: solve the correction equation approximately with the non-deflated $\mathbf{A}$ and put the resulting solution orthogonal to the detected Schur vectors. In this approach we avoid expensive matrix vector multiplications, but explicit deflation appears to improve the condition number of the linear system, and this leads to a faster converging process for the correction equation (4). The decrease of the number of iteration steps, for the correction equation, appears often to compensate for the more expensive multiplications (for a numerical illustration of this, see §4.6).

Moreover, the deflated correction equations (12) appear to lead to more stable results. This can be understood as follows. Without deflation the resulting solution of the correction equation may have a significant component in the space spanned by the detected Schur vectors. By subtracting this component (as in implicit deflation) cancellation may occur. If we work with an explicitly deflated matrix, such cancellation is avoided.

REMARK 6 As in implicitly deflated Arnoldi methods, the accuracy of an approximate Schur pair in our method does not only depend on the norm of the residual and the condition number of the pair, but also on the approximation errors in the previously detected Schur pairs (cf. e.g., [16, Ch.IV, §2.5] and [9, §6.4.1]): in the derivation of the algorithms it is assumed that $\mathbf{V}^*\mathbf{Q} = 0$ implies $\mathbf{V}^*\mathbf{A}\mathbf{Q} = 0$, which is true for exact Schur vectors. In practice, span$\{\mathbf{A}\mathbf{Q}\}$ will not be contained in span$\{\mathbf{Q}\}$.

REMARK 7 Note that the vector $\mathbf{v}$ obtained from the correction equation (12) gives the near optimal expansion of the search subspace $\mathbf{V}$ and hence will, in general, not make a pathological small angle with the current search subspace. This implies that one cycle of modified Gram-Schmidt suffices for the computation of the orthogonal complement of $\mathbf{v}$ to $\mathbf{V}$.

2.4 Preconditioning. In this section we will discuss preconditioning for the correction equation. Preconditioning is not straight forward, because of the projections involved. We will derive explicit expressions for left and right preconditioned correction equations.

In each iteration step we need to solve a deflated correction equation (12), for a given $\widetilde{\mathbf{q}}$ and $\widetilde{\lambda}$ (cf. (12)). For the approximate solution of this equation we may use a Krylov subspace method, e.g., GMRES [17], or BiCGstab(ℓ) [19]. The rate of convergence and the efficiency of Krylov subspace methods is often improved by preconditioning. The identification of an effective preconditioner may be a problem. For instance, for interior eigenvalues the construction of an effective incomplete LU-factorization [10], [8] for $\mathbf{A} - \widetilde{\lambda}\mathbf{I}$ may require much fill-in[2], which makes the construction expensive. As we will argue in §2.7, it may be a good strategy to compute a preconditioner $\mathbf{K}$ for $\mathbf{A} - \tau\mathbf{I}$ for a fixed value of τ only, and to use

$$\widetilde{\mathbf{K}} \equiv (\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\mathbf{K}(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*) \tag{13}$$

as the preconditioner for various $\widetilde{\mathbf{q}}$ and $\widetilde{\lambda}$. Note that the projections on $\mathbf{K}$ are necessary to let $\widetilde{\mathbf{K}}$ operate on the proper subspace (cf. [18]).

We will now derive the expressions for the preconditioned correction equation. For convenience, we introduce the following notations:

NOTATION 1

$\widetilde{\mathbf{Q}}_k \equiv [\,\mathbf{Q}_{k-1}, \widetilde{\mathbf{q}}\,]$,	the matrix $\mathbf{Q}_{k-1}$ expanded by the approximate Schur vector $\widetilde{\mathbf{q}}$,
$\widetilde{\mathbf{Y}}_k \equiv \mathbf{K}^{-1}\widetilde{\mathbf{Q}}_k$,	the matrix of the preconditioned "Schur" vectors,
$\widetilde{H}_k \equiv \widetilde{\mathbf{Q}}_k^*\widetilde{\mathbf{Y}}_k$,	the projected preconditioner $\widetilde{\mathbf{Q}}_k^*\,\mathbf{K}^{-1}\widetilde{\mathbf{Q}}_k$.

A typical use of the preconditioner in a Krylov subspace method, for our purposes looks like:

$$\begin{array}{l} \textit{solve } \mathbf{t}, \textit{ with } \widetilde{\mathbf{Q}}_k^*\mathbf{t} = 0, \textit{ from} \\ (\mathbf{I} - \widetilde{\mathbf{Q}}_k\widetilde{\mathbf{Q}}_k^*)\,\mathbf{K}\,(\mathbf{I} - \widetilde{\mathbf{Q}}_k\widetilde{\mathbf{Q}}_k^*)\,\mathbf{t} = \mathbf{s} \\ \textit{for } \mathbf{s}, \textit{ with } \widetilde{\mathbf{Q}}_k^*\mathbf{s} = 0, \end{array} \tag{14}$$

where $\mathbf{K}$ is a preconditioner for $\mathbf{A} - \tau\,\mathbf{I}$.

The following identities, formulated in a lemma, are useful for obtaining an explicit expression for the solution of (14).

[2] These incomplete factorizations have not been designed for linear systems related to eigenproblems. The solutions for which these factorizations are most effective are usually rather smooth, which means that components of slowly varying eigenvectors are favored by the preconditioning.

LEMMA 1 *If $\widetilde{H}_k$ is nonsingular, then*

$$(\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)(\mathbf{I} - \widetilde{\mathbf{Q}}_k \widetilde{\mathbf{Q}}_k^*) = (\mathbf{I} - \widetilde{\mathbf{Q}}_k \widetilde{\mathbf{Q}}_k^*), \tag{15}$$

$$(\mathbf{I} - \widetilde{\mathbf{Q}}_k \widetilde{\mathbf{Q}}_k^*)(\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*) = (\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*), \tag{16}$$

$$(\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\,\mathbf{K}^{-1}(\mathbf{I} - \widetilde{\mathbf{Q}}_k \widetilde{\mathbf{Q}}_k^*) = (\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\,\mathbf{K}^{-1}. \tag{17}$$

PROOF These identities follow straight forward from expanding the products. For example:

$$\begin{aligned}(\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)(\mathbf{I} - \widetilde{\mathbf{Q}}_k \widetilde{\mathbf{Q}}_k^*) &= (\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*) - (\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\,\widetilde{\mathbf{Q}}_k \widetilde{\mathbf{Q}}_k^* \\ &= (\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*) - (\widetilde{\mathbf{Q}}_k \widetilde{\mathbf{Q}}_k^* - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*) \\ &= (\mathbf{I} - \widetilde{\mathbf{Q}}_k \widetilde{\mathbf{Q}}_k^*). \qquad \square\end{aligned}$$

The following lemma gives us an explicit expression for the solution $\mathbf{t}$ of (14), in terms of easily computable matrix vector products with $\widetilde{H}_k^{-1}$ and $\mathbf{K}^{-1}$. The lemma generalizes PROP. 7.5 in [18]. Note that $\widetilde{H}_k$ is of small dimension, so it is easily inverted. $\mathbf{K}$ has never to be explicitly inverted, instead $\mathbf{v} = \mathbf{K}^{-1}\mathbf{s}$ can be computed by solving $\mathbf{v}$ from $\mathbf{K}\mathbf{v} = \mathbf{s}$.

LEMMA 2 *If $\widetilde{H}_k$ is nonsingular, then the solution $\mathbf{t}$ of equation (14) is given by*

$$\mathbf{t} = (\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\,\mathbf{K}^{-1}\mathbf{s}, \tag{18}$$

PROOF The expression (18) follows from multiplying (14) by $(\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\,\mathbf{K}^{-1}$, and using identities of LEMMA 1. $\square$

REMARK 8 If one stores the matrix $\mathbf{Y}_{k-1} \equiv \mathbf{K}^{-1}\mathbf{Q}_{k-1}$ of preconditioned Schur vectors, one only has to compute the last column $\mathbf{K}^{-1}\widetilde{\mathbf{q}}$ of the matrix $\widetilde{\mathbf{Y}}_k = [\,\mathbf{Y}_{k-1}\,, \mathbf{K}^{-1}\widetilde{\mathbf{q}}\,]$, at each iteration step. Furthermore, when storing the projected preconditioner $H_{k-1} \equiv \mathbf{Q}\,^*_{k-1}\,\mathbf{K}^{-1}\mathbf{Q}_{k-1}$, only the last column and last row of

$$\widetilde{H}_k = \begin{bmatrix} H_{k-1} & \mathbf{Q}_k^*\mathbf{K}^{-1}\widetilde{\mathbf{q}} \\ \widetilde{\mathbf{q}}^*\mathbf{Y}_k & \widetilde{\mathbf{q}}^*\mathbf{K}^{-1}\widetilde{\mathbf{q}} \end{bmatrix}$$

have to be computed in an iteration step.

REMARK 9 If the preconditioner $\mathbf{K}$ is indefinite, then the matrix $\widetilde{H}_k$ may become singular, for an "unlucky" choice of approximate Ritz pair $(\widetilde{\mathbf{q}}, \widetilde{\lambda})$. This causes a breakdown, but it never happened in our experiments. The breakdown may be cured by selecting a different nearby approximating Ritz pair $(\widetilde{\mathbf{q}}', \widetilde{\lambda}')$ temporarily, for the current Jacobi-Davidson iteration.

Left preconditioning. From LEMMA 2, it follows that the left preconditioned correction equation is equivalent with

$$\widetilde{\mathbf{Q}}_k^*\mathbf{v} = 0 \quad \text{and} \quad (\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\,\mathbf{K}^{-1}(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})(\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\,\mathbf{v} = -\widehat{\mathbf{r}}, \tag{19}$$

where $\widehat{\mathbf{r}} \equiv (\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\,\mathbf{K}^{-1}\mathbf{r}$ (cf. [18, Th. 7.4]).

Note that the projection has to be applied explicitly to the residual. For the unpreconditioned case there was no need for explicit projection, since there, the fact that the residual is associated with a deflated matrix and with a Ritz pair, implied orthogonality to $\widetilde{\mathbf{Q}}_k$.

Observe that, for $\mathbf{K} = \mathbf{I}$, this equation (19) is equivalent to the one in (12).

Right preconditioning. Also right preconditioning for the correction equation may be used. With $\widehat{\mathbf{Y}}_k \equiv (K^*)^{-1}\widetilde{\mathbf{Q}}_k$, this leads to

$$\widehat{\mathbf{Y}}_k^*\widehat{\mathbf{v}} = 0 \quad \text{and} \quad (\mathbf{I} - \widetilde{\mathbf{Q}}_k \widetilde{H}_k^{-1} \widehat{\mathbf{Y}}_k^*)(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})\,\mathbf{K}^{-1}(\mathbf{I} - \widetilde{\mathbf{Q}}_k \widetilde{H}_k^{-1} \widehat{\mathbf{Y}}_k^*)\widehat{\mathbf{v}} = -\widehat{\mathbf{r}}, \tag{20}$$

where $\widehat{\mathbf{r}} \equiv (\mathbf{I} - \widetilde{\mathbf{Q}}_k \widetilde{H}_k^{-1} \widehat{\mathbf{Y}}_k^*)\,\mathbf{r}$. The vector $\mathbf{v}$ can be obtained from $\widehat{\mathbf{v}}$ by $\mathbf{v} = \mathbf{K}^{-1}\widehat{\mathbf{v}}$.

REMARK 10 If one uses Krylov subspace methods for solving the second equation in (19), then one encounters matrix vectors products of the form

$$(\mathbf{I}-\widetilde{\mathbf{Y}}_k\,\widetilde{H}_k^{-1}\widetilde{\mathbf{Q}}_k^*)\mathbf{K}^{-1}(\mathbf{A}-\widetilde{\lambda}\,\mathbf{I})(\mathbf{I}-\widetilde{\mathbf{Y}}_k\,\widetilde{H}_k^{-1}\widetilde{\mathbf{Q}}_k^*)\,\mathbf{t}, \tag{21}$$

with $\mathbf{t}$ of the form $\mathbf{t}=(\mathbf{I}-\widetilde{\mathbf{Y}}_k\,\widetilde{H}_k^{-1}\widetilde{\mathbf{Q}}_k^*)\,\mathbf{s}$. Then, obviously, $\widetilde{\mathbf{Q}}_k^*\mathbf{t}=0$, and for the approximate solution $\mathbf{v}$ it holds that $\widetilde{\mathbf{Q}}_k^*\mathbf{v}=0$, if that is the case for the initial guess. Moreover, the projection $(\mathbf{I}-\widetilde{\mathbf{Y}}_k\,\widetilde{H}_k^{-1}\widetilde{\mathbf{Q}}_k^*)$ in front of $\mathbf{t}$ in (21) is redundant then, and (21) reduces to

$$(\mathbf{I}-\widetilde{\mathbf{Y}}_k\,\widetilde{H}_k^{-1}\widetilde{\mathbf{Q}}_k^*)\mathbf{K}^{-1}(\mathbf{A}-\widetilde{\lambda}\,\mathbf{I})\,\mathbf{t}, \tag{22}$$

A similar remark applies for (20).

A pseudo-code for the preconditioned Jacobi-Davidson QR algorithm is given in ALG. 2. In TAB. 2 we have listed the main computational ingredients per iteration of JDQR.

Part	DOTS	AXPYS	MVs	K
The correction equation	variable			
The projected problem	$3j$	$j-1$	1 [a]	0
Ritz approximation	$k+1$	$j+k$	1 [b]	1
Found	0	$2j-2$	0	0
Restart	0	$2j_{\text{min}}$	0	0

[a] If Krylov subspace methods are used to solve the correction equation, then the product $\mathbf{Av}$ is often already available as a side-product. No MV is needed in this part then.
[b] Instead of computing the residual $\mathbf{r}$ as $(\mathbf{A}-\widetilde{\lambda}\,\mathbf{I})\widetilde{\mathbf{q}}$, $\mathbf{r}$ may also be computed as $\mathbf{V}_A U(:,1)-\widetilde{\lambda}\,\widetilde{\mathbf{q}}$, where $\mathbf{V}_A\equiv\mathbf{AV}$ (cf. ALG. 2); depending on the number of nonzeros in $\mathbf{A}$ and the value j, this may be more efficient.

TABLE 2. *The computational costs of JDQR per iteration. The integers j and k are the dimensions of* $\operatorname{span}\{\mathbf{V}\}$ *and* $\operatorname{span}\{\mathbf{Q}\}$*, respectively.*

2.5 The selection of Ritz pairs. At each iteration step of the Jacobi-Davidson method, an approximate eigenpair $(\widetilde{\mathbf{q}},\widetilde{\lambda})$ has to be selected from the Ritz pairs $(\mathbf{V}u_i,\theta_i)$, that are solutions of the projected eigenproblem $(\mathbf{V}^*\mathbf{AV}-\theta\,I)\,u=0$. For the selected Ritz pair, the correction equation provides the optimal expansion of the search subspace. However, the speed of convergence also depends on the ability to identify the most suitable Ritz pair (see also [20, §5]).

A poor choice of the Ritz pair will lead to an expansion of the search subspace in a poor direction. This affects the convergence to the wanted eigenvector, and this is a situation that we want to avoid. For limiting the dimension of the search subspace (cf. §2.2), it is even more important to avoid poor selections, because by discarding correct information, convergence can be hampered, and even completely destroyed.

Suppose we are interested in the eigenpair(s) with eigenvalue close to some specified *target* value τ. By selecting the pair with Ritz value closest to the target value, we hope to have the best approximation that is available in the search subspace. With respect to extremal eigenvalues, at least for standard normal eigenproblems, this approach can be justified theoretically as follows.

Recall that, for normal matrices, Ritz values are convex combinations of eigenvalues, and consider the Ritz value $\widetilde{\lambda}$ that is close to an extremal eigenvalue λ (an extreme point of the field of values). Then the λ-component in the convex combination for $\widetilde{\lambda}$ will be the largest with a size depending on the separation of λ and the distance of λ to $\widetilde{\lambda}$. Correspondingly, among all

```
function [ Q, R ] = JDQR (A, K, τ, v_0, ε, k_max, j_max, j_min)
Q = [ ]; R = [ ]; Y = [ ]; H = [ ];
V = [ ]; V_A = [ ]; M = [ ];
k = 0; j = 0;
while k < k_max,
    if j == 0,
        v = v_0;
    else
        — the correction equation —
        r = (I − Ỹ H̃^{-1} Q̃^*) K^{-1} r;
        Solve v (approximately) from:
            Q̃^* v = 0    and
            (I − Ỹ H̃^{-1} Q̃^*) K^{-1}(A − λ I)(I − Ỹ H̃^{-1} Q̃^*) v = −r.
    end
    — the projected problem —
    v = mgs (V, v); v = v/||v||_2; v_A = Av;
    M = [ M, V^* v_A; v^* V_A, v^* v_A ];
    V = [ V, v]; V_A = [ V_A, v_A ];
    [ U, S ] = schur (M); [ U, S ] = qrsort (τ, U, S);
    j = j + 1; found = 1;
    while found,
        — Ritz approximation —
        λ = S(1, 1); q = V U(:, 1); y = K^{-1} q;
        r = (A − λ I) q; [ r, s ] = mgs (Q, r);
        Q̃ = [ Q, q ]; Ỹ = [ Y, y ]; H̃ = [ H, Q^* y; q^* Y, q^* y ];
          ⋮
        "found and implicit restart part", see ALG. 3
          ⋮
    end
end
```

JDQR returns a partial Schur form $(\mathbf{Q}, R)$ of the matrix $\mathbf{A}$ of dimension $k_{\max}$ with eigenvalues near the target τ. $\mathbf{K}$ is a preconditioner for $\mathbf{A} - \tau\,\mathbf{I}$, $\mathbf{v}_0$ is an initial guess, and ϵ is the stopping tolerance. $j_{\max}$ and $j_{\min}$ specify the dimension of the subspaces $\mathbf{V}$ before and after implicit restart, respectively. schur is a MATLAB function that computes a Schur decomposition. The functions mgs (modified Gram-Schmidt) and qrsort (Sort Schur form) are given in Appendix B.

ALGORITHM 2. *Preconditioned JDQR.*

eigenvector components of the Ritz vector associated with $\widetilde{\lambda}$, the component of the Ritz vector, in the eigenvector direction associated with λ, will be the largest in modulus (the components for eigenvectors are square roots of the corresponding components of eigenvalues). If $\widetilde{\lambda}$ is still not close to λ, as in the initial stage of the process, the fact that (for symmetric problems) extremal eigenvalues optimize the Rayleigh quotient on the whole space, while extremal Ritz values optimize this quotient on the search subspace, may provide some confidence that the extremal Ritz values are the most appropriate approximations for the extremal eigenvalues. For interior eigenvalues λ, convex combinations of eigenvalues, in which the λ-component is small or even zero, can be close (or even equal) to λ: the Ritz value closest to the target value may

```
found = (‖r‖₂ < ϵ) & (j > 1 | k = k_max − 1);
if found,
    — found —
    Q = Q̃;  R = [ R, s; zeros (1, k), λ ];
    k = k + 1;   if k == k_max, break; end
    Y = Ỹ;  H = H̃;
    J = [ 2:j ];  j = j − 1;
    V = V U(:, J);  V_A = V_A U(:, J);
    S = S(J, J);  M = S;  U = I;
elseif j == j_max,
    — implicit restart —
    j = j_min;  J = [ 1:j ];
    V = V U(:, J);  V_A = V_A U(:, J);
    S = S(J, J);  M = S;  U = I;
end
```

ALGORITHM 3. *"Found and implicit restart part" of JDQR.*

be associated with a Ritz vector of which the angle with the target eigenvector is much larger than for other Ritz vectors. In [12], [13], [20] it is suggested to use *harmonic Ritz* pairs for these eigenvalues. As we will see in §3.5.1, the harmonic Ritz value closest to the target value can be viewed as an extremal Ritz values for a related problem. Therefore, we also adopt this approach. However, since the definition of harmonic Ritz values fits better in the context of generalized eigenproblems, we postpone our discussion on this subject to our treatment of generalized eigenproblems in §3.1. For the standard problem, we consider also a strategy for selecting standard Ritz values closest to the target value, also if the target is in the interior of the spectrum.

2.5.1 Identification of suitable Ritz values (tracking). If the Ritz vector in the previous iteration is already a fair approximation, then the norm of the residual gives information on the selected Ritz vector in the current step: in case of a poor selection, the new residual can be much larger than the previous one. It would then require additional computational work to find a Ritz pair with small residual norm (and still close enough to the target τ). A cheap alternative in this case is to select a Ritz value that is close to a previously accepted one (and forget about τ). In our experiments we have replaced in such cases the target by the Ritz value that is selected and accepted in the previous step, where we consider a Ritz value *acceptable* if the associated residual is smaller than some specified threshold ϵ_{tr}. After convergence of the Ritz pair, the original target value is restored at the start of the computation for the next eigenpair. This *tracking strategy* does not require any additional computational costs per step, while it appears to reduce the number of steps significantly.

In view of the discussion in §2.5 for extremal eigenvalues, improvement for these eigenvalues may not expected with the tracking strategy for normal problems.

2.6 Notes on the speed of convergence. The JDQR algorithm has nice properties with respect to the overall performance. While adjusting for one Schur pair, the subspace span$\{\mathbf{V}\}$ also accumulates components for other Schur pairs. As a result, after one Schur pair has been detected, other Schur pairs may follow more quickly than after a complete restart. These components will appear in a similar way as for the Shift-and-Invert Arnoldi [16] process, with a shift $\widetilde{\lambda}$ for a (deflated) eigenproblem, as can be understood as follows.

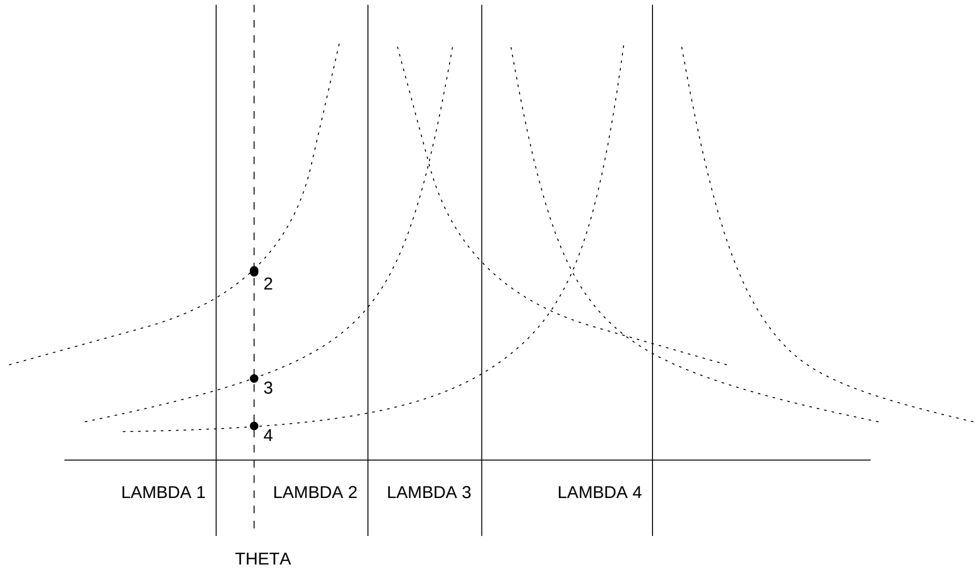


FIG. 1. *Amplification factors of eigenvectors*

For simplicity, suppose that $\mathbf{A}$ has a complete set of eigenpairs $(\mathbf{x}_i, \lambda_i)$ with $\lambda_1 < \lambda_2 < \cdots < \lambda_n$ and that we are trying to find an approximation $(\widetilde{\mathbf{q}}, \widetilde{\lambda})$ for $(\mathbf{x}_1, \lambda_1)$. The exact solution of (4) is given by

$$\mathbf{v} = -\widetilde{\mathbf{q}} + (\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})^{-1}\widetilde{\mathbf{q}}\epsilon, \tag{23}$$

with $\epsilon = 1/(\widetilde{\mathbf{q}}^*(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})^{-1}\widetilde{\mathbf{q}})$ (cf. [20, §4.1]). Writing $\widetilde{\mathbf{q}}$ as $\sum \gamma_i \mathbf{x}_i$, it follows that

$$(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})^{-1}\widetilde{\mathbf{q}} = \sum_i \frac{\gamma_i}{\lambda_i - \widetilde{\lambda}}\,\mathbf{x}_i. \tag{24}$$

We may assume, without loss of generality, that $\gamma_i \neq 0$, because $\widetilde{\mathbf{q}}$ is a Ritz vector which means that $\gamma_i = 0$ either if $\angle(\mathbf{x}_i, \mathbf{V}) = 0$ or $\pi/2$. The latter case is unlikely to happen, due to rounding errors, and the first case indicates full convergence.

Hence, eigenvector components corresponding to eigenvalues closer to $\widetilde{\lambda}$ will be amplified more in $(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})^{-1}\widetilde{\mathbf{q}}$. The component orthogonal to $\widetilde{\mathbf{q}}$ is used as an expansion for $\mathbf{V}$ and thus, as soon as $\widetilde{\mathbf{q}}$ has a large component in the direction of $\mathbf{x}_1$, say that the angle is less than $\pi/4$, then necessarily components other than $\mathbf{x}_1$ become dominant. That is

$$\mathbf{v} \sim \sum_{i \neq 1} \frac{\gamma_i}{\lambda_i - \widetilde{\lambda}}\,\mathbf{x}_i. \tag{25}$$

In FIG. 1 we have illustrated this phenomenon. The bullets represent the amplification factors $1/|\lambda_i - \widetilde{\lambda}|$ for components in the direction of $\mathbf{x}_i$ $(i = 2, 3, 4)$; `THETA` represents $\widetilde{\lambda}$.
In the subsequent iterations similar amplifications will occur and the closer λ_i is to $\widetilde{\lambda}$, the more rapid the angle $\angle(\mathbf{x}_i, \mathbf{V})$ will decrease.

This argument is repetitive: if the angle $\angle(\mathbf{x}_2, \mathbf{V})$ becomes very small, then the corresponding γ_2 will be very small and other components, due to orthogonalization, will become more dominant.

Consequently, while the process converges to a Schur pair, the search subspace $\mathbf{V}$ will provide good initial approximations for the nearby Schur pairs. Moreover, slow convergence during one stage may be compensated for by faster convergence in the next stage, because the subspace span$\{\mathbf{V}\}$ will be enriched with more components of other Schur pairs, due to repeated amplifications. This is observed in our numerical experiments, see §4.

2.7 The quality of the deflated preconditioner. Even when the preconditioner $\mathbf{K}$ is constructed for a fixed τ, then the correction equation still involves projections that become more expensive after each Schur pair that has been detected, but this does not necessarily lead to a more expensive computational process (compared with explicit restart). When iterative solvers are used, they may converge faster because the field of values of the projected operator $(\mathbf{I}-\mathbf{Q}_k\mathbf{Q}_k^*)(\mathbf{A}-\widetilde{\lambda}\,\mathbf{I})(\mathbf{I}-\mathbf{Q}_k\mathbf{Q}_k^*)$ is contained in the field of values of $\mathbf{A}-\widetilde{\lambda}\,\mathbf{I}$, and that may be smaller, specially after exterior eigenvalues have been detected.

The projections may also have a positive effect on the preconditioner.
With $(\mathbf{A}-\tau\,\mathbf{I})=\mathbf{K}-\mathbf{R}$ it follows that

$$\begin{aligned}(26)\quad &(\mathbf{I}-\mathbf{Q}_k\mathbf{Q}_k^*)(\mathbf{A}-\widetilde{\lambda}\,\mathbf{I})(\mathbf{I}-\mathbf{Q}_k\mathbf{Q}_k^*) = \\ &\quad(\mathbf{I}-\mathbf{Q}_k\mathbf{Q}_k^*)\,\mathbf{K}\,(\mathbf{I}-\mathbf{Q}_k\mathbf{Q}_k^*)-(\mathbf{I}-\mathbf{Q}_k\mathbf{Q}_k^*)\,\mathbf{R}\,(\mathbf{I}-\mathbf{Q}_k\mathbf{Q}_k^*)-(\tau-\widetilde{\lambda})(\mathbf{I}-\mathbf{Q}_k\mathbf{Q}_k^*),\end{aligned}$$

We see that, on the one hand, the preconditioning error is enlarged by a small shift $(\tau-\widetilde{\lambda})\mathbf{I}$, but on the other hand, the projections diminish the error by filtering out the detected Schur vectors. If the error $\mathbf{R}$ is large with respect to eigenvectors corresponding to eigenvalues near τ, then the projected error $(\mathbf{I}-\mathbf{Q}_k\mathbf{Q}_k^*)\,\mathbf{R}\,(\mathbf{I}-\mathbf{Q}_k\mathbf{Q}_k^*)$ will be significantly smaller, and the only penalty is a (small) shift due to $\tau-\widetilde{\lambda}$. It seems plausible (cf. [26, Ch. IV]) that this will not lead to a significantly less effective preconditioner, and it may help to explain the effectiveness of a fixed preconditioner running with JDQR in our experiments.

3 The generalized eigenproblem. We now consider the generalized eigenproblem (2) and derive the JDQZ algorithm along the lines as for JDQR.

CONVENTION 1 We denote a generalized eigenvalue of the matrix pair $(\mathbf{A},\mathbf{B})$ as a pair $\langle\alpha,\beta\rangle$. 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 meaning– and useful [11], [15], [24, Ch.VI].

REMARK 11 Observe that, for each $\gamma\neq 0$, the pairs $\langle\alpha,\beta\rangle$ and $\langle\gamma\alpha,\gamma\beta\rangle$ correspond to the same generalized eigenvalue. Rather than scaling the coefficients of $\langle\alpha,\beta\rangle$ in our algorithms (for instance, such that $\beta\in[0,1]$ and $\beta^2+|\alpha|^2=1$), we follow the advise in [11], and we show the results as produced by the QZ-algorithm: the size of α and β may give valuable information on the conditioning of the computed eigenpair. However, in the construction of our algorithm, scaling does play a role.

3.1 Jacobi-Davidson. For standard eigenproblems, the search subspace and the test subspace in Jacobi-Davidson are identical, but for generalized problems, other choices appear to be more natural (cf. [18]). A similar observation applies to the projectors in the correction equation.

As appears from the analysis in [18], for asymptotic quadratic speed of convergence we are restricted in our choices, although there are still a number of possibilities. As we will see, the computation of a *partial generalized Schur form*, will lead naturally to basically two specific choices for the test subspace, and to two related choices for the projectors in the correction equation.

For generalized eigenproblems, a partial generalized Schur form is defined as follows.

DEFINITION 2 A *partial generalized Schur form* of dimension k for a matrix pair $(\mathbf{A},\mathbf{B})$ is the decomposition

$$(27)\qquad \mathbf{A}\mathbf{Q}_k=\mathbf{Z}_kS_k,\quad \mathbf{B}\mathbf{Q}_k=\mathbf{Z}_kT_k,$$

where $\mathbf{Q}_k$ and $\mathbf{Z}_k$ are orthogonal $(n \times k)$-matrices, and S_k and T_k are upper triangular $(k \times k)$-matrices. A column $\mathbf{q}_i$ of $\mathbf{Q}_k$ is referred to as a *generalized Schur vector*, and we refer to a pair $(\mathbf{q}_i, \langle \alpha_i, \beta_i \rangle)$, with $\langle \alpha_i, \beta_i \rangle = \langle S_k(i,i), T_k(i,i) \rangle$ as a *generalized Schur pair.*

The formulation in (27) is equivalent with

$$\mathbf{Z}_k^* \mathbf{A} \mathbf{Q}_k = S_k \quad \text{and} \quad \mathbf{Z}_k^* \mathbf{B} \mathbf{Q}_k = T_k. \tag{28}$$

Furthermore, if $(x, \langle \alpha, \beta \rangle)$ is a generalized eigenpair of (S_k, T_k) then $(\mathbf{Q}_k x, \langle \alpha, \beta \rangle)$ is a generalized eigenpair of $(\mathbf{A}, \mathbf{B})$.

For presentation purposes, we will briefly describe Jacobi-Davidson for the generalized eigenproblem (2); for details we refer to [18].

As for standard eigenproblems, in each step the approximate eigenvector $\widetilde{\mathbf{q}}$ is selected from a *search subspace* span$\{\mathbf{V}\}$. This approximation with associated approximate generalized eigenvalue $\langle \widetilde{\alpha}, \widetilde{\beta} \rangle$ is "tested" with respect to some subspace span$\{\mathbf{W}\}$:

$$\widetilde{\beta}\, \mathbf{A}\, \widetilde{\mathbf{q}} - \widetilde{\alpha}\, \mathbf{B}\, \widetilde{\mathbf{q}} \perp \text{span}\{\mathbf{W}\}. \tag{29}$$

For the generalized case, it is, in view of (27) and (28), natural to take the *test subspace* span$\{\mathbf{W}\}$ different from the search subspace: the Petrov-Galerkin approach. Search subspace and test subspace are of the same dimension, say j. Equation (29) leads to the *projected eigenproblem*

$$(\widetilde{\beta}\, \mathbf{W}^* \mathbf{A} \mathbf{V} - \widetilde{\alpha}\, \mathbf{W}^* \mathbf{B} \mathbf{V})\, u = 0, \tag{30}$$

that can be solved by conventional techniques, and a solution $(u, \langle \widetilde{\alpha}, \widetilde{\beta} \rangle)$ is selected. Observe that (30) is a j-dimensional problem, and u is a j-vector. The *Petrov vector* $\widetilde{\mathbf{q}} \equiv \mathbf{V} u$ and the residual $\mathbf{r} \equiv \widetilde{\beta}\, \mathbf{A}\, \widetilde{\mathbf{q}} - \widetilde{\alpha}\, \mathbf{B}\, \widetilde{\mathbf{q}}$, associated with the *Petrov value* $\langle \widetilde{\alpha}, \widetilde{\beta} \rangle$, are computed. The subspaces span$\{\mathbf{V}\}$ and span$\{\mathbf{W}\}$ are expanded in each step of the iterative process. For certain vectors $\mathbf{p}$, $\widetilde{\mathbf{z}}$, and $\mathbf{y}$, the search subspace is expanded by a vector $\mathbf{v}$ that is orthogonal to $\mathbf{p}$ and that solves approximately the *correction equation*

$$\left(\mathbf{I} - \frac{\widetilde{\mathbf{z}}\, \mathbf{y}^*}{\mathbf{y}^* \widetilde{\mathbf{z}}}\right) (\widetilde{\beta}\, \mathbf{A} - \widetilde{\alpha}\, \mathbf{B}) \left(\mathbf{I} - \frac{\widetilde{\mathbf{q}}\, \mathbf{p}^*}{\mathbf{p}^* \widetilde{\mathbf{q}}}\right) \mathbf{v} = -\mathbf{r}. \tag{31}$$

In the next step, span$\{\mathbf{V}, \mathbf{v}\}$ is the new search subspace. Under special but obvious conditions that prevent $\mathbf{y}^* \widetilde{\mathbf{z}}$, and similar other expressions, from converging towards 0, the choice $\mathbf{y}$ in span$\{\mathbf{W}\}$ and $\widetilde{\mathbf{z}}$ in span$\{\mathbf{A}\widetilde{\mathbf{q}}, \mathbf{B}\widetilde{\mathbf{q}}\}$ leads to quadratic convergence, see [18, Th. 3.2]. In the general context in [18], there are no other restrictions on $\mathbf{p}$, $\widetilde{\mathbf{z}}$, and $\mathbf{y}$.

In our present approach we want orthogonal matrices $\mathbf{V}$ and $\mathbf{W}$, similar to $\mathbf{Z}$ and $\mathbf{Q}$ in (27), and we favor orthogonal projections. Therefore, we construct $\mathbf{V}$ and $\mathbf{W}$ to be orthogonal: new columns of $\mathbf{V}$ and $\mathbf{W}$ are orthogonalized by modified Gram-Schmidt. With $\mathbf{p} = \widetilde{\mathbf{q}}$ the right projection in (31) is orthogonal. With $\mathbf{y} = \widetilde{\mathbf{z}}$ the left projection is orthogonal also, and, as we will see, this choice is in line with the natural choice for the test subspace, without violating the restrictions for quadratic convergence.

We use the QZ algorithm [11] to reduce (30) to a generalized Schur form. With j the dimension of span$\{\mathbf{V}\}$, this algorithm yields orthogonal $(j \times j)$-matrices U_R and U_L, and upper triangular $(j \times j)$-matrices S_A and S_B, such that

$$U_L{}^*(\mathbf{W}^* \mathbf{A} \mathbf{V}) U_R = S_A \quad \text{and} \quad U_L{}^*(\mathbf{W}^* \mathbf{B} \mathbf{V}) U_R = S_B. \tag{32}$$

As for the standard case, for this generalized situation, there is also an algorithm that reorders this decomposition such that the first column of U_R and the $(1,1)$-entries of S_A and S_B represent the wanted Petrov solution of (30). We will discuss this algorithm in §3.2.

With the decomposition in (32), we construct an approximate partial generalized Schur form (cf. (27)): $\mathbf{V}U_R$ approximates a $\mathbf{Q}_k$, and $\mathbf{W}U_L$ approximates the associated $\mathbf{Z}_k$. The Jacobi-Davidson method generates a $\mathbf{V}$ for which $\mathbf{V}U_R$ approximates a $\mathbf{Q}_k$ (in general the first column will be in leading position). Since $\text{span}\{\mathbf{Z}_k\} = \text{span}\{\mathbf{A}\mathbf{Q}_k\} = \text{span}\{\mathbf{B}\mathbf{Q}_k\}$ (cf. (27)), it makes sense to choose $\mathbf{W}$ such that, for some scalars ν_0, μ_0 with, say, $|\nu_0|^2 + |\mu_0|^2 = 1$, the space $\text{span}\{\mathbf{W}\}$ coincides with $\text{span}\{\nu_0\mathbf{A}\mathbf{V} + \mu_0\mathbf{B}\mathbf{V}\}$. This choice is also in line with the restriction $\mathbf{w} = \widetilde{\mathbf{z}}$ and the other restrictions on $\mathbf{w}$ and $\widetilde{\mathbf{z}}$, mentioned above.

In summary, for suitable fixed scalars ν_0, μ_0, we propose the following Jacobi-Davidson method:

• reduce the projected problem (30) to an ordered generalized Schur decomposition (32) and select as approximate generalized eigenpair:

$$(\widetilde{\mathbf{q}}, \langle\widetilde{\alpha}, \widetilde{\beta}\rangle) \equiv (\mathbf{V}U_R(:,1), \langle S_A(1,1), S_B(1,1)\rangle);$$

• for

$$\gamma\,\widetilde{\mathbf{z}} \equiv \nu_0\mathbf{A}\widetilde{\mathbf{q}} + \mu_0\mathbf{B}\widetilde{\mathbf{q}}, \quad \text{and} \quad \mathbf{r} \equiv \widetilde{\beta}\,\mathbf{A}\widetilde{\mathbf{q}} - \widetilde{\alpha}\,\mathbf{B}\widetilde{\mathbf{q}}, \tag{33}$$

where γ is a normalization constant, compute an approximate solution $\mathbf{v} \perp \widetilde{\mathbf{q}}$ of the resulting correction equation

$$\widetilde{\mathbf{q}}^*\mathbf{v} = 0 \quad \text{and} \quad (\mathbf{I} - \widetilde{\mathbf{z}}\,\widetilde{\mathbf{z}}^*)(\widetilde{\beta}\,\mathbf{A} - \widetilde{\alpha}\,\mathbf{B})(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)\,\mathbf{v} = -\mathbf{r}; \tag{34}$$

• expand $\mathbf{V}$ with $\mathbf{v}$ and $\mathbf{W}$ with $\mathbf{w}$, where

$$\mathbf{w} \equiv \nu_0\,\mathbf{A}\mathbf{v} + \mu_0\,\mathbf{B}\mathbf{v}. \tag{35}$$

It can be shown that, with the above choices for $\widetilde{\mathbf{z}}$ and $\mathbf{W}$,

$$\widetilde{\mathbf{z}} = \mathbf{W}U_L(:,1). \tag{36}$$

In this approach, the relation between the partial generalized Schur form for the large problem and the complete generalized Schur form for the small problem (30) via right vectors ($\widetilde{\mathbf{q}} = \mathbf{V}U_R(:,1)$) is similar to the relation via left vectors ($\widetilde{\mathbf{z}} = \mathbf{W}U_L(:,1)$). The fact that $\widetilde{\mathbf{z}} = \mathbf{W}U_L(:,1)$, is also convenient for restart purposes, as we will see in §3.2.

Finally, we will explain how to select the scalars ν_0 and μ_0. The restriction $|\nu_0|^2 + |\mu_0|^2 = 1$ is for scaling and avoids trivial expansions. We pursue three approaches. The first two, in §§3.1.1-3.1.2, are closely related, and can be viewed as a generalizations of the approach by Ritz values for standard eigenproblems, for optimal expansion of the test subspace. The third one, in §3.5.1, is more closely related to the approach by harmonic Ritz values, and aims for optimal selection of Petrov pairs.

3.1.1 Fixed values for ν_0 and μ_0. If $\mathbf{v}$ is the expansion vector for the search subspace then, in the general setting, we have to expand the test subspace by $\nu_0\mathbf{A}\mathbf{v} + \mu_0\mathbf{B}\mathbf{v}$. Note that, if $\widetilde{\mathbf{q}}$ is the new approximate eigenvector then expanding the old search subspace by $\mathbf{v}$ is equivalent to expanding it by $\widetilde{\mathbf{q}}$, so that the new test subspace can also be obtained by expanding with $\nu_0\mathbf{A}\widetilde{\mathbf{q}} + \mu_0\mathbf{B}\widetilde{\mathbf{q}}$. For $\mathbf{B} = \mathbf{I}$, the obvious choice would be, $\nu_0 = 0$ and $\mu_0 = 1$. However, if $\mathbf{A} = \mathbf{I}$, the obvious choice would be $\nu_0 = 1$ and $\mu_0 = 0$. In this case, although $\mathbf{B}\widetilde{\mathbf{q}}$ is in the direction of $\widetilde{\mathbf{q}}$, if $\widetilde{\mathbf{q}}$ is close to some eigenvector $\mathbf{q}$, multiplication by $\mathbf{B}$ may diminish the most important eigenvector components of $\widetilde{\mathbf{q}}$, if the eigenvalue of $\mathbf{B}$ associated to $\mathbf{q}$ is (very) small. Therefore, expanding the test space by $\mathbf{B}\widetilde{\mathbf{q}}$ may be (much) less optimal than expanding by $\widetilde{\mathbf{q}}$. In the presence of rounding errors, this effect may be even more prominent.

For the standard case, where either $\mathbf{A} = \mathbf{I}$ or $\mathbf{B} = \mathbf{I}$, $\mathbf{A}\mathbf{q} = \alpha\mathbf{q}$ and $\mathbf{B}\mathbf{q} = \beta\mathbf{q}$, for $\langle\alpha, \beta\rangle$ with either $\alpha = 1$ or $\beta = 1$, the optimal choice with respect to the $\mathbf{q}$-components seems to be

$$\nu_0 = \frac{\overline{\alpha}}{\sqrt{|\alpha|^2 + |\beta|^2}} \quad \text{and} \quad \mu_0 = \frac{\overline{\beta}}{\sqrt{|\alpha|^2 + |\beta|^2}}, \tag{37}$$

since this choice maximizes

$$\|\nu_0\mathbf{A}\mathbf{q} + \mu_0\mathbf{B}\mathbf{q}\|_2 \quad (= |\nu_0\alpha + \mu_0\beta|\,\|\mathbf{q}\|_2).$$

Also, for the generalized problem, where $\mathbf{A}\mathbf{q} = \alpha\mathbf{z}$ and $\mathbf{B}\mathbf{q} = \beta\mathbf{z}$, if $(\mathbf{q}, \langle\alpha, \beta\rangle)$ is the desired eigenpair, we can select ν_0 and μ_0 as in (37). With respect to $\mathbf{q}$, this choice maximizes

$$\|\nu_0\mathbf{A}\mathbf{q} + \mu_0\mathbf{B}\mathbf{q}\|_2 \quad (= |\nu_0\alpha + \mu_0\beta|\,\|\mathbf{z}\|_2).$$

This approach can be viewed as an attempt to expand the test subspace span$\{\mathbf{W}\}$ optimally in the direction of $\mathbf{z}$, where $\mathbf{z}$ is the normalized vector $\mathbf{A}\mathbf{q}$ (or $\mathbf{B}\mathbf{q}$).

Since we have to choose ν_0 and μ_0 before we even know the generalized eigenvalue $\langle\alpha, \beta\rangle$, the best we can do, certainly in the initial phase of the process, is to select

$$\nu_0 \equiv \frac{\overline{\tau}}{\sqrt{1 + |\tau|^2}} \quad \text{and} \quad \mu_0 \equiv \frac{1}{\sqrt{1 + |\tau|^2}}, \tag{38}$$

where τ is the target value.

3.1.2 Adaptive values for ν_0 and μ_0. For a well-balanced expansion of the test subspace, the expressions in (37) suggest that we take advantage of the current approximations $\langle\widetilde{\alpha}, \widetilde{\beta}\rangle$ for the generalized eigenpair. Intuitively

$$\nu_0 \equiv \frac{\overline{\widetilde{\alpha}}}{\sqrt{|\widetilde{\alpha}|^2 + |\widetilde{\beta}|^2}} \quad \text{and} \quad \mu_0 \equiv \frac{\overline{\widetilde{\beta}}}{\sqrt{|\widetilde{\alpha}|^2 + |\widetilde{\beta}|^2}} \tag{39}$$

might be a good choice. Ideally, at convergence, the space span$\{\mathbf{W}\}$ should coincide with span$\{\overline{\alpha}\mathbf{A}\mathbf{V} + \overline{\beta}\mathbf{A}\mathbf{V}\}$. Since $\mathbf{V}$ is expanded by $\mathbf{v}$, this suggests to expand $\mathbf{W}$ by

$$\mathbf{w} \equiv \nu_0\mathbf{A}\mathbf{v} + \mu_0\mathbf{B}\mathbf{v}. \tag{40}$$

Unfortunately, since all intermediate ν_0 and μ_0 are different, this does not guarantee that at convergence $\overline{\alpha}\mathbf{A}\mathbf{q} + \overline{\beta}\mathbf{B}\mathbf{q} \in \text{span}\{\mathbf{W}\}$. Nevertheless this *adaptive* variant (39)-(40) turns out to work better in practice than the fixed variant in §3.1.1. For the $\widetilde{\mathbf{z}}$ in the correction equation (34) we took $\mathbf{W}U_L(:,1)$ (see (36)). This vector is in the direction of the orthogonal projection of $\nu_0\mathbf{A}\widetilde{\mathbf{q}} + \mu_0\mathbf{B}\widetilde{\mathbf{q}}$ onto span$\{\mathbf{W}\}$ and, in practice, the angle between the vectors $\mathbf{W}U_L(:,1)$ and $\nu_0\mathbf{A}\widetilde{\mathbf{q}} + \mu_0\mathbf{B}\widetilde{\mathbf{q}}$ appears to converge rapidly towards zero.

Remark 12 If, for standard eigenproblems, test subspaces coincide with the corresponding search subspaces, then we refer to the approximate eigenpairs as Ritz pairs: the approximate eigenpairs are associated with the Ritz-Galerkin approximation. If the subspaces do not coincide, we prefer the name Petrov pairs, because of its relation with Petrov-Galerkin approximation. Petrov values associated with the standard eigenproblem for $\mathbf{A}$ and the choice $\mathbf{W} = \mathbf{A}\mathbf{V}$ are called harmonic Ritz values. The reference to "Ritz" in their name expresses the fact that these Petrov values are the reciprocals of the Ritz values associated with the standard eigenproblem for $\mathbf{A}^{-1}$ with test and search subspace both equal to $\mathbf{A}\mathbf{V}$.

For generalized eigenproblems, there is no canonical relation between search and test subspace (except for unitary $\mathbf{B}$) and we believe that the name Petrov pairs is appropriate here. However, in our approach, orthogonality and the QZ-algorithm are central. As we have explained above, this leads quite naturally to some specific choice of the test subspaces. Therefore, we call the Petrov pairs associated with the choice made in §3.1.1 the *standard Petrov* pairs.

3.2 Practical selection and implicit restart. When we reduce the projected eigenproblem (30) to a generalized Schur form by the QZ algorithm [11], then we can exploit the generalized Schur form for various purposes: — selection of a Petrov pair $(\widetilde{\mathbf{q}}, \langle\widetilde{\alpha}, \widetilde{\beta}\rangle)$, — selection of the corresponding left vector $\widetilde{\mathbf{z}}$, — restriction of the dimension of the subspaces span$\{\mathbf{V}\}$ and span$\{\mathbf{W}\}$, if necessary. We will explain the first and last point in more detail.

Suppose that the generalized Schur form of the interaction pair

$$(M_A, M_B) \equiv (\mathbf{W}^*\mathbf{A}\mathbf{V}, \mathbf{W}^*\mathbf{B}\mathbf{V}),$$

given by

$$U_L^* M_A U_R = S_A \quad \text{and} \quad U_L^* M_B U_R = S_B,$$

is ordered with respect to τ such that

$$|S_A(1,1)/S_B(1,1) - \tau| \leq |S_A(2,2)/S_B(2,2) - \tau| \leq \cdots \leq |S_A(j,j)/S_B(j,j) - \tau|, \tag{41}$$

where j is the dimension of span$\{\mathbf{V}\}$. Then

$$(\widetilde{\mathbf{q}}, \langle\widetilde{\alpha}, \widetilde{\beta}\rangle) \equiv (\mathbf{V}U_R(:,1), \langle S_A(1,1), S_B(1,1)\rangle))$$

is the Petrov approximation corresponding to the projected system (30) with Petrov value closest to the target τ. The corresponding left vector is given by $\widetilde{\mathbf{z}} \equiv \mathbf{W}U_L(:,1)$. Furthermore, $\mathbf{V}U_R(:,1\!:\!i)$, with $i < j$, spans the subspace that contains the i most promising Petrov vectors. The corresponding test subspace is given by $\mathbf{W}U_L(:,1\!:\!i)$. Therefore, similar to the approach in §2.2 (cf. [20, §5.3]), when we want to reduce the dimension of the subspace ("implicit restart") to $j_{\min}$, $j_{\min} < j$, then we simply discard the columns $\mathbf{v}_{j_{\min}+1}$ through $\mathbf{v}_j$, and $\mathbf{w}_{j_{\min}+1}$ through $\mathbf{w}_j$, and continue the Jacobi-Davidson algorithm with

$$\mathbf{V} = \mathbf{V}U_R(:,1\!:\!j_{\min}) \quad \text{and} \quad \mathbf{W} = \mathbf{W}U_L(:,1\!:\!j_{\min}).$$

For a reordering algorithm for the generalized Schur form, see, for instance, [27], [28]. A simple MATLAB implementation for reordering with respect to a target value τ is given in Appendix C. A theoretical explanation is given there as well.

3.3 JDQZ. In this section we focus on the efficient computation of a set of generalized eigenpairs. The idea is to use the Jacobi-Davidson method for generalized eigenproblems (§3.1) for the computation of a partial generalized Schur form.

Suppose that we have the partial generalized Schur form $\mathbf{A}\mathbf{Q}_{k-1} = \mathbf{Z}_{k-1}S_{k-1}$ and $\mathbf{B}\mathbf{Q}_{k-1} = \mathbf{Z}_{k-1}T_k$. Analogously to (8), we want to expand this partial generalized Schur form with a suitable $\mathbf{q}$ and $\mathbf{z}$, to

$$\mathbf{A}\begin{bmatrix}\mathbf{Q}_{k-1} & \mathbf{q}\end{bmatrix} = \begin{bmatrix}\mathbf{Z}_{k-1} & \mathbf{z}\end{bmatrix}\begin{bmatrix}S_{k-1} & s \\ 0 & \alpha\end{bmatrix} \quad \text{and} \quad \mathbf{B}\begin{bmatrix}\mathbf{Q}_{k-1} & \mathbf{q}\end{bmatrix} = \begin{bmatrix}\mathbf{Z}_{k-1} & \mathbf{z}\end{bmatrix}\begin{bmatrix}T_k & t \\ 0 & \beta\end{bmatrix}. \tag{42}$$

From this we deduce that the generalized Schur pair $(\mathbf{q}, \langle\alpha, \beta\rangle)$ satisfies

$$\mathbf{Q}_{k-1}^*\mathbf{q} = 0 \quad \text{and} \quad (\beta\,\mathbf{A} - \alpha\,\mathbf{B})\,\mathbf{q} - \mathbf{Z}_{k-1}u = \mathbf{0}, \tag{43}$$

for $u \equiv \mathbf{Z}_{k-1}^*(\beta\,\mathbf{A} - \alpha\,\mathbf{B})\,\mathbf{q}$. This leads to

$$\mathbf{Q}_{k-1}^*\mathbf{q} = 0 \quad \text{and} \quad (\mathbf{I} - \mathbf{Z}_{k-1}\,\mathbf{Z}_{k-1}^*)(\beta\,\mathbf{A} - \alpha\,\mathbf{B})\,\mathbf{q} = \mathbf{0}.$$

Hence, $(\mathbf{q}, \langle\alpha, \beta\rangle)$ satisfies

$$\mathbf{Q}_{k-1}^*\mathbf{q} = 0 \quad \text{and} \quad (\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)(\beta\,\mathbf{A} - \alpha\,\mathbf{B})(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,\mathbf{q} = \mathbf{0}. \tag{44}$$

and the generalized Schur pair $(\mathbf{q}, \langle\alpha, \beta\rangle)$ is therefore also an eigenpair of the *deflated matrix pair*

$$(\,(\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)\mathbf{A}(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*), (\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)\mathbf{B}(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,). \tag{45}$$

In JDQZ we solve this eigenproblem with the Jacobi-Davidson method for the generalized eigenproblem.

In more detail, the procedure is as follows. Let $\mathbf{V}$ and $\mathbf{W}$ be orthogonal $(n \times j)$-matrices such that $\mathbf{V}^*\mathbf{Q}_{k-1} = \mathbf{W}^*\mathbf{Z}_{k-1} = 0$. Let

$$M_A \equiv \mathbf{W}^*(\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)\mathbf{A}(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,\mathbf{V} = \mathbf{W}^*\mathbf{A}\mathbf{V} \quad \text{and}$$
$$M_B \equiv \mathbf{W}^*(\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)\mathbf{B}(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,\mathbf{V} = \mathbf{W}^*\mathbf{B}\mathbf{V},$$

and denote the generalized Schur form of the matrix pair (M_A, M_B) by

$$U_L^* M_A U_R = S_A \quad \text{and} \quad U_L^* M_B U_R = S_B. \tag{46}$$

If this generalized Schur form is ordered with respect to the target value τ, then

$$(\widetilde{\mathbf{q}}, \langle\widetilde{\alpha}, \widetilde{\beta}\rangle) \equiv (\mathbf{V}U_R(:,1), \langle S_A(1,1), S_B(1,1)\rangle)$$

is a Petrov pair approximation for a solution of (44). The corresponding left vector is given by

$$\widetilde{\mathbf{z}} \equiv \mathbf{W}U_L(:,1).$$

The Jacobi-Davidson method expands $\mathbf{V}$ with the orthogonal complement of $\mathbf{v}$ that is an (approximate) solution of the *generalized deflated correction equation*

$$\begin{cases} \mathbf{Q}_{k-1}^*\mathbf{v} = 0, \quad \widetilde{\mathbf{q}}^*\mathbf{v} = 0, \quad \text{and} \\ (\mathbf{I} - \widetilde{\mathbf{z}}\,\widetilde{\mathbf{z}}^*)(\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)(\widetilde{\beta}\,\mathbf{A} - \widetilde{\alpha}\,\mathbf{B})(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)\,\mathbf{v} = -\mathbf{r}, \end{cases} \tag{47}$$

where $\mathbf{r} \equiv (\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)(\widetilde{\beta}\,\mathbf{A} - \widetilde{\alpha}\,\mathbf{A})(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,\widetilde{\mathbf{q}}$. We also have to expand $\mathbf{W}$; we expand with the orthogonal complement of $(\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)(\nu_0\,\mathbf{A} + \mu_0\,\mathbf{B})(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,\widetilde{\mathbf{q}}$ to $\mathbf{W}$.

When the generalized Schur pair $(\widetilde{\mathbf{q}}, \langle\widetilde{\alpha}, \widetilde{\beta}\rangle)$ is sufficiently close to $(\mathbf{q}, \langle\alpha, \beta\rangle)$, then we may continue for still another generalized Schur pair. In that case $\mathbf{V}$ and $\mathbf{W}$ are replaced by $\mathbf{V}U_R(:,2\!:\!j)$ and $\mathbf{W}U_L(:,2\!:\!j)$, in order to obtain a new search subspace orthogonal to span$\{\mathbf{Q}_{k-1}, \widetilde{\mathbf{q}}\}$, and a new test subspace orthogonal to span$\{\mathbf{Z}_{k-1}, \widetilde{\mathbf{z}}\}$, respectively, and we continue the process.

3.4 Preconditioning. In this section we discuss briefly preconditioning for the generalized deflated correction equation.

The correction equation (44) involves an operator for which the domain and the image space differ. This means that Krylov subspace methods can not be applied right away. Fortunately, this can be fixed easily by incorporating preconditioning.

Similarly to §2.4, we propose to use

$$(\mathbf{I} - \widetilde{\mathbf{z}}\,\widetilde{\mathbf{z}}^*)(\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)\mathbf{K}(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*). \tag{48}$$

for some preconditioner $\mathbf{K} \approx \mathbf{A} - \tau\,\mathbf{B}$.

We modify our notation slightly (cf. NOTATION 1):

NOTATION 2

$$\begin{aligned}
\widetilde{\mathbf{Q}}_k &\equiv [\,\mathbf{Q}_{k-1}, \widetilde{\mathbf{q}}\,], && \text{the matrix } \mathbf{Q}_{k-1} \text{ expanded by } \widetilde{\mathbf{q}},\\
\widetilde{\mathbf{Z}}_k &\equiv [\,\mathbf{Z}_{k-1}, \widetilde{\mathbf{z}}\,], && \text{the matrix } \mathbf{Z}_{k-1} \text{ expanded by } \widetilde{\mathbf{z}},\\
\widetilde{\mathbf{Y}}_k &\equiv \mathbf{K}^{-1}\widetilde{\mathbf{Z}}_k, && \text{the expanded matrix of preconditioned vectors},\\
\widetilde{H}_k &\equiv \widetilde{\mathbf{Q}}_k^*\widetilde{\mathbf{Y}}_k, && \text{the projected preconditioner } \widetilde{\mathbf{Q}}_k^*\mathbf{K}^{-1}\widetilde{\mathbf{Z}}_k.
\end{aligned}$$

In this notation the left preconditioned correction equation for the generalized correction equation can be written as

$$\widetilde{\mathbf{Q}}_k^*\mathbf{v} = 0 \quad\text{and}\quad (\mathbf{I} - \widetilde{\mathbf{Y}}_k\,\widetilde{H}_k^{-1}\widetilde{\mathbf{Q}}_k^*)\,\mathbf{K}^{-1}(\widetilde{\beta}\,\mathbf{A} - \widetilde{\alpha}\,\mathbf{B})(\mathbf{I} - \widetilde{\mathbf{Y}}_k\,\widetilde{H}_k^{-1}\widetilde{\mathbf{Q}}_k^*)\,\mathbf{v} = -\widehat{\mathbf{r}}, \tag{49}$$

where $\widehat{\mathbf{r}} \equiv (\mathbf{I} - \widetilde{\mathbf{Y}}_k\,\widetilde{H}_k^{-1}\widetilde{\mathbf{Q}}_k^*)\,\mathbf{K}^{-1}\mathbf{r}$. Observe that, for $\mathbf{K} = \mathbf{I}$, this equation is equivalent with the one in (47).

Of course, right preconditioned generalized correction equations can be derived in a similar manner. With $\widehat{\mathbf{Y}}_k \equiv (\mathbf{K}^*)^{-1}\widetilde{\mathbf{Q}}_k$,

$$\widehat{\mathbf{Y}}_k^*\widehat{\mathbf{v}} = 0 \quad\text{and}\quad (\mathbf{I} - \widetilde{\mathbf{Z}}_k\widetilde{H}_k^{-1}\widehat{\mathbf{Y}}_k^*)(\widetilde{\beta}\,\mathbf{A} - \widetilde{\alpha}\,\mathbf{B})\,\mathbf{K}^{-1}(\mathbf{I} - \widetilde{\mathbf{Z}}_k\widetilde{H}_k^{-1}\widehat{\mathbf{Y}}_k^*)\,\widehat{\mathbf{v}} = -\widehat{\mathbf{r}},$$

where $\widehat{\mathbf{r}} \equiv (\mathbf{I} - \widetilde{\mathbf{Z}}_k\widetilde{H}_k^{-1}\widehat{\mathbf{Y}}_k^*)\,\mathbf{r}$. Then $\mathbf{v} = \mathbf{K}^{-1}\widehat{\mathbf{v}}$.

Note that for the operators in the preconditioned correction equation, the domain and the image space coincide, so that Krylov subspace methods can be used.

A pseudo-code for the preconditioned Jacobi-Davidson QZ-algorithm with harmonic Petrov values (to be discussed in §3.5.1) is given in ALG. 4.

In TAB. 3 we have listed the main computational ingredients per iteration of JDQZ.

Part	DOTS	AXPYS	MVs	K
The correction equation	variable			
The Projected problem	$6j+k$	$2j-1$	2 [a]	0
The Petrov approximation	$2k+1$	$2j+3k$	2 [b]	1
Found	0	$4j-4$	0	0
Restart	0	$4j_{\min}$	0	0

[a] If Krylov subspace methods are used to solve the correction equation, then the products $\mathbf{Av}$ and $\mathbf{Bv}$ are often already available, as side-products. No MVs are needed in this part then.
[b] Instead of computing the residual $\mathbf{r}$ as $(\widetilde{\beta}\,\mathbf{A} - \widetilde{\alpha}\,\mathbf{B})\,\widetilde{\mathbf{q}}$, $\mathbf{r}$ may also be computed as $\widetilde{\beta}\,\mathbf{V}_A U(:,1) - \widetilde{\alpha}\,\mathbf{V}_B U(:,1)$, where $\mathbf{V}_A \equiv \mathbf{AV}$ and $\mathbf{V}_B \equiv \mathbf{BV}$ (cf. ALG. 4); depending on the number of nonzeros in $\mathbf{A}$, $\mathbf{B}$ and the value of j, this may be more efficient.

TABLE 3. *The computational costs of JDQZ per iteration. The integers j and k are the dimensions of* $\operatorname{span}\{\mathbf{V}\}$ *and* $\operatorname{span}\{\mathbf{Q}\}$*, respectively.*

3.5 The selection of Petrov pairs. In the approaches with fixed and adaptive parameters for the generalized case in § 3.1.1 and §3.1.2, our goal is optimal expansion of the test subspace, implicitly leading to optimal expansion of the search subspace as well. But the speed of convergence of a Jacobi-Davidson algorithm also depends on the ability to select the "correct" Petrov pairs (cf. §2.5).

Of course, here the selection can be improved too by a tracking strategy, as explained in §2.5.1. We need some confidence that we are tracking the correct Petrov value. In §2.5.1, we

function $[\mathbf{Q}, \mathbf{Z}, R_A, R_B] = \mathrm{JDQZ}\,(\mathbf{A}, \mathbf{B}, \mathbf{K}, \tau, \mathbf{v}_0, \epsilon, k_{\max}, j_{\min}, j_{\max})$
$\mathbf{Q} = [\,];\ \mathbf{Z} = [\,];\ R_A = [\,];\ R_B = [\,];\ \mathbf{Y} = [\,];\ H = [\,];$
$\mathbf{V} = [\,];\ \mathbf{V}_A = [\,];\ \mathbf{V}_B = [\,];\ \mathbf{W} = [\,];\ M_A = [\,];\ M_B = [\,];$
$\gamma = \sqrt{1 + |\tau|^2};\ \alpha_0 = \tau/\gamma,\ \beta_0 = 1/\gamma;\ \ k = 0;\ j = 0;$
while $k < k_{\max}$,
  if $j == 0$,
    $\mathbf{v} = \mathbf{v}_0;$
  else
    — *the correction equation* —
    $\mathbf{r} = (\mathbf{I} - \widetilde{\mathbf{Y}} \widetilde{H}^{-1} \widetilde{\mathbf{Q}}^*)\, \mathbf{K}^{-1} \mathbf{r};$
    Solve $\mathbf{v}$ *(approximately) from:*
      $\widetilde{\mathbf{Q}}^* \mathbf{v} = 0$ *and*
      $(\mathbf{I} - \widetilde{\mathbf{Y}} \widetilde{H}^{-1} \widetilde{\mathbf{Q}}^*)\, \mathbf{K}^{-1} (\beta\, \mathbf{A} - \alpha\, \mathbf{B})(\mathbf{I} - \widetilde{\mathbf{Y}} \widetilde{H}^{-1} \widetilde{\mathbf{Q}}^*)\, \mathbf{v} = -\mathbf{r}.$
  end
  — *the projected problem* —
  $\mathbf{v} = \mathsf{mgs}\,(\mathbf{V}, \mathbf{v});\ \mathbf{v} = \mathbf{v}/\|\mathbf{v}\|_2;\ \mathbf{v}_A = \mathbf{A}\mathbf{v};\ \mathbf{v}_B = \mathbf{B}\mathbf{v};$
  $\mathbf{w} = (\beta_0 \mathbf{v}_A - \alpha_0 \mathbf{v}_B);\ \mathbf{w} = \mathsf{mgs}\,(\mathbf{Z}, \mathbf{w});\ \mathbf{w} = \mathsf{mgs}\,(\mathbf{W}, \mathbf{w});\ \mathbf{w} = \mathbf{w}/\|\mathbf{w}\|_2;$
  $M_A = [\, M_A, \mathbf{W}^* \mathbf{v}_A; \mathbf{w}^* \mathbf{V}_A, \mathbf{w}^* \mathbf{v}_A\,];\ M_B = [\, M_B, \mathbf{W}^* \mathbf{v}_B; \mathbf{w}^* \mathbf{V}_B, \mathbf{w}^* \mathbf{v}_B\,];$
  $\mathbf{V} = [\,\mathbf{V}, \mathbf{v}\,];\ \mathbf{V}_A = [\,\mathbf{V}_A, \mathbf{v}_A\,];\ \mathbf{V}_B = [\,\mathbf{V}_B, \mathbf{v}_B\,];\ \mathbf{W} = [\,\mathbf{W}, \mathbf{w}\,];$
  $[U_L, U_R, S_A, S_B] = \mathsf{qz}\,(M_A, M_B);$
  $[U_L, U_R, S_A, S_B] = \mathsf{qzsort}\,(\tau, U_L, U_R, S_A, S_B);$
  $j = j + 1;\ \mathrm{found} = 1;$
  while found,
    — *harmonic Petrov approximation* —
    $\alpha = S_A(1,1);\ \beta = S_B(1,1);\ \mathbf{q} = \mathbf{V} U_R(:,1);\ \mathbf{z} = \mathbf{W} U_L(:,1);\ \mathbf{y} = \mathbf{K}^{-1}\mathbf{z};$
    $\mathbf{r} = \mathbf{A}\,\mathbf{q};\ [\,\mathbf{r}, s_A] = \mathsf{mgs}\,(\mathbf{Z}, \mathbf{r});\ \mathbf{r}_B = \mathbf{B}\,\mathbf{q};\ [\,\mathbf{r}_B, s_B] = \mathsf{mgs}\,(\mathbf{Z}, \mathbf{r}_B);$
    $\mathbf{r} = \beta\,\mathbf{r} - \alpha\,\mathbf{r}_B;$
    $\widetilde{\mathbf{Q}} = [\,\mathbf{Q}, \mathbf{q}\,];\ \widetilde{\mathbf{Y}} = [\,\mathbf{Y}, \mathbf{y}\,];\ \widetilde{H} = [\,H, \mathbf{Q}^*\mathbf{y}; \mathbf{q}^*\mathbf{Y}, \mathbf{q}^*\mathbf{y}\,];$
      ⋮
    "found and implicit restart part", see ALG. 5
      ⋮
  end
end

JDQZ returns a partial generalized Schur form $(\mathbf{Q}, \mathbf{Z}, R_A, R_B)$ of dimension $k_{\max}$ of the matrix pair $(\mathbf{A}, \mathbf{B})$ with generalized eigenvalues near the target τ. $\mathbf{K}$ is a preconditioner for $\mathbf{A} - \tau\mathbf{B}$, $\mathbf{v}_0$ is an initial guess, and ϵ is the stopping tolerance. $j_{\max}$ and $j_{\min}$ specify the dimension of the search subspace before and after implicit restart, respectively. qz is a MATLAB function that computes a generalized Schur decomposition. The functions mgs (modified Gram-Schmidt) and qzsort (Sort generalized Schur form) can be found in Appendix C.

ALGORITHM 4. *Preconditioned JDQZ, using harmonic Petrov values.*

proposed to rely on the norm of the previous residual. Unfortunately, this strategy will not be helpful in the initial stage of the process, where there are no small residuals.

In §3.5.1 we will introduce harmonic Petrov values. We will see that the harmonic Petrov values, that are closest to a target, can be obtained as extremal Ritz values for a specific test subspace, also if the target is in the interior of the spectrum. Specifically for such a situation, the harmonic Petrov values appear to be attractive competitors for the standard Petrov values of

```
found = (||r||_2 < ε) & (j > 1 | k = k_max − 1);
if found,
    — found —
    Q = Q̃;  Z = Z̃;
    R_A = [ R_A, s_A; zeros (1, k), α];  R_B = [ R_B, s_B; zeros (1, k), β];
    k = k + 1;  if k = k_max,  break;  end
    Y = Ỹ;  H = H̃;
    J = [2:j];  j = j − 1;
    V = V U_R(:, J);  V_A = V_A U_R(:, J);  V_B = V_B U_R(:, J);
    W = W U_L(:, J);  S_A = S_A(J, J);  S_B = S_B(J, J);
    M_A = S_A;  M_B = S_B;  U_R = I;  U_L = I;
elseif j == j_max,
    — implicit restart —
    j = j_min;  J = [1:j];
    V = V U_R(:, J);  V_A = V_A U_R(:, J);  V_B = V_B U_R(:, J);
    W = W U_L(:, J);  S_A = S_A(J, J);  S_B = S_B(J, J);
    M_A = S_A;  M_B = S_B;  U_R = I;  U_L = I;
end
```

ALGORITHM 5. *"Found and implicit restart part" of preconditioned JDQZ, with harmonic Petrov values.*

the approaches in § 3.1.1 and §3.1.2: for generalized eigenproblems the costs for the computation of the standard Petrov values is the same as for harmonic Petrov values, and, because of the extremality property, harmonic Petrov values closest to the target appear to be the best choices, also in early stages of the process (cf. §2.5). A tracking strategy is not required nor helpful in this case.

3.5.1 A harmonic Petrov value approach. We first consider the computation of the eigenvalues of a standard eigenproblem ($\mathbf{B} = \mathbf{I}$), that are close to some target value τ in the interior of (the convex hull of) the spectrum. The transformation $\lambda \rightsquigarrow 1/(\lambda - \tau)$ maps these eigenvalues λ to extremal eigenvalues of $(\mathbf{A} - \tau\,\mathbf{I})^{-1}$ and in that case the "correct" eigenpair approximations can be obtained easily (cf. §2.5). But, we want to avoid matrix inversion. With some formula manipulation, it can be shown that this can be achieved by taking the search subspace and the test subspace both equal to span$\{(\mathbf{A} - \tau\,\mathbf{I})\mathbf{V}\}$ (cf. [20, §5.1]): the resulting eigenvalue approximations $\widetilde{\lambda}$ for $\mathbf{A}$ are then the solutions of

$$\mathbf{W}^*\mathbf{A}\mathbf{V}u - \widetilde{\lambda}\,\mathbf{W}^*\mathbf{V}u = 0, \quad \text{where } \mathbf{W} \equiv (\mathbf{A} - \tau\,\mathbf{I})\mathbf{V}. \tag{50}$$

The solutions $\widetilde{\lambda}$ are called *harmonic Ritz values* of $\mathbf{A}$, with respect to τ (cf. [13], [20], and also [12]); $\mathbf{V}u$ is the associated *harmonic Ritz vector.* Since $\mathbf{W}$ and $\gamma\mathbf{W}$, with $\gamma \equiv 1/\sqrt{1+|\tau|^2}$, span the same space, the harmonic Ritz values appear as Petrov values for the test subspace generated as in (35), with

$$\nu_0 \equiv \frac{1}{\sqrt{1+|\tau|^2}} \quad \text{and} \quad \mu_0 \equiv -\frac{\tau}{\sqrt{1+|\tau|^2}}. \tag{51}$$

For generalized problems, with ν_0 and μ_0 as in (51), the Petrov values closest to the target value correspond to absolute largest Ritz values of the standard eigenproblem, with matrix $(\mathbf{A} - \tau\,\mathbf{B})^{-1}(\overline{\tau}\mathbf{A} + \mathbf{B})$. Therefore, for this generalized case also a better selection of appropriate eigenpair approximations may be expected. We refer to the Petrov values associated with this choice of test subspace as *harmonic Petrov* values.

Observe that it does not make sense to try to improve the mixture $\mathbf{W}$ of $\mathbf{AV}$ and $\mathbf{BV}$ by taking $\nu_0 = \widetilde{\beta}$ and $\mu_0 = -\widetilde{\alpha}$, if a better generalized eigenvalue approximation $\langle\widetilde{\alpha}, \widetilde{\beta}\rangle$ becomes available (cf. §3.1.2). On the contrary, as is explained in §3.1.1, this would spoil precisely the interesting components in the expansion vector of $\mathbf{W}$. However, recall that, with the harmonic approach in (51), our goal is not an optimal mixture, but a less risky selection procedure.

From the above arguments, it will be clear that the harmonic approach (51) will be not better than the adaptive approach (39), when we want extremal eigenvalues: no better selection may be expected, but the expansion of the test subspace will be less optimal. However, for interior eigenvalues, experiments show (cf. §4.4 and §4.8) that the harmonic approach can work much better than the adaptive approach, even if the latter one is enhanced with the tracking strategy.

4 Numerical experiments. In this section we present numerical results, obtained with JDQR and JDQZ, for several eigenproblems and generalized eigenproblems. The purpose of these experiments is to get a good impression of the actual behavior of these methods. We have not tried to find the most efficient parameter choices for each particular problem. We will illustrate the effect of a more accurately solving the correction equation, and the effect of including appropriate preconditioning. We will show that the harmonic Petrov value choice for the test subspace may lead to superior convergence behavior, not only for the generalized eigenproblem, but also for the standard eigenproblem. We will demonstrate that the projections in the correction equation (12), involving detected Schur vectors, are essential components of the algorithms. We will see that the tracking strategy (§2.5.1) can be very helpful. We will also consider eigenproblems where multiple eigenvalues are involved.

The computations were done in double complex precision ($\approx$ 15 digits) on a Sun workstation. To facilitate comparison, we have selected for all cases $j_{\max} = 15$, $j_{\min} = 10$ (the dimension of the subspace before and after implicit restart, respectively), and a fixed random real vector $\mathbf{v}_0$ as an initial guess (cf. ALG. 2 and ALG. 4).

As iterative solvers for the correction equation, we have considered full GMRES [17] with a maximum of m steps, denoted by GMRES$_m$, and BiCGstab(2) [19]. For BiCGstab(2) a maximum of 100 matrix multiplications was allowed. As stopping criterion for the iterative methods for the correction equation, we have used $\|\widetilde{\mathbf{r}}_i\|_2 < 2^{-j}\|\widetilde{\mathbf{r}}_0\|_2$, where $\widetilde{\mathbf{r}}_0$ is the initial residual, $\widetilde{\mathbf{r}}_i$ is the residual corresponding to the approximate solution produced by the inner method, and j is the iteration number for the current eigenvalue approximation in the outer iteration. Hence, as the outer iterations proceed, the inner iterations are solved more accurately. This choice was inspired by the fact that the Jacobi-Davidson method may be viewed as a Newton process [18], [21], and for Newton processes this stopping criterion may lead to efficient algorithms [5]. As the initial guess for the inner iteration method we always took the null-vector. The tracking strategy of §2.5.1 has been used in all examples, except where stated differently.

In the figures of the convergence behavior for JDQR and JDQZ, the performance is plotted in terms of the actual amount of work, in millions of floating point operations (flops), versus $\log_{10}$ of the residual norm. The reason for this is that the computational work in JDQR and JDQZ consists of two parts of a different nature: one part is for the *inner iteration* process, in which a correction equation is (approximately) solved; the other part is for the *outer iteration*, in which an approximation for the (generalized) Schur pair is constructed. If in the inner iteration the correction equation is solved more accurately, then the number of outer iterations may decrease. Therefore, it would be misleading to monitor the total number of matrix multiplications. It might give a bad impression of the total costs, because most of the matrices are sparse and therefore the dot products and vector updates in the outer and the inner iteration represent substantial costs in JDQR and JDQZ.

Furthermore, we have plotted the entire convergence behavior. This means that the convergence history of the residuals of all sub-sequentially selected approximate eigenpairs is plotted. Whenever the residual norm curve drops below the acceptation level, indicated by the dotted horizontal line, an eigenvalue is accepted and the search process for the next one is continued. A large residual norm in the step immediately after acceptation marks the start of a new search.

Construction of suitable initial subspaces. Specifically in the first few steps of the process the Ritz or Petrov vectors are usually poor approximations of the wanted eigenvectors, and the target value τ may be relatively (much) closer to the wanted eigenvalues than any of the approximate eigenvalues. In these cases, the correction equations (19) and (49), lead to relatively poor expansions of the search subspace. To see this, recall that the wanted eigenvector would be in the new search subspace if this space would have been expanded by the exact solution for the correction equation with the wanted eigenvalue instead of $\widetilde{\lambda}$ (cf. §2.1). This observation indicates how to improve the expansion in the first few steps: take in the correction equation τ instead of $\widetilde{\lambda}$. To detect whether $\widetilde{\lambda}$ is close enough to replace τ, we monitor the norm of the residual: we take $\widetilde{\lambda}$ ($\langle\widetilde{\alpha},\widetilde{\beta}\rangle$) instead of τ ($\langle\tau,1\rangle$) in the correction equation as soon as the first residual norm drops below a threshold value ϵ_{tr}.

Moreover, in all experiments we used GMRES$_1$ for the first $j_{\min}$ iterations, in order to build up a search subspace span$\{\mathbf{V}\}$ in a relatively inexpensive way. Especially when a preconditioner is involved, this approach can be justified with arguments similar those in the preceding paragraph (cf. [18, §9.4]).

In our experiments we will vary values for some of the parameters in JDQR and JDQZ. For easy reference, we recall their meaning:

parameter	*description*
τ	the target value
$k_{\max}$	the number of wanted Schur pairs
ϵ	the stopping tolerance in the outer iteration
ϵ_{tr}	the tracking threshold (cf. §2.5.1)

4.1 The influence of the correction equation. The purpose of this example is to show the effect of a more accurate solution of the correction equation. We consider the Square Dielectric Waveguide standard eigenproblem DW4096 of order 4096 [2]. The problem comes from an integrated circuit application. The rightmost eigenvalues and their eigenvectors are wanted.

We took $\tau = 1.0$, $k_{\max} = 5$, $\epsilon = 10^{-9}$, $\epsilon_{\mathrm{tr}} = 10^{-4}$, and we have not used preconditioning.

The computed eigenvalues are given in TAB. 4. The convergence history is plotted in FIG. 2 for JDQR, for GMRES$_1$ and GMRES$_{10}$. A summary of the number of iterations, the number of matrix multiplications (MVs), and the number of flops is given in TAB. 5.

When solving the correction equation more accurately, the number of MVs is increased, but the number of outer iterations is reduced significantly (see TAB. 5), resulting in a much better overall performance.

With GMRES$_1$ the search subspace is the span of the residuals, and in that case JD (with implicit restart) generates the same subspaces as Implicitly Restarted Arnoldi [22]. The eigenvalues are not well separated in this case and therefore Arnoldi converges only slowly. This explains the poor convergence of JDQR with GMRES$_1$.

Note that after an initial stagnation phase, JDQR converges quite fast. For the next eigenvalues there is no such initial stagnation. Apparently, in the iterations for the first eigenvalue, components for the next Schur vectors are already collected in span$\{\mathbf{V}\}$ (cf. §2.6).

$8.6479e-01$
$8.6528e-01$
$8.6564e-01$
$8.6594e-01$
$8.6657e-01$

TABLE 4. *5 eigenvalues of* DW4096, *computed by JDQR (cf. §4.1).*

Method for the correction equation	JDQR iterations	MVs	flops $\times 10^6$
GMRES_1	229	229	$2.892e+02$
GMRES_{10}	52	341	$1.311e+02$

TABLE 5. *Summary of results for* DW4096 *(cf. §4.1).*

4.1.1 Comparison with Implicitly Restarted Arnoldi. In exact arithmetic, JD with implicit restart and with GMRES_1 generates the same search subspaces as Implicitly Restarted Arnoldi (IRA) [22] if the same restart parameters (maximal and minimal dimension of the search subspaces) are used. In our experiment, IRA was implicitly restarted with a 10-dimensional Krylov subspace whenever the dimension of the Krylov subspace was 15, searching for the eigenvalue with largest real part. Since IRA explores Krylov subspaces, it requires less inner products per step and it is a little more efficient per step. However, for the present example, IRA still did not detect any eigenpair at the point of termination of the process at which 440×10^6 flops were needed. Apparently, our approach is somewhat more stable.

JD can be improved in this case with GMRES_m for $m > 1$. Similar improvements for IRA were not possible here. IRA does not allow for variable preconditioning and, we could not apply "Shift-and-Invert" techniques: exact inversion (i.e., exact factorization) was not feasible with

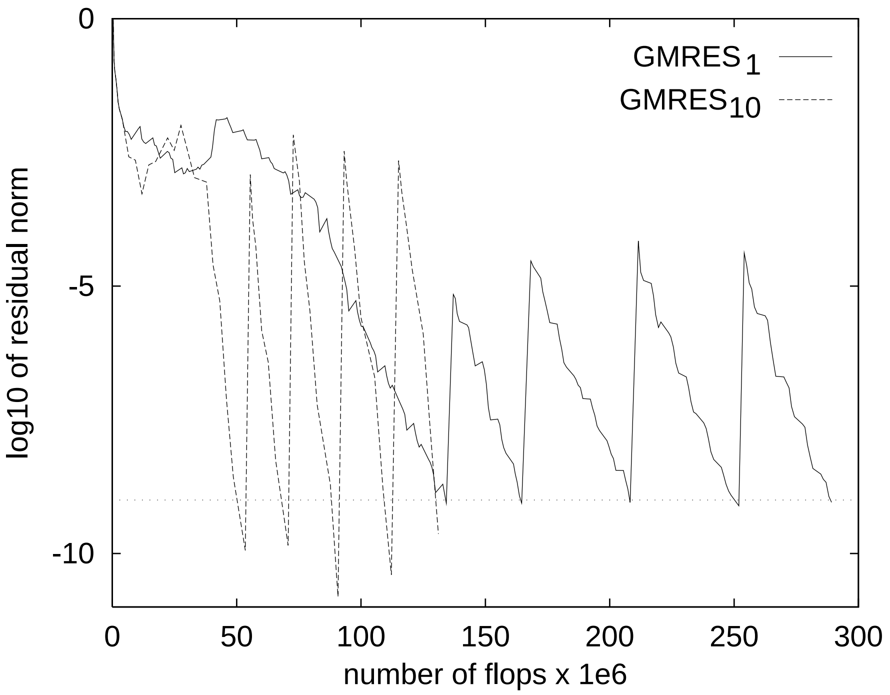


FIG. 2. *Convergence history for* DW4096, *showing the effect of solving the correction equations more accurately (cf. §4.1).*

our software versions (the band-width of the matrix is ≥ 2000).

4.2 The effect of preconditioning. When increasing the number of steps in GMRES, the correction equation will be solved more accurately, and the number of outer iterations may decrease as we have seen. But sometimes we need too many inner iterations with GMRES for acceptable convergence of the outer iterations. However, with appropriate preconditioning, we may see a dramatic improvement.

We consider the standard eigenproblem BWM2000 of order 2000 for the Brusselator wave model [2], [16]. The problem models the concentration waves for reaction and transport interaction of chemical solutions in a tubular reactor. Our task is to determine the eigenvalues with largest real part in order to verify whether their real parts are positive or negative (corresponding to stable or unstable modes).

For this problem, we have selected: $\tau = 1.0$, $k_{\max} = 5$, $\epsilon = 10^{-9}$, and $\epsilon_{\rm tr} = 10^{-4}$.

The computed eigenvalues are listed in TAB. 6. The convergence history is plotted in FIG. 3 for JDQR with unpreconditioned GMRES_{10}, and with GMRES_{10} + ILU(0) preconditioning. A summary of the results is given in TAB. 7.

From FIG. 3 we see that JDQR with GMRES_{10} does not converge (we checked even up to GMRES_{50}, with little or no improvement), but with preconditioning JDQR performs rather well. Again we see that the speed of convergence for the first eigenvalue is somewhat slower than the speed of convergence for the other eigenvalues. Note that, although the projections in the correction equation become more expensive with each detected eigenvalue, the computational work for each eigenvalue is roughly constant, except for the first eigenvalue.

$-1.8000e+00$	$+$	$3.0327e+00\,i$
$-6.7500e-01$	$-$	$2.5287e+00\,i$
$-6.7500e-01$	$+$	$2.5287e+00\,i$
$2.4427e-07$	$-$	$2.1395e+00\,i$
$2.4427e-07$	$+$	$2.1395e+00\,i$

TABLE 6. *5 eigenvalues of* BWM2000*, computed by JDQR (cf.* §4.2*).*

Method for the correction equation	JDQR iterations	MVs	flops $\times 10^6$
GMRES_{10} (no convergence)	$*$	$*$	$*$
GMRES_{10} + ILU(0)	45	213	$4.518e+01$

TABLE 7. *Summary of results for* BWM2000 *(cf.* §4.2*).*

4.3 Multiple eigenvalues. In this example multiple eigenvalues are involved: the Chuck Matrix of order 656 (CK656) [2]. The goal is to compute the eigenvalues with magnitude greater than 1.0. The eigenvalues appear in clusters: each cluster consists of two pairs of almost multiple eigenvalues.

For this problem, we have selected: $\tau = 5.0$, $k_{\max} = 10$, $\epsilon = 10^{-9}$, and $\epsilon_{\rm tr} = 10^{-4}$.

The computed eigenvalues are listed in TAB. 8. The convergence history is plotted in FIG. 4 for JDQR with GMRES_1, GMRES_{10}, and BiCGstab(2), all with ILU(2) preconditioning, for the correction equation. A summary of the results is given in TAB. 9.

For all three combinations the multiple eigenvalues are easily detected. For this example, GMRES_{10} results in a somewhat better overall performance in comparison with GMRES_1, but the combination with BiCGstab(2) is the clear winner (cf. TAB. 9).

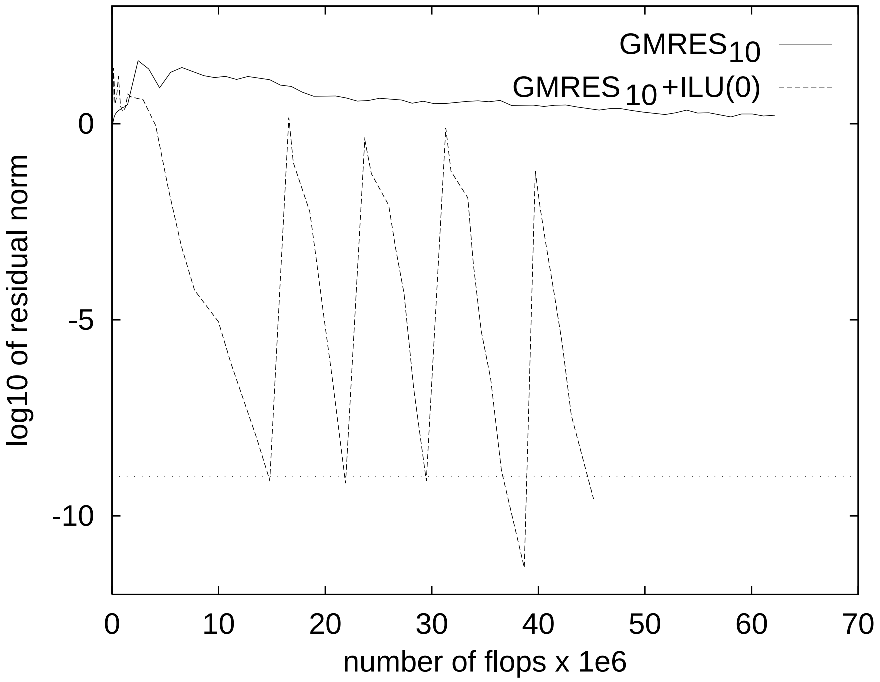


FIG. 3. *Convergence history for* BWM2000*, illustrating the effect of including preconditioning in the solver of the correction equation (cf. §4.2).*

The initial fast convergence may be explained by the fact the target τ is close to the rightmost double eigenvalue, which is relatively well separated from the other eigenvalues.

$1.1980e+00$	$1.1980e+00$
$1.5940e+00$	$1.5940e+00$
$1.4120e+00$	$1.4120e+00$
$1.4190e+00$	$1.4190e+00$
$5.5024e+00$	$5.5024e+00$

TABLE 8. *10 eigenvalues of* CK656*, computed by JDQR (cf. §4.3).*

Method for the correction equation	JDQR iterations	MVs	flops $\times 10^6$
GMRES_1	121	121	$3.346e+01$
GMRES_{10}	64	291	$2.959e+01$
BiCGstab(2)	48	226	$2.260e+01$

TABLE 9. *Summary of results for* CK656 *(cf. §4.3).*

Implicitly Restarted Shift-and-Invert Arnoldi with shift $\tau = 5.0$ required 192×10^6 flops for detecting the wanted 10 eigenvalues. The LU-factorization for exact inversion took $\approx 89\%$ of the computational costs. Even if the costs for the factorization are not taken into account, the IRA process is still not much more efficient than JDQR with BiCGstab(2)+ILU(2).
In our experiment, IRA was implicitly restarted with a 15-dimensional Krylov subspace whenever the dimension of the Krylov subspace was 20.

4.4 Harmonic Ritz values. The JDQR algorithm computes a partial Schur form for the standard eigenproblem with Ritz pairs for the Schur pairs. However, with JDQZ for $\mathbf{B} = \mathbf{I}$,

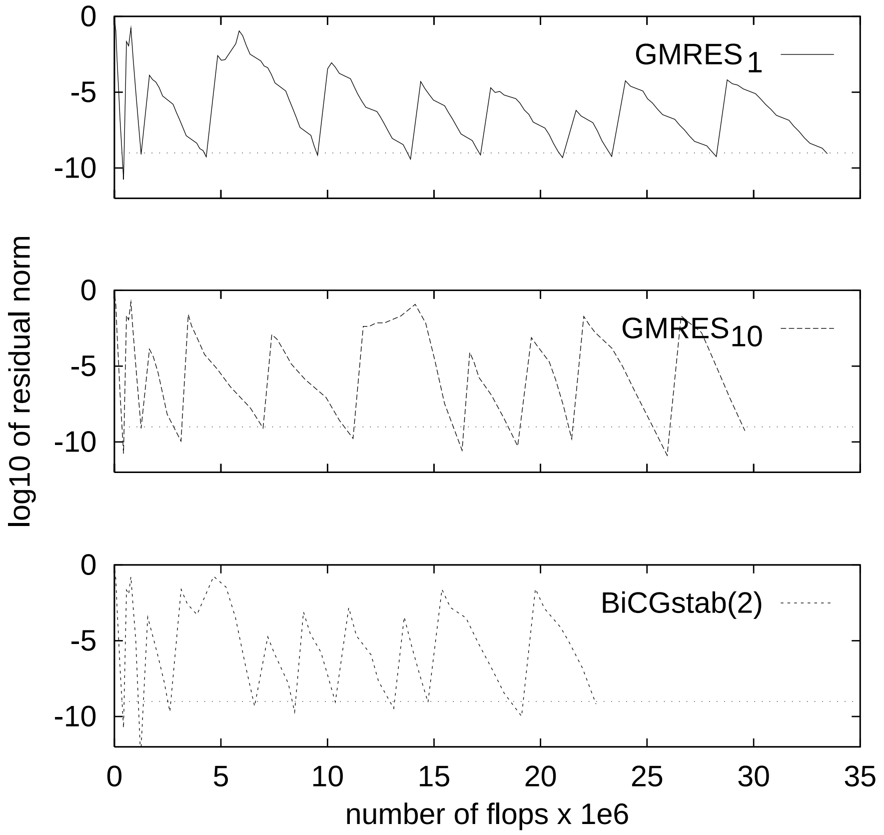


FIG. 4. *Convergence history for* CK656. *Also multiple eigenvalues are easily detected (cf.* §4.3*).*

we can also compute a partial Schur form for the standard eigenproblem with harmonic Petrov pairs. Here we give an example that illustrates the improved convergence behavior for harmonic Petrov values.

We consider the Quebec Hydroelectric Power System problem QH882 of order 882 [2]. This matrix represents the Hydro-Quebec power system's small-signal model. The eigenvalues λ of interest are the eigenvalues in the box $-300 < \mathrm{Re}(\lambda) < 100$, $0 < \mathrm{Im}(\lambda) < 120\pi$ in the complex plane.

For this problem, we have selected: $\tau = -150.0+180.0\,i$, $k_{\max} = 5$, $\epsilon = 10^{-6}$, and $\epsilon_{\mathrm{tr}} = 10^{-3}$.

The computed eigenvalues are given in TAB. 10. The convergence history is plotted in FIG. 7, for JDQR, JDQZ with the adaptive choice of test subspace (cf. §3.1.2), and JDQZ with the harmonic choice (cf. §3.5.1).

This problem is rather difficult: the eigenvalues in the neighborhood of τ are in the interior of the spectrum, see FIG. 5 and 6. For all three methods, the correction equation was solved with GMRES$_{20}$, preconditioned with the exact inverse of $\mathbf{A} - \tau\,\mathbf{I}$. A summary of the results is given in TAB. 9.

Although the computational complexity of JDQR is less than the computational complexity of JDQZ (cf. TAB. 2 and 3), it is not the most efficient method here. From the irregular convergence behavior of JDQR in FIG. 7 we may conclude that JDQR has problems in selecting the "correct" Ritz pairs and as a result the convergence is delayed. As anticipated (cf. §3.1), JDQZ with the harmonic choice of test subspace makes better selections, as is indicated by the smooth convergence, and hence, its performance is much better. The adaptive version of JDQZ converges faster than JDQR, but not faster than the harmonic version of JDQZ. Apparently, for fast convergence in this case, it is more important to have "correct" Petrov approximations, than to focus on a more optimal test subspace.

$-1.8665e+02$	$+$	$1.9464e+02\,i$
$-1.6661e+02$	$+$	$1.9768e+02\,i$
$-1.6349e+02$	$+$	$1.9524e+02\,i$
$-1.4913e+02$	$+$	$1.9729e+02\,i$
$-1.3607e+02$	$+$	$2.0215e+02\,i$

TABLE 10. *5 eigenvalues of* QH882*, computed by JDQR (cf.* §4.4*).*

Method	Iterations	MVs	flops $\times 10^6$
JDQR	99	1482	$1.221e+02$
JDQZ Adaptive	62	802	$8.252e+01$
JDQZ Harmonic	57	665	$7.133e+01$

TABLE 11. *Summary of results for* QH882 *(cf.* §4.4*).*

4.5 Tracking. Tracking was proposed in §2.5.1, in order to avoid the performance degeneration, that is caused by irregular convergence behavior of Ritz values. Here we illustrate (in FIG. 8) the effects one may see without tracking for JDQR. We applied JDQR to the example of §4.4 with the same choice of parameters, except for $\epsilon_{\rm tr}$ ($\epsilon_{\rm tr} = 0$ here).

In the previous example we have already seen that the convergence behavior of JDQR was rather irregular. By leaving out the tracking mechanism, the irregularities are even more pronounced. Eventually JDQR loses track completely and stagnates. The peaks in the convergence behavior show that sometimes the Ritz pair, that is selected in the JDQR process, does not correspond to the close-by Schur pair. As a result the search subspace is expanded in a poor direction. Clearly, for this example, this may lead to failure of convergence.

It also appears that, as was anticipated, the convergence behavior of JDQZ with the harmonic choice of test subspace is hardly affected by leaving out the tracking mechanism. Also the adaptive choice does pretty well without, but this does not mean that it always selects the "correct" Ritz pairs, as we will see in our final example in §4.8.

4.6 The influence of $\mathbf{Q}_k$ and $\mathbf{Z}_k$ in the correction equation. In this example we show that the projections with detected Schur vectors (cf. (47)) are very essential in the correction equation (cf. §2.6) and we show what happens when these projections are neglected. Note that we take the Jacobi projections (with $\widetilde{\mathbf{q}}$ and $\widetilde{\mathbf{z}}$) still into account.

We consider the Bounded Finline Dielectric Waveguide generalized eigenproblem BFW782 [2] of order 782. This problem stems from a finite element discretization of the Maxwell equation

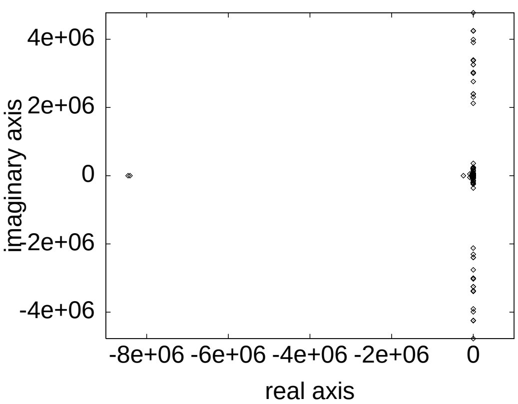


FIG. 5. *Spectrum of* QH882 *(cf.* §4.4*).*

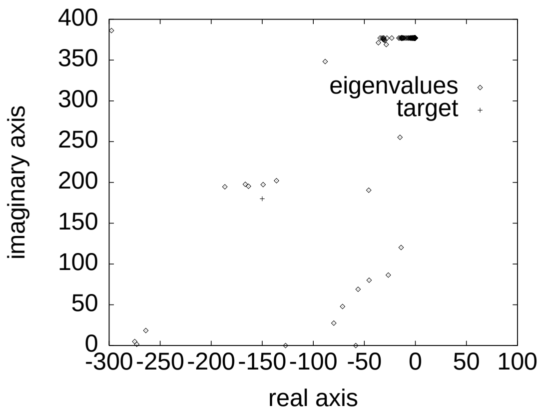


FIG. 6. *Part of the spectrum of* QH882.

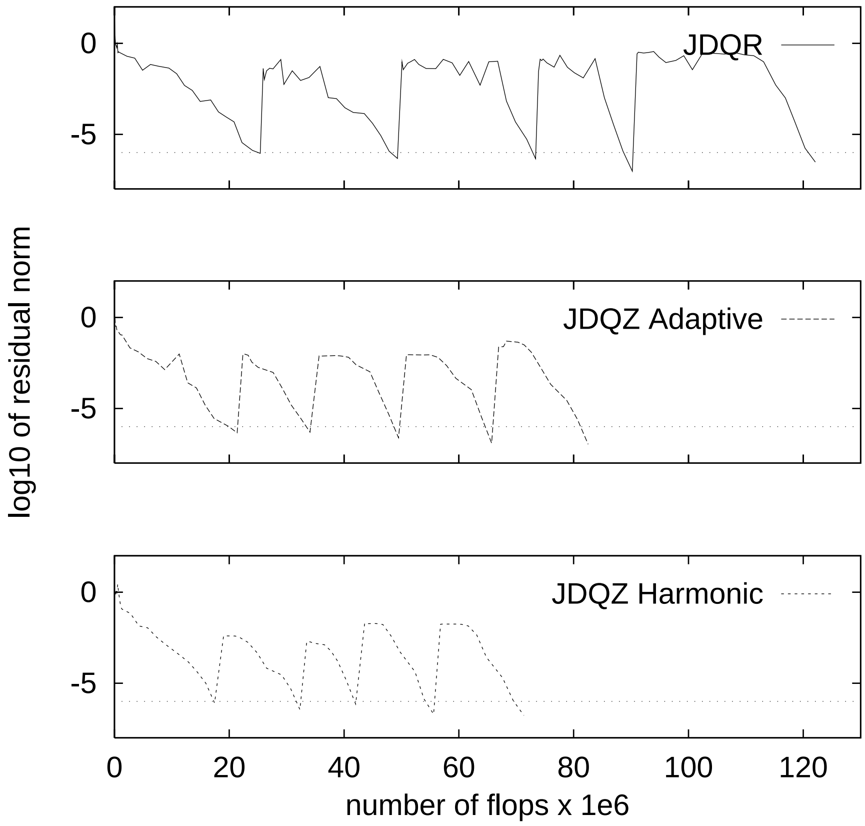


FIG. 7. *Convergence history for* QH882 *obtained with the tracking strategy (for all variants). Although* QH882 *is a standard eigenproblem, for computing interior eigenvalues it is more efficient to use test subspaces that are different from the search subspaces (middle and lower figure). A better selection of the Petrov pair (lower figure) appears to compensate for a less optimum expansion of the test subspace. (cf.* §4.4*).*

for propagating modes and magnetic field profiles of a rectangular waveguide filled with dielectric and PEC structures. The resulting matrix $\mathbf{A}$ is nonsymmetric and the matrix $\mathbf{B}$ is positive definite. Of special interest are the generalized eigenvalues $\langle\alpha,\beta\rangle$ with positive real part (i.e., $\mathrm{Re}(\alpha/\beta)\geq 0$) and their corresponding eigenvectors.

For this problem, the parameters were set to: $\tau = 2750.0$, $k_{\max} = 5$, $\epsilon = 10^{-9}$, and $\epsilon_{\mathrm{tr}} = 10^{-6}$.

The spectrum of this matrix pair is shown in FIG. 9. A magnification of the region of interest is plotted in FIG. 10. The computed generalized eigenvalues, represented as α/β, are given in TAB. 12. With JDQZ we discovered all 4 positive generalized eigenvalues.

The convergence history is plotted in FIG. 11, for the harmonic version of JDQZ with GMRES_{10}, and for BiCGstab(2). A summary of the results is given in TAB. 13. We see that JDQZ converges quite nicely for GMRES_{10} and BiCGstab(2). It should be noted that although it seems that with BiCGstab(2) only 4 generalized eigenvalue are computed, in fact 5 generalized eigenvalue are computed: the 2 rightmost generalized eigenvalue, that are relatively close, are found almost simultaneously.

In FIG. 12 the convergence behavior of JDQZ with GMRES_{10}, and with BiCGstab(2), is given for the case where the correction equation (47) is solved without taking into account the projections involving $\mathbf{Q}_k$ and $\mathbf{Z}_k$. Of course, the correction equations that are used include the rank one projection involving $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{z}}$: these projections are essential for Jacobi-Davidson. Furthermore, deflation in this case is realized by making the approximate solution of the correction equation orthogonal to the detected Schur vectors with modified Gram-Schmidt. By doing the latter *twice*, the overall performance improved significantly: in the results shown here (cf.

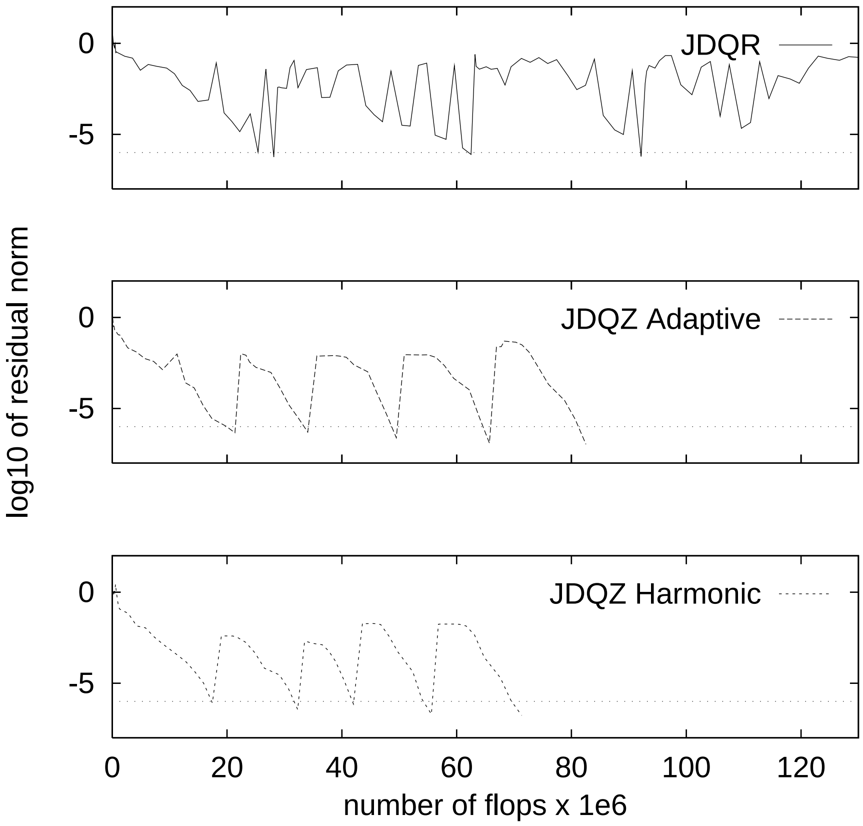


FIG. 8. *Convergence history for* QH882 *without tracking. For interior eigenvalues, the tracking strategy improves JDQR significantly (compare the present upper figure with the upper figure in* FIG. *7), while there is no improvement for the two JDQZ variants (compare the two middle figures and the two lower figures of the present figure and* FIG. *7) (cf.* §4.5*)*

FIG. 12) modified Gram-Schmidt is applied twice.

However, as explained in §2.6, we do not benefit from an improved operator in the inner iteration. Although the resulting algorithm is computationally cheaper, FIG. 12 shows that this does not lead to an overall better performance: the speed of convergence becomes increasingly slower and even stagnates eventually.

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

TABLE 12. *5 generalized eigenvalues of* BFW782*, computed by JDQZ (cf.* §4.6*).*

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

TABLE 13. *Summary of results for* BFW782 *(cf.* §4.6*).*

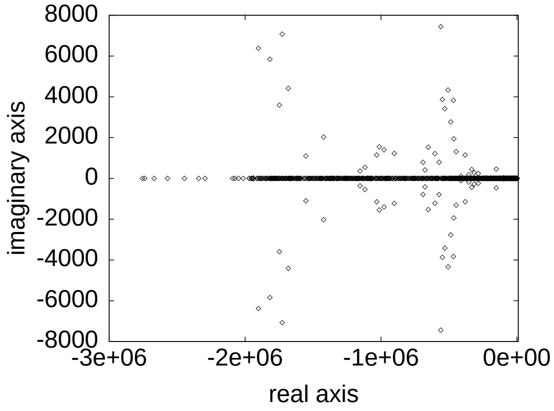


FIG. 9. *Spectrum of* BFW782 *(cf. §4.6).*

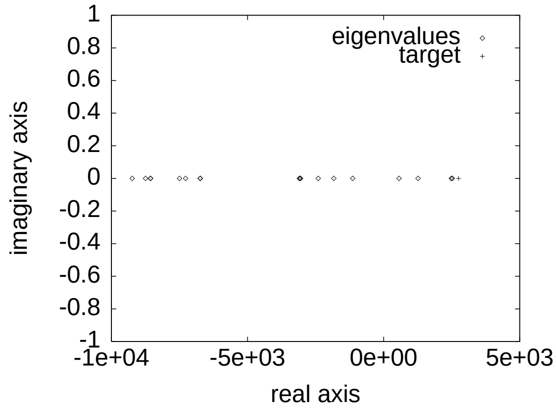


FIG. 10. *Part of the spectrum of* BFW782.

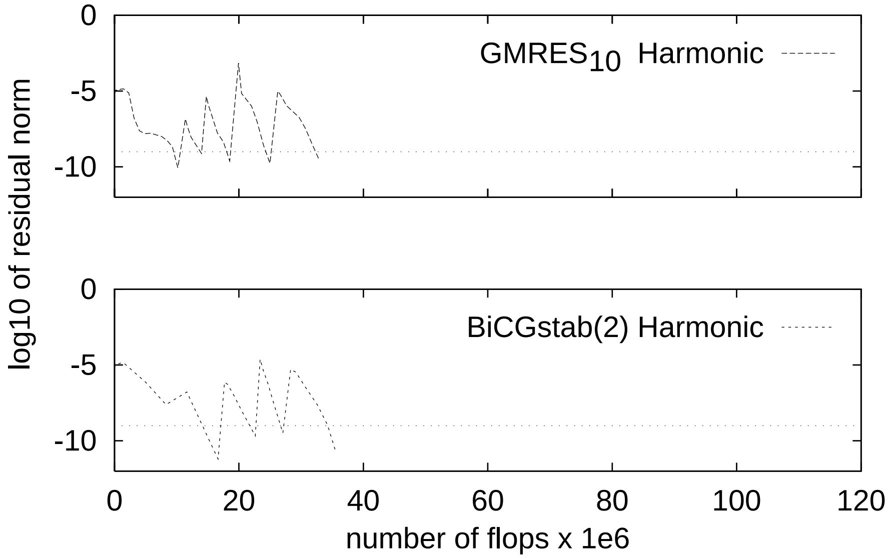


FIG. 11. *Convergence history for* BFW782.

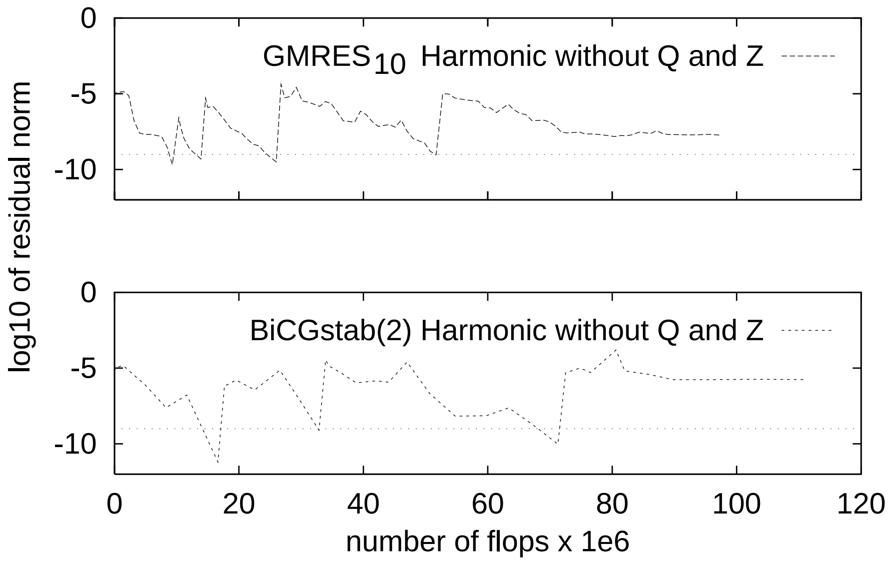


FIG. 12. *Convergence history for* BFW782 *without deflating the matrices in the correction equations, with respect to the detected Schur vectors. Without deflation the convergence of the linear solver for the correction is much slower (compare the above figures with those in* FIG. 11*) (cf. §4.6).*

4.7 More multiple eigenvalues. We consider the eigenproblem

$$\Delta u = \lambda u,$$

with Neumann boundary conditions on the cube $[0,4]^3$. Finite element discretization of this equation on an $11 \times 11 \times 11$ regular grid, with tetrahedral elements and linear interpolation functions, leads to a generalized eigenproblem of order 1331 (AC1331). It has one positive generalized eigenvalue $\langle \alpha, \beta \rangle$ relatively close to zero (i.e., $\alpha/\beta \approx 0$). The other generalized eigenvalues are also positive and may be doublets or even triplets.
For this problem, the parameters were set to: $\tau = 0.0$, $k_{\max} = 15$, $\epsilon = 10^{-9}$, and $\epsilon_{\mathrm{tr}} = 10^{-4}$.

The computed 15 leftmost generalized eigenvalues, represented as α/β are given in TAB. 14. The residual norm versus the number of flops is plotted in FIG. 13 for the harmonic version of JDQZ with GMRES_{10} and with BiCGstab(2), respectively. A summary of the results is given in TAB. 15.

From the plots we see the effect that multiple generalized eigenvalues may have on the convergence behavior. JDQZ converges initially quite fast until the point that it "discovers" that the generalized eigenvalue is actually double or triple. The convergence speed stagnates for a few iterations (2 or 3 peaks in the plot with GMRES, and a plateau in the plot with BiCGstab(2)), after which they are discovered quickly one after another. This behavior is in agreement with §2.6: during the stagnation phase components of other Schur vectors are amplified in the inner iteration and collected in the search subspace, leading to faster convergence for the next Schur pairs. The stagnation can be explained by the fact, that with rank 1 Jacobi projections the correction equation may become (nearly) singular when selecting Petrov approximations for multiple generalized eigenvalues. The iterative methods used for solving the correction equation often suffer from this (see also [25]). (Variable) block versions of the correction equation that take this multiplicity into account may be preferable in such cases, but this falls outside the scope of this paper.

$3.8178e-02$	$1.4503e+05$	$2.9540e+05$
$7.1673e+04$	$1.4503e+05$	$3.7044e+05$
$7.1673e+04$	$2.1811e+05$	$3.7044e+05$
$7.1673e+04$	$2.9411e+05$	$3.7044e+05$
$1.4503e+05$	$2.9411e+05$	$3.7242e+05$

TABLE 14. *15 generalized eigenvalues of* AC1331, *computed by JDQZ (cf. §4.7).*

Method for the correction equation	JDQZ iterations	MVs	flops $\times 10^6$
GMRES_{10}	93	601	$2.325e+03$
BiCGstab(2)	61	1253	$3.368e+03$

TABLE 15. *Summary of results for* AC1331 *(cf. §4.7).*

4.8 Harmonic Ritz values for generalized problems. Our last example shows again that for interior generalized eigenvalues the harmonic version JDQZ is superior to the adaptive version.

We consider the MHD416 generalized eigenproblem of order 416 [2], [18], [3]. This problem stems from a magnetohydrodynamics (MHD) model, where the interaction of hot plasma and a magnetic field is studied. The matrix $\mathbf{A}$ is non-Hermitian and the matrix $\mathbf{B}$ is Hermitian positive definite. Our goal is to compute interior generalized eigenvalues corresponding to the so called "Alfvén" branch of the spectrum, see FIG. 14 and 15.

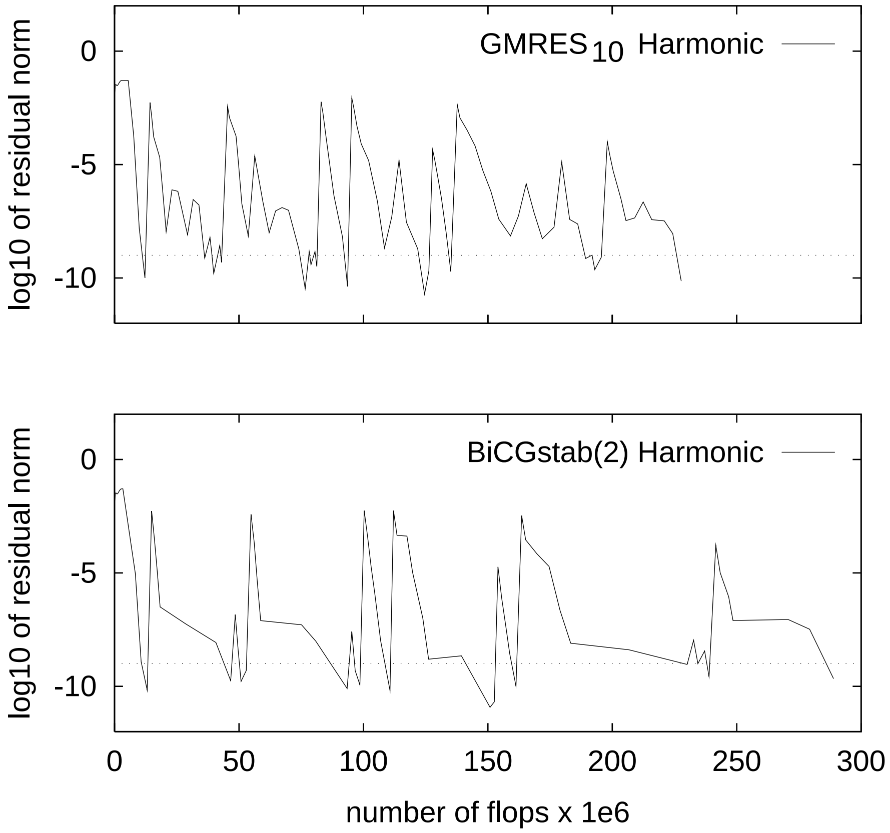


FIG. 13. *Convergence history for* AC1331*: stagnation followed by fast detection of triple generalized eigenvalues (cf.* §4.7*).*

For this problem, the parameters were set to: $\tau = -0.1 + 0.5\,i$, $k_{\max} = 20$, $\epsilon = 10^{-9}$, and $\epsilon_{\mathrm{tr}} = 10^{-4}$.

The computed generalized eigenvalues are plotted in FIG. 16. The convergence history for the adaptive and the harmonic version of JDQZ with GMRES_1 are plotted in FIG. 17. The exact inverse of $\mathbf{A} - \tau\,\mathbf{B}$ (for τ fixed) was used as preconditioner.

Note the irregular behavior of the adaptive version of JDQZ. The tracking mechanism fails and the convergence becomes more and more problematic as the iteration count increases. It even stagnates after a while. The harmonic version of JDQZ does not encounter such problems here. For all generalized eigenvalues the rate of convergence is almost the same: in the computation for one Schur pair, the search subspace apparently accumulates components for next Schur pairs as well.

5 Conclusions. We have proposed two algorithms, JDQR and JDQZ, for computing several selected eigenpair approximations for standard and generalized eigenproblems, respectively. The methods are based on the Jacobi-Davidson method and compute iteratively a partial (generalized) Schur form with (generalized) eigenvalues near a user-specified target value. For both methods, no exact inversion of any matrix is strictly necessary, so that they are suitable for solving large eigenproblems.

Fast convergence is obtained with a projected correction equation that is solved (approximately) by iterative methods with appropriate preconditioning. The convergence of JDQR and JDQZ is asymptotically quadratical if this correction equation is solved exactly. Furthermore, while converging to a particular Schur pair, the search subspace accumulates components of other Schur pairs with (generalized) eigenvalues near the target as well. This usually leads to faster convergence for the next eigenpairs.

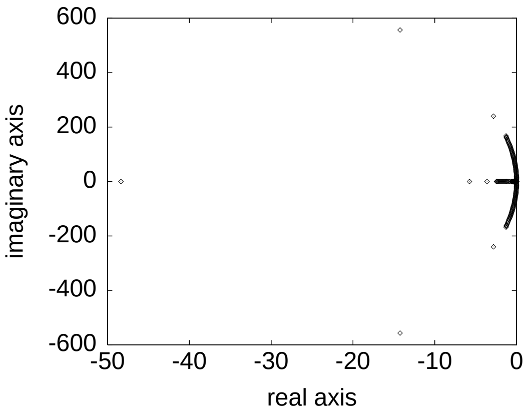


FIG. 14. *Spectrum of* MHD416 *(cf.* §4.8*).*

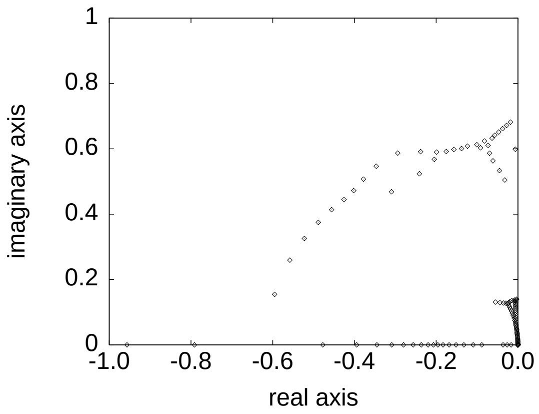


FIG. 15. *Alfvén branch of* MHD416 *(cf.* §4.8*).*

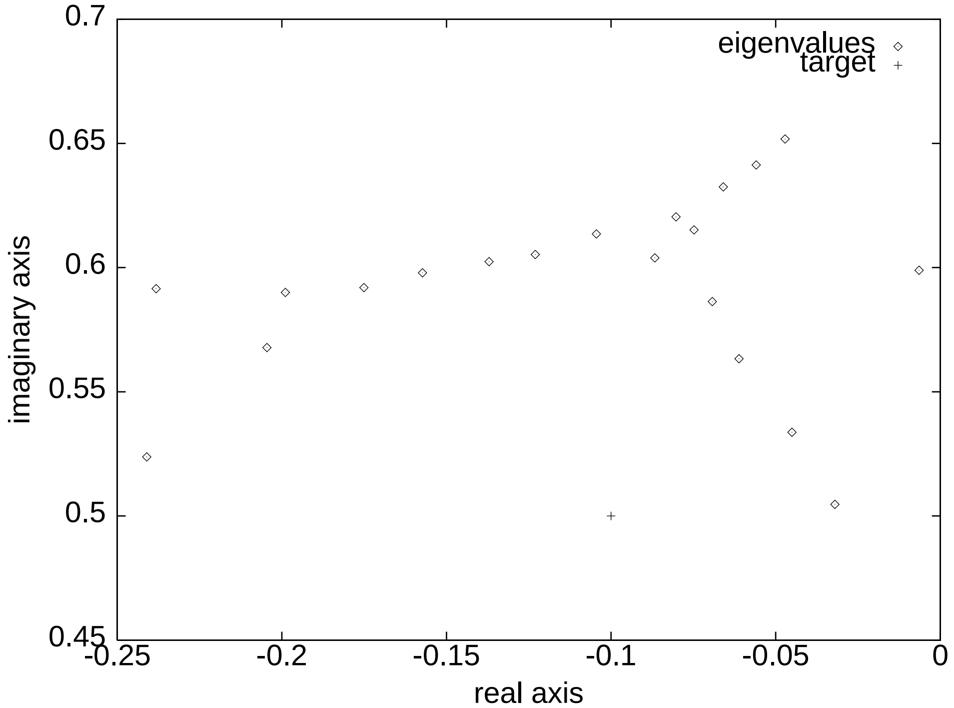


FIG. 16. *20 generalized eigenvalues computed by JDQZ for* MHD416 *(cf.* §4.8*).*

The dimension of the involved subspaces can be controlled by an efficient implicit restart technique in such a way that the most relevant part of the subspace is maintained at restart.

The algorithms incorporate simple mechanisms for selecting the wanted eigenpair approximations. Also multiple (generalized) eigenvalues can be detected.

Whereas in the Jacobi-Davidson method the test subspace can be chosen arbitrarily, in the JDQZ algorithm essentially two choices for the test subspace remain: the (adaptive) standard Petrov value choice, and the harmonic Petrov value choice. It is argued, and confirmed as well by our experiments, that especially for interior eigenvalues, the harmonic approach is also superior for generalized eigenproblems

Acknowledgements. We acknowledge helpful discussions with Paul Smit and with Beresford Parlett (on naming conventions). Michiel Kooper kindly gave us the results of runs with Implicitly Restarted Arnoldi.

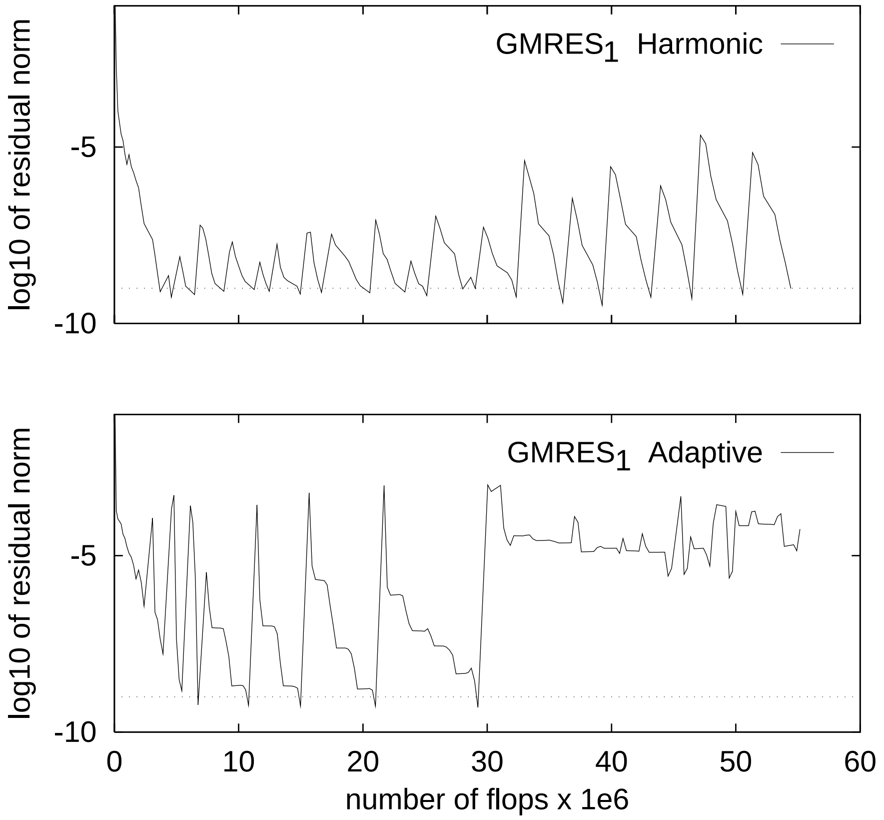


FIG. 17. *Convergence history of JDQZ for* MHD416. *harmonic Petrov values allow a good selection of Petrov pairs for computing interior generalized eigenvalues (upper figure), also for generalized problems (cf.* §4.8*).*

REFERENCES

[1] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. D. CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, S. OSTROUCHOV, AND D. C. SORENSEN, *LAPACK Users' Guide*, SIAM, Philadelphia, 1992.

[2] Z. BAI, R. BARRET, D. DAY, J. DEMMEL, AND J. DONGARRA, *Test matrix collection (non-Hermitian eigenvalue problems)*, Department of Mathematics, University of Kentucky, June 6, 1995.

[3] J. G. L. BOOTEN, D. R. FOKKEMA, G. L. G. SLEIJPEN, AND H. A. VAN DER VORST, *Jacobi-Davidson methods for generalized MHD-eigenvalue problems*, in ICIAM proceedings, July 3–7 1995. To appear in Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM).

[4] F. CHATELIN AND V. FRAYSSÉ, *Qualitative Computing; Elements of a theory for finite precision computation*, Thomson–CSF, June 10, 1993.

[5] R. S. DEMBO, S. C. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[6] J. FRANCIS, *The QR transformation: A unitary analogue to the LR transformation, parts I and II*, Comput. J., 4 (1961), pp. 265–272, 332–345.

[7] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore and London, 1989. Second Edition.

[8] I. GUSTAFSSON, *A class of first order factorizations methods*, BIT, 18 (1978), pp. 142–156.

[9] R. B. LEHOUCQ, *Analysis and Implementation of an Implicitly Restarted Arnoldi Iteration*, PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 1995.

[10] J. A. MEIJERINK AND H. A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.

[11] C. B. MOLER AND G. W. STEWART, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.

[12] R. B. MORGAN, *Computing interior eigenvalues of large matrices*, Linear Algebra Appl., 154/156 (1991), pp. 289–309.

[13] 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.

[14] A. Ruhe, *Perturbation bounds for means of eigenvalues and invariant subspaces*, BIT, 10 (1970), pp. 343–354.
[15] ———, *Rational Krylov algorithms for nonsymmetric eigenvalue problems II. Matrix Pairs*, Lin. Alg. and its Appl., 197/198 (1994), pp. 283–295.
[16] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, U.K., 1992.
[17] Y. Saad and M. H. Schultz, *GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[18] G. L. G. Sleijpen, J. G. L. Booten, D. R. Fokkema, and H. A. Van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems: Part I*, Preprint 923, Departement of Mathematics, Utrecht University, Utrecht, Revised version, November 1995.
[19] G. L. G. Sleijpen and D. R. Fokkema, *BiCGstab(ℓ) for linear equations involving matrices with complex spectrum*, Electronic Transactions on Numerical Analysis (ETNA), 1 (1993), pp. 11–32.
[20] G. L. G. Sleijpen and H. A. Van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, Preprint 856, Departement of Mathematics, Utrecht University, Utrecht, Revised version, May 1995. To appear in SIAM J. Matrix Anal. Appl.
[21] ———, *The Jacobi-Davidson method for eigenvalue problems and its relation to accelerated inexact Newton schemes*, preprint, Department of Mathematics, Utrecht University, Utrecht, 1995. To appear in the Proceedings of the "Second IMACS International Symposium on Iterative Methods in Linear Algebra", June 17-20, 1995, Blagoevgrad, "Contemporary Mathematics (CONM)" of the AMS.
[22] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
[23] G. W. Stewart, *Algorithm 406 HQR3 and EXCHNG: Fortran subroutines for calculating and ordering eigenvalues of a real upper Hessenberg matrix*, ACM Trans. Math. Software, 2 (1976), pp. 275–280.
[24] G. W. Stewart and Ji guang Sun, *Matrix Perturbation Theory*, Academic Press, San Diego, California, 1990.
[25] A. van der Sluis and H. A. Van der Vorst, *The rate of convergence of conjugate gradients*, Numerische Mathematik, 48 (1986), pp. 543–560.
[26] H. A. Van der Vorst, *Preconditioning by Incomplete Decompositions*, PhD thesis, Utrecht University, Utrecht, The Netherlands, 1982.
[27] P. Van Dooren, *A generalized eigenvalue approach for solving Ricatty equations*, SIAM J. Sci. Comput., 2 (1981), pp. 121–135.
[28] ———, *Algorithm 590, DUSBSP and EXCHQZ: FORTRAN subroutines for computing deflating subspaces with specified spectrum*, ACM Trans. Math. Software, 8 (1982), pp. 376–382.

APPENDIX

A Modified Gram-Schmidt. Here we give a MATLAB implementation of Modified Gram-Schmidt. As input it takes a $(n \times j)$-matrix $\mathbf{V}$, with $\mathbf{V}^*\mathbf{V} = I$, and a n-vector $\mathbf{v}$. On return, $\mathbf{v}$ is made orthogonal to span$\{\mathbf{V}\}$. The j-vector s holds the values of the orthogonalization coefficients involved.

```
function [v, s] = mgs(V, v)
j = size(V, 2);
s = [ ];
for i = 1:j,
    s = [s, V(:, i)' * v];
    v = v - V(:, i) * s(i);
end
```

B Sorting the Schur form. Before giving MATLAB functions for sorting a Schur form, we indicate how the diagonal entries of the upper triangular matrix S in a Schur form can be reordered in any order by unitary transformations without destroying the upper triangular form.

Clearly, it is sufficient to discuss the 2×2 case, and to show that the order of the diagonal entries can be reversed:

$$S \equiv \begin{bmatrix} \lambda & \gamma \\ 0 & \mu \end{bmatrix}.$$

With

$$T \equiv S - \mu I = \begin{bmatrix} \lambda - \mu & \gamma \\ 0 & 0 \end{bmatrix},$$

let G be the Givens rotation such that the first column of TG is zero. Since the first column of G^*TG and the $(2,1)$-entry of $G^*IG = I$ are both zero, we necessarily must have that

$$G^*SGe_1 = G^*TGe_1 + \mu G^*IGe_1 = \begin{bmatrix} \mu \\ 0 \end{bmatrix}.$$

Therefore, G^*SG is upper triangular and similar to S and, consequently, the order of the diagonal entries of G^*SG is reversed with respect to those of S.

The MATLAB functions below implement the sorting of a Schur decomposition $MU = US$. As input qrsort takes the target τ, a unitary matrix U and an upper triangular matrix S. On return, the diagonal entries of S are ordered with respect to τ and the matrix U is updated correspondingly.

The function select detects the position in a sequence of a scalar closest to a value τ. With plane-rotations, the function swap swaps the diagonal entries $S(k,k)$ and $S(k+1,k+1)$ for for $k = j-1, j-2, \ldots, i$, assuming that $j > i$. It updates the other entries of S and U accordingly. The resulting matrix S contains at position (i,i) the value $S(j,j)$ of the original matrix, while the values in the left $i-1 \times i-1$ upper block are unchanged.

```
function [U,S] = qrsort(τ,U,S)
k = size(S,1);
for i = 1:k−1,
    s = diag(S(i:k,i:k));
    j = select(τ,s) + i − 1;
    [U,S] = swap(U,S,j,i);
end

function j = select(τ,s)
[a,j] = min(abs(τ − s));

function [U,S] = swap(U,S,j,i)
for k = j−1:−1:i,
    x = [S(k,k+1), S(k,k) − S(k+1,k+1)];
    G([2,1],[2,1]) = planerot(x*)*;
    S(:,[k,k+1]) = S(:,[k,k+1])G;
    S([k,k+1],:) = G*S([k,k+1],:)
    U(:,[k,k+1]) = U(:,[k,k+1])G;
end
```

C Sorting the generalized Schur form. Before giving the MATLAB functions, we explain how the generalized Schur form can be sorted.

Clearly, it is sufficient to consider the 2×2 case, and to show that the order of the eigenvalue pairs can be reversed. Thereto, consider

$$S_A \equiv \begin{bmatrix} \alpha_1 & \gamma_A \\ 0 & \alpha_2 \end{bmatrix}, \quad S_B \equiv \begin{bmatrix} \beta_1 & \gamma_B \\ 0 & \beta_2 \end{bmatrix}.$$

With

$$T \equiv \beta_2 S_A - \alpha_2 S_B = \begin{bmatrix} \beta_2\alpha_1 - \alpha_2\beta_1 & \beta_2\gamma_A - \alpha_2\gamma_B \\ 0 & 0 \end{bmatrix},$$

let G_R be the Givens rotation such that the first column of TG_R is zero and let G_L be the Givens rotation such that, for some scalar γ,

$$\gamma G_L^* S_B G_R e_1 = \begin{bmatrix} \beta_2 \\ 0 \end{bmatrix}.$$

Since the first column of $G_L^* T G_R$ and the $(2,1)$-entry of $G_L^* S_B G_R$ are both zero, we necessarily must have that

$$\gamma G_L^* S_A G_R e_1 = \begin{bmatrix} \alpha_2 \\ 0 \end{bmatrix}.$$

Hence, both $G_L^* S_A G_R$ and $G_L^* S_B G_R$ are upper triangular and similar to S_A and S_B and consequently the order of the eigenvalue pairs are reversed, where pairs are considered to be identical if the associated quotients are equal.

The MATLAB functions below implement the sorting of a generalized Schur decomposition $M_A U_R = U_L S_A$ and $M_B U_R = U_L S_B$. As input **qzsort** takes the target τ, the upper triangular matrices S_A and S_B, and the unitary matrices U_L and U_R. On return, the eigenvalue pairs of (S_A, S_B) are ordered with respect to τ and the matrices U_L and U_R are updated correspondingly.

The function **select** detects the position of the scalar closest τ.
With plane-rotations, the function **swap** transfers the eigenvalue pairs $\langle S_A(j,j), S_B(j,j)\rangle$ to the position (i,i), leaving the pairs in the positions $(1,1)$, ..., $(i-1,i-1)$ unchanged. The other entries of S_A, S_B, U_L and U_R are updated correspondingly.

```
function [U_L, U_R, S_A, S_B] = qzsort (τ, U_L, U_R, S_A, S_B)
k = size (S_A, 1);
for i = 1:k-1,
    s = diag (S_A(i:k, i:k));
    t = diag (S_B(i:k, i:k));
    j = select (τ, s, t) + i - 1;
    [U_L, U_R, S_A, S_B] = swap (U_L, U_R, S_A, S_B, j, i);
end

function j = select (τ, s, t)
[a, j] = min(abs (τ - s./t));

function [U_L, U_R, S_A, S_B] = swap (U_L, U_R, S_A, S_B, j, i)
for k = j-1:-1:i,
    x = S_B(k+1, k+1)S_A(k, k+1) - S_A(k+1, k+1)S_B(k, k+1);
    x = [x, S_B(k+1, k+1)S_A(k, k) - S_A(k+1, k+1)S_B(k, k)];
```

```
    G([2,1],[2,1]) = planerot (x*)*;
    S_A(:,[k,k+1]) = S_A(:,[k,k+1])G;
    S_B(:,[k,k+1]) = S_B(:,[k,k+1])G;
    U_R(:,[k,k+1]) = U_R(:,[k,k+1])G;
    x = S_A(k,k);
    x = [x; S_A(k+1,k)];
    G = planerot (x);
    S_A([k,k+1],:) = G S_A([k,k+1],:);
    S_B([k,k+1],:) = G S_B([k,k+1],:);
    U_L([k,k+1],:) = U_L([k,k+1],:)G*;
end
```