SIAM J. SCI. COMPUT.
Vol. 20, No. 1, pp. 94–125
© 1998 Society for Industrial and Applied Mathematics

JACOBI–DAVIDSON STYLE QR AND QZ ALGORITHMS FOR THE REDUCTION OF MATRIX PENCILS*

DIEDERIK R. FOKKEMA†‡, GERARD L. G. SLEIJPEN†, AND HENK A. VAN DER VORST†

Abstract. Recently the Jacobi–Davidson subspace iteration method has been introduced as a new powerful technique for solving a variety of eigenproblems. In this paper we will further exploit this method and enhance it with several techniques so that practical and accurate algorithms are obtained. We will present two algorithms, JDQZ for the generalized eigenproblem and JDQR for the standard eigenproblem, that are based on the iterative construction of a (generalized) partial Schur form. 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. An attractive property of our algorithms is that explicit inversion of operators is avoided, which makes them potentially attractive for very large sparse matrix problems.

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. Then we will discuss the use of preconditioning, and, finally, we will illustrate the behavior of our algorithms by a number of well-chosen numerical experiments.

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

AMS subject classifications. 65F15, 65N25

PII. S1064827596300073

1. Introduction.

In this paper we expand on the usage of the Jacobi–Davidson method [26], [24] for the computation of several solutions of the generalized eigenproblem[1]

$$(\beta\,\mathbf{A} - \alpha\,\mathbf{B})\,\mathbf{q} = \mathbf{0}, \tag{1}$$

where $\mathbf{A}$ and $\mathbf{B}$ are large and sparse $(n \times n)$-matrices, which may be complex and/or nonnormal. We will also discuss the standard eigenproblem

$$(\mathbf{A} - \lambda\,\mathbf{I})\,\mathbf{q} = \mathbf{0}. \tag{2}$$

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, and some phenomena are easier to explain.

Our algorithms are based on the Jacobi–Davidson method described in [26] and are adapted for generalized eigenproblems (and other polynomial eigenproblems) in [24]. We have modified the Jacobi–Davidson approach so that partial (generalized) Schur forms are computed. The partial Schur forms have been chosen mainly

*Received by the editors March 4, 1996; accepted for publication (in revised form) March 5, 1997; published electronically August 7, 1998.

http://www.siam.org/journals/sisc/20-1/30007.html

†Department of Mathematics, Utrecht University, P.O. Box 80.010, NL-3508 TA Utrecht, The Netherlands (sleijpen@math.ruu.nl, vorst@math.ruu.nl).

‡Current address: ISE Integrated Systems Engineering AG, Technopark Zürich, Technoparkstrasse 1, CH-8005 Zürich, Switzerland (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 (1), are also called eigenvalues of the matrix pencil (cf., e.g., [30]).

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 low-dimensional search subspace is generated onto which the given eigenproblem is projected. This is the standard "Rayleigh–Ritz" procedure that also underlies the Lanczos, Arnoldi, and Davidson methods. The small projected eigenproblem is solved by standard techniques, and this leads to approximations for the wanted eigenvectors and eigenvalues of the given large problem.

In the Davidson method [5], the solution of a simplified correction equation is used for the expansion of the search subspace. Following an old idea of Jacobi [11], we can also set up a correction equation, acting in the subspace orthogonal to the current eigenvector approximation, which defines an optimal orthogonal expansion of the search subspace. To be more precise, if the exact value for the eigenvalue is known, then the correction equation defines the exact eigenvector. This modification of Davidson's method is referred to as the Jacobi–Davidson method (note that this has nothing to do with the diagonal preconditioning that is popular in combination with Davidson's method).

The "Jacobi" 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 the only place where preconditioning is exploited in the Jacobi–Davidson method. It should be noted that this 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 tailored to find eigenpairs close to a given target value. More details will be given in sections 2.2 and 3.

The small projected problem is reduced to (generalized) Schur form by the QZ method [15] or by QR [8] when $\mathbf{B} = \mathbf{I}$. The construction of the subspace and the projected system too, may be viewed as iterative inexact forms of QZ and QR. For this reason we have named our new methods JDQZ and JDQR, respectively. JDQZ produces a partial generalized Schur form for the generalized eigenproblem: a partial "QZ-decomposition"; JDQR generates a partial Schur form for the standard eigenproblem: a partial "QR-decomposition".

Restarts form an essential ingredient of almost any iterative method, and also for the Jacobi–Davidson method, either for the computation of other eigenpairs, after one eigenpair has converged, or because of limits on the dimension of the subspaces (memory limitations). In any case, the usual restart procedure has the disadvantage that a subspace that may contain very useful information is replaced by one single vector, so that much valuable information may be lost. This problem has been solved elegantly for the Arnoldi method [28] (see also [17]), and our approach (cf. section 2.3) is related to this (see also [26, section 5.3]). In this approach the subspace is suitably filtered to retain as much relevant information as possible. Expansion and filtering are used in a repetitive way.

The generalized eigenproblem in section 2 forms the heart of the paper. It is shown how (implicit) restart techniques and preconditioning can be used in order to get inverse-free computationally efficient algorithms. The resulting algorithm, JDQZ, is enhanced with a deflation technique so that several solutions for the eigenproblem can be computed. The computation of interior eigenvalues is a normally a risky affair, if we want to avoid shift-and-invert operations. We will discuss a rather robust technique based on the idea of harmonic Ritz values [16], [19], [26].

Section 3 focuses on the standard eigenproblem. Of course, the standard eigenproblem can be viewed as a simplification of the generalized eigenproblem, and in an obvious way JDQZ simplifies to the JDQR method for standard eigenproblems. However, we have chosen to pay slightly more attention to the standard eigenproblem since this simplification makes it easier to discuss some computational aspects of our algorithms. In particular, we will consider the problem of preconditioning in more detail, and we will ponder on the observed speed of convergence using well-known arguments.

In section 4, we illustrate the convergence behavior of JDQZ and JDQR with numerical experiments for a number of eigenproblems. Aspects that are investigated concern, among others, the effect of approximation errors on the solution of the correction equation (section 4.1), the effect of preconditioning (section 4.2), multiple eigenvalues (section 4.6), interior eigenvalues (sections 4.3 and 4.7), different approaches for the construction of the projected deflated problem (sections 4.3 and 4.7), and implicit versus explicit deflation (section 4.5).

In section 5 we have collected some 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 computed in real arithmetic for real matrices. It is possible to derive a variant of Jacobi–Davidson based on this blocked form; we will not discuss this alternative variant here.

REMARK 2. With boldface 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 a 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 generalized eigenproblem.

2.1. Preliminaries.

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 [15], [21], [30, Ch.VI].

REMARK 3. 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 [15], 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, the choice of some parameters leads to an implicit scaling.

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

DEFINITION 1. *A* partial generalized Schur form *of dimension k for a matrix pair $(\mathbf{A}, \mathbf{B})$ is the decomposition*

$$\mathbf{A}\mathbf{Q}_k = \mathbf{Z}_k S_k, \quad \mathbf{B}\mathbf{Q}_k = \mathbf{Z}_k T_k, \tag{3}$$

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 (3) is equivalent to

$$\mathbf{Z}_k^* \mathbf{A} \mathbf{Q}_k = S_k \quad \text{and} \quad \mathbf{Z}_k^* \mathbf{B} \mathbf{Q}_k = T_k. \tag{4}$$

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 $(\mathbf{A}, \mathbf{B})$.

2.2. Jacobi–Davidson. We will briefly describe Jacobi–Davidson for the generalized eigenproblem (1); for details we refer to [24].

Similar to subspace approaches for standard eigenproblems, in each step the approximate eigenvector $\widetilde{\mathbf{q}}$ is selected from a *search subspace* span$\{\mathbf{V}\}$. The Galerkin condition, with associated approximate generalized eigenvalue $\langle \widetilde{\alpha}, \widetilde{\beta} \rangle$, requires orthogonality with respect to some *test subspace* span$\{\mathbf{W}\}$:

$$\widetilde{\beta}\, \mathbf{A}\, \widetilde{\mathbf{q}} - \widetilde{\alpha}\, \mathbf{B}\, \widetilde{\mathbf{q}} \perp \text{span}\{\mathbf{W}\}. \tag{5}$$

For the generalized case, it is, in view of (3) and (4), 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 (5) leads to the *projected eigenproblem*

$$(\widetilde{\beta}\, \mathbf{W}^* \mathbf{A} \mathbf{V} - \widetilde{\alpha}\, \mathbf{W}^* \mathbf{B} \mathbf{V})\, u = 0 \tag{6}$$

that can be solved by conventional techniques, and a solution $(u, \langle \widetilde{\alpha}, \widetilde{\beta} \rangle)$ is selected (note that (6) is a j-dimensional problem).

Then 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. In the variant of the Jacobi–Davidson method used in this paper, the search subspace is expanded by a vector $\mathbf{v}$ that is orthogonal to $\widetilde{\mathbf{q}}$ and that solves (approximately) the *Jacobi correction equation*

$$\left(\mathbf{I} - \frac{\widetilde{\mathbf{z}}\, \widetilde{\mathbf{z}}^*}{\widetilde{\mathbf{z}}^* \widetilde{\mathbf{z}}} \right) (\widetilde{\beta}\, \mathbf{A} - \widetilde{\alpha}\, \mathbf{B}) \left(\mathbf{I} - \frac{\widetilde{\mathbf{q}}\, \widetilde{\mathbf{q}}^*}{\widetilde{\mathbf{q}}^* \widetilde{\mathbf{q}}} \right) \mathbf{v} = -\mathbf{r}. \tag{7}$$

The essential difference with the Davidson approach is in the inclusion of the left and right orthogonal projections in this correction equation. It can be shown that for nonzero $\widetilde{\mathbf{z}} \in \text{span}\{\mathbf{A}\widetilde{\mathbf{q}}, \mathbf{B}\widetilde{\mathbf{q}}\}$, and if (7) is solved exactly, the convergence of the generalized eigenvalue will be quadratic; see [24, Th. 3.2].

In the next iteration of the algorithm, span$\{\mathbf{V}, \mathbf{v}\}$ defines the new search subspace, and we will explain how to expand $\mathbf{W}$ appropriately. Since we prefer orthogonal matrices $\mathbf{V}$ and $\mathbf{W}$, similar to $\mathbf{Z}$ and $\mathbf{Q}$ in (3), the new columns of $\mathbf{V}$ and $\mathbf{W}$ are orthonormalized by modified Gram–Schmidt (or some other stable variant of Gram–Schmidt).

We use the QZ algorithm [15] to reduce (6) 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

$$U_L{}^*(\mathbf{W}^* \mathbf{A} \mathbf{V}) U_R = S_A \quad \text{and} \quad U_L{}^*(\mathbf{W}^* \mathbf{B} \mathbf{V}) U_R = S_B. \tag{8}$$

The decomposition can be reordered 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 (6) [33], [34], [7]. For a MATLAB version of this reordering algorithm, see [7, Ch. 6.C].

With the decomposition in (8), we construct an approximate partial generalized Schur form (cf. (3)): $\mathbf{V}U_R$ approximates a $\mathbf{Q}_k$, and $\mathbf{W}U_L$ approximates the associated $\mathbf{Z}_k$. Since $\mathrm{span}\{\mathbf{Z}_k\} = \mathrm{span}\{\mathbf{A}\mathbf{Q}_k\} = \mathrm{span}\{\mathbf{B}\mathbf{Q}_k\}$ (cf. (3)), 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 $\mathrm{span}\{\mathbf{W}\}$ coincides with $\mathrm{span}\{\nu_0\mathbf{A}\mathbf{V} + \mu_0\mathbf{B}\mathbf{V}\}$. This choice is also in line with the restriction on $\widetilde{\mathbf{z}}$ for quadratic convergence.

In summary, the proposed method has the following main ingredients:

• Form the projected system (6) and reduce it to an ordered generalized Schur form (8). Select as approximate generalized eigenpair

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

• Form the *Jacobi correction* equation:

$$\widetilde{\mathbf{q}}^*\mathbf{v} = 0 \quad \text{and} \quad (\mathbf{I} - \widetilde{\mathbf{z}}\,\widetilde{\mathbf{z}}^*)(\widetilde{\beta}\,\mathbf{A} - \widetilde{\alpha}\,\mathbf{B})(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)\,\mathbf{v} = -\mathbf{r}, \tag{9}$$

with

$$\gamma\,\widetilde{\mathbf{z}} \equiv \nu_0\mathbf{A}\widetilde{\mathbf{q}} + \mu_0\mathbf{B}\widetilde{\mathbf{q}} \quad \text{and} \quad \mathbf{r} \equiv \widetilde{\beta}\,\mathbf{A}\widetilde{\mathbf{q}} - \widetilde{\alpha}\,\mathbf{B}\widetilde{\mathbf{q}}, \tag{10}$$

where γ is a normalization constant; the choice for ν_0 and μ_0 will be discussed later (in section 2.4). Compute an approximate solution $\mathbf{v} \perp \widetilde{\mathbf{q}}$ of (9). Note that $\widetilde{\mathbf{q}}$ $(= \mathbf{V}U_R(:,1))$ is normalized.

• Expand $\mathbf{V}$ with the orthonormal complement of $\mathbf{v}$ and $\mathbf{W}$ with the orthonormal complement of $\mathbf{w}$, where

$$\mathbf{w} \equiv \nu_0\,\mathbf{A}\mathbf{v} + \mu_0\,\mathbf{B}\mathbf{v}. \tag{11}$$

Modified Gram–Schmidt is used for the computation of the orthonormal complements.

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

$$\widetilde{\mathbf{z}} = \mathbf{W}U_L(:,1). \tag{12}$$

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 (6) 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 section 2.3.

• After convergence, expand the partial Schur form with the converged Schur vector, and repeat the algorithm with a deflated pencil for other eigenpairs. For details on deflation, see section 2.5.

2.3. Practical selection and implicit restart. When we have reduced the projected eigenproblem (6) to a generalized Schur form by the QZ algorithm [15], 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 $\widetilde{\mathbf{z}}$ (cf. (12)),
— restriction of the dimension of the subspaces $\mathrm{span}\{\mathbf{V}\}$ and $\mathrm{span}\{\mathbf{W}\}$ if necessary (restart),
— deflation after convergence of a Petrov pair.

We will explain the first and third points in more detail in this section.

Suppose that the generalized Schur form of the pencil

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

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

$$\left|\frac{S_A(1,1)}{S_B(1,1)} - \tau\right| \leq \left|\frac{S_A(2,2)}{S_B(2,2)} - \tau\right| \leq \cdots \leq \left|\frac{S_A(j,j)}{S_B(j,j)} - \tau\right|, \tag{13}$$

where j is the dimension of $\text{span}\{\mathbf{V}\}$. 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 (6) 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, 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

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

(cf. [26, section 5.3]).

REMARK 4. Our restart strategy follows similar ideas as in the implicitly restarted Arnoldi (IRA) [28]. However, in [28] 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 [28], [13].

2.4. The values of ν_0 and μ_0. We will now explain how to select the scalars ν_0 and μ_0 in (10). The restriction $|\nu_0|^2 + |\mu_0|^2 = 1$ is for scaling and avoids trivial expansions. We discuss two approaches. The first one, in section 2.4.1, can be viewed as a generalization of the approach by Ritz values for standard eigenproblems for optimal expansion of the test subspace. The second one, in section 2.4.2, is related to the approach by harmonic Ritz values and aims for optimal selection of Petrov pairs.

2.4.1. 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 with $\mathbf{q}$ 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.

The value of

$$\|\nu\mathbf{Aq} + \mu\mathbf{Bq}\|_2 \quad (= |\nu\alpha + \mu\beta|\,\|\mathbf{z}\|_2),$$

as function of ν and μ, is maximal if

$$\nu = \frac{\overline{\alpha}}{\sqrt{|\alpha|^2 + |\beta|^2}} \quad \text{and} \quad \mu = \frac{\overline{\beta}}{\sqrt{|\alpha|^2 + |\beta|^2}}, \tag{14}$$

where $\overline{\alpha}$ denotes the complex conjugate of α. This approach can be seen 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 $\mathbf{Aq}$ (or $\mathbf{Bq}$).

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

$$\nu_0 \equiv \frac{\overline{\tau}}{\sqrt{1 + |\tau|^2}} \quad \text{and} \quad \mu_0 \equiv \frac{1}{\sqrt{1 + |\tau|^2}}, \tag{15}$$

where τ is the target value.

For an approach by which the choice for the scalars is made adaptively, see [7, Ch. 6, section 3.1]. In practice, we have not seen much advantage of that approach compared with the approach to be discussed in section 2.4.2.

2.4.2. Values for ν_0 and μ_0 based on harmonic Petrov values. In this section, we will introduce harmonic Petrov values. We will see that the harmonic Petrov values that are closest to a target can be considered as extremal Ritz values for a specific test subspace, also if the target is in the interior of the spectrum. In particular for the computation of interior eigenvalues, the harmonic Petrov values appear to be attractive competitors for the standard Petrov values of the approaches in section 2.4.1: for generalized eigenproblems the costs for the computation of the harmonic Petrov values are the same as for standard 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. This is in line with observations for the standard eigenproblems made in [16], [26].

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.

However, 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. [26, section 5.1]): the resulting eigenvalue approximations $\widetilde{\lambda}$ for $\mathbf{A}$ are then the solutions of

$$\mathbf{W}^*\mathbf{AV}u - \widetilde{\lambda}\,\mathbf{W}^*\mathbf{V}u = 0, \quad \text{where } \mathbf{W} \equiv (\mathbf{A} - \tau\,\mathbf{I})\mathbf{V}. \tag{16}$$

The solutions $\widetilde{\lambda}$ are called *harmonic Ritz values* of $\mathbf{A}$, with respect to τ (cf. [19], [26]; they have also been used in [16]); $\mathbf{V}u$ 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 (11) with

$$\nu_0 \equiv \frac{1}{\sqrt{1 + |\tau|^2}} \quad \text{and} \quad \mu_0 \equiv -\frac{\tau}{\sqrt{1 + |\tau|^2}}. \tag{17}$$

For generalized problems, with ν_0 and μ_0 as in (17), the Petrov values closest to the target value correspond to absolute largest Ritz values of the standard eigenproblem with matrix $(\mathbf{A}-\tau\,\mathbf{B})^{-1}(\bar{\tau}\mathbf{A}+\mathbf{B})$. 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.

2.5. Expansion of Schur form and deflation. 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 the computation of a partial generalized Schur form as a major step in solving generalized eigenproblems.

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

$$\begin{aligned}\mathbf{A}\begin{bmatrix}\mathbf{Q}_{k-1} & \mathbf{q}\end{bmatrix} &= \begin{bmatrix}\mathbf{Z}_{k-1} & \mathbf{z}\end{bmatrix}\begin{bmatrix}S_{k-1} & s\\ 0 & \alpha\end{bmatrix} \quad \text{and}\\ \mathbf{B}\begin{bmatrix}\mathbf{Q}_{k-1} & \mathbf{q}\end{bmatrix} &= \begin{bmatrix}\mathbf{Z}_{k-1} & \mathbf{z}\end{bmatrix}\begin{bmatrix}T_{k-1} & t\\ 0 & \beta\end{bmatrix}.\end{aligned} \tag{18}$$

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

$$\mathbf{q} \perp \mathbf{Q}_{k-1} \quad \text{and} \quad (\beta\,\mathbf{A}-\alpha\,\mathbf{B})\,\mathbf{q} - \mathbf{Z}_{k-1}u = \mathbf{0} \tag{19}$$

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

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

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

$$\begin{aligned}&\mathbf{Q}_{k-1}^*\mathbf{q} = 0 \quad \text{and}\\ &(\mathbf{I}-\mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)(\beta\,\mathbf{A}-\alpha\,\mathbf{B})(\mathbf{I}-\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,\mathbf{q} = \mathbf{0},\end{aligned}^{2} \tag{20}$$

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

$$\begin{aligned}\big(&(\mathbf{I}-\mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)\mathbf{A}(\mathbf{I}-\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*),\\ &\qquad (\mathbf{I}-\mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)\mathbf{B}(\mathbf{I}-\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\big).\end{aligned} \tag{21}$$

In JDQZ we solve this eigenproblem again with the Jacobi–Davidson method.

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

$$\begin{aligned}&M_A \equiv \mathbf{W}^*(\mathbf{I}-\mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)\mathbf{A}(\mathbf{I}-\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,\mathbf{V} = \mathbf{W}^*\mathbf{A}\mathbf{V}\\ &\text{and}\\ &M_B \equiv \mathbf{W}^*(\mathbf{I}-\mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)\mathbf{B}(\mathbf{I}-\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,\mathbf{V} = \mathbf{W}^*\mathbf{B}\mathbf{V},\end{aligned} \tag{22}$$

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

$$U_L^*M_AU_R = S_A \quad \text{and} \quad U_L^*M_BU_R = S_B. \tag{23}$$

[2]Note that the Jacobi correction equation (7) has a similar structure and can be derived in a similar way.

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 (20). 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 Jacobi correction equation*

$$\begin{aligned} &\mathbf{Q}_{k-1}^* \mathbf{v} = 0, \quad \widetilde{\mathbf{q}}^* \mathbf{v} = 0, \quad \text{and} \\ &(\mathbf{I} - \widetilde{\mathbf{z}}\,\widetilde{\mathbf{z}}^*)(\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)(\widetilde{\beta}\,\mathbf{A} - \widetilde{\alpha}\,\mathbf{B}) \\ &\qquad\qquad \times (\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)\,\mathbf{v} = -\mathbf{r}, \end{aligned} \tag{24}$$

where $\mathbf{r} \equiv (\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)(\widetilde{\beta}\,\mathbf{A} - \widetilde{\alpha}\,\mathbf{A})(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,\widetilde{\mathbf{q}}$. Note that $\mathbf{Q}_{k-1}^*\,\widetilde{\mathbf{q}} = 0$. We also have to expand $\mathbf{W}$; we expand it with the complement of $(\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)(\nu_0\,\mathbf{A} + \mu_0\,\mathbf{B})(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\,\widetilde{\mathbf{q}}$, orthogonal with respect 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 $\operatorname{span}\{\mathbf{Q}_{k-1}, \widetilde{\mathbf{q}}\}$ and a new test subspace orthogonal to $\operatorname{span}\{\mathbf{Z}_{k-1}, \widetilde{\mathbf{z}}\}$, respectively, and we continue the process.

2.6. Solution of (deflated) correction equation. In this section we discuss how the generalized deflated Jacobi correction equation can be solved and how preconditioning is involved.

The correction equation (20) involves an operator for which the domain and the image space differ. This means that Krylov subspace methods cannot be applied right away. Fortunately, this can be fixed easily by incorporating preconditioning.

For preconditioning of the correction equation (20), we propose to use

$$(\mathbf{I} - \widetilde{\mathbf{z}}\,\widetilde{\mathbf{z}}^*)(\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Z}_{k-1}^*)\mathbf{K}(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*) \tag{25}$$

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

We introduce the following notation.

NOTATION 1.

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

In this notation the left preconditioned correction equation for the generalized correction equation (24) can be written as

$$\begin{aligned} &\widetilde{\mathbf{Q}}_k^*\mathbf{v} = 0 \quad \text{and} \\ &(\mathbf{I} - \widetilde{\mathbf{Y}}_k\,\widetilde{H}_k^{-1}\widetilde{\mathbf{Q}}_k^*)\,\mathbf{K}^{-1}(\widetilde{\beta}\,\mathbf{A} - \widetilde{\alpha}\,\mathbf{B})(\mathbf{I} - \widetilde{\mathbf{Y}}_k\,\widetilde{H}_k^{-1}\widetilde{\mathbf{Q}}_k^*)\,\mathbf{v} = -\widehat{\mathbf{r}}, \end{aligned} \tag{26}$$

where $\widehat{\mathbf{r}} \equiv (\mathbf{I} - \widetilde{\mathbf{Y}}_k\,\widetilde{H}_k^{-1}\widetilde{\mathbf{Q}}_k^*)\,\mathbf{K}^{-1}\mathbf{r}$ (in section 3.2, we give more details for the simpler standard eigenproblem).

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

$$\begin{aligned} &\widehat{\mathbf{Y}}_k^*\widehat{\mathbf{v}} = 0 \quad \text{and} \\ &(\mathbf{I} - \widetilde{\mathbf{Z}}_k\widetilde{H}_k^{-1}\widehat{\mathbf{Y}}_k^*)(\widetilde{\beta}\,\mathbf{A} - \widetilde{\alpha}\,\mathbf{B})\,\mathbf{K}^{-1}(\mathbf{I} - \widetilde{\mathbf{Z}}_k\widetilde{H}_k^{-1}\widehat{\mathbf{Y}}_k^*)\,\widehat{\mathbf{v}} = -\widehat{\mathbf{r}}, \end{aligned} \tag{27}$$

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 (26), the domain and the image space coincide, so that Krylov subspace methods can be used.

A pseudocode for the preconditioned Jacobi–Davidson QZ algorithm with harmonic Petrov values (as discussed in section 2.4.2) is given in Algorithm 1.

In Table 1 we have listed the main computational ingredients per iteration of JDQZ.

TABLE 1
The computational costs of JDQZ *per iteration. The integers j and k denote the dimensions of* span$\{\mathbf{V}\}$ *and* span$\{\mathbf{Q}\}$*, respectively.*

Part	DOTS	AXPYS	MVS	K
The correction equation	Variable costs			
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. Then no MVs are needed in this part.
[b] Instead of computing the residual $\mathbf{r}$ as $\widetilde{\beta}\,\mathbf{A}\,\widetilde{\mathbf{q}} - \widetilde{\alpha}\,\mathbf{B}\,\widetilde{\mathbf{q}}$, $\mathbf{r}$ 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. Algorithm 1); depending on the number of nonzeros in $\mathbf{A}$, $\mathbf{B}$, and the value of j, this may be more efficient.

3. Jacobi–Davidson for the standard eigenproblem. When $\mathbf{B} = \mathbf{I}$, the JDQZ algorithm simplifies greatly, and this can be used to improve computational efficiency. We will also consider this situation in order to discuss specific aspects such as preconditioning. It should be noted that these simplifications only lead to a memory efficient algorithm when we give up working with harmonic Ritz values since these involve skew instead of orthogonal projections in the standard setting [26]. For the computation of exterior eigenvalues this does not pose serious problems, but for interior eigenvalues we prefer the more robust harmonic eigenvalue approximations (section 2.4.2) and orthogonal projections, which means that we prefer JDQZ (with $\mathbf{B} = \mathbf{I}$) instead of the algorithm discussed in this section.

With the natural choice $\nu_0 = 0$, $\mu_0 = 1$, we have that $\mathbf{W} = \mathbf{V}$, so that the *projected eigenproblem* reduces to

$$(\mathbf{V}^*\mathbf{AV} - \widetilde{\lambda}\,\mathbf{V}^*\mathbf{V})\,u = 0; \tag{28}$$

that is, $\widetilde{\beta} = 1, \widetilde{\alpha} = \widetilde{\lambda}$ (cf. (6)).

For this low-dimensional problem we select a solution $(u, \widetilde{\lambda})$ by standard computational techniques. The *Ritz value* $\widetilde{\lambda}$ and the *Ritz vector* $\widetilde{\mathbf{q}} \equiv \mathbf{V}u$ form an approximate

ALGORITHM 1
Preconditioned JDQZ, *using harmonic Petrov values.*

function $[\,\mathbf{Q},\mathbf{Z},R_A,R_B] = \text{JDQZ}\,(\mathbf{A},\mathbf{B},\mathbf{K},\tau,\mathbf{v}_0,\epsilon,k_{\max},j_{\min},j_{\max})$
$\mathbf{Q} = [\,];\ \mathbf{Z} = [\,];\ R_A = [\,];\ R_B = [\,];\ \mathbf{Y} = [\,];\ H = [\,];$
$\mathbf{V} = [\,];\ \mathbf{V}_A = [\,];\ \mathbf{V}_B = [\,];\ \mathbf{W} = [\,];\ M_A = [\,];\ M_B = [\,];$
$\gamma = \sqrt{1+|\tau|^2};\ \nu_0 = 1/\gamma;\ \mu_0 = -\tau/\gamma;\ k = 0;\ j = 0;$
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* —
  $\mathbf{v} = \text{mgs}\,(\mathbf{V},\mathbf{v});\ \mathbf{v} = \mathbf{v}/\|\mathbf{v}\|_2;\ \mathbf{v}_A = \mathbf{A}\mathbf{v};\ \mathbf{v}_B = \mathbf{B}\mathbf{v};$
  $\mathbf{w} = (\nu_0\mathbf{v}_A + \mu_0\mathbf{v}_B);\ \mathbf{w} = \text{mgs}\,(\mathbf{Z},\mathbf{w});\ \mathbf{w} = \text{mgs}\,(\mathbf{W},\mathbf{w});\ \mathbf{w} = \mathbf{w}/\|\mathbf{w}\|_2;$
  $M_A = [\,M_A, \mathbf{W}^*\mathbf{v}_A; \mathbf{w}^*\mathbf{V}_A, \mathbf{w}^*\mathbf{v}_A\,];\ M_B = [\,M_B, \mathbf{W}^*\mathbf{v}_B; \mathbf{w}^*\mathbf{V}_B, \mathbf{w}^*\mathbf{v}_B\,];$
  $\mathbf{V} = [\,\mathbf{V},\mathbf{v}\,];\ \mathbf{V}_A = [\,\mathbf{V}_A,\mathbf{v}_A\,];\ \mathbf{V}_B = [\,\mathbf{V}_B,\mathbf{v}_B\,];\ \mathbf{W} = [\,\mathbf{W},\mathbf{w}\,];$
  $[U_L,U_R,S_A,S_B] = \text{qz}\,(M_A,M_B);$
  $[U_L,U_R,S_A,S_B] = \text{qzsort}\,(\tau,U_L,U_R,S_A,S_B);$
  $j = j+1;\ \text{found} = 1;$
  while found,
    — *harmonic Petrov approximation* —
    $\alpha = S_A(1,1);\ \beta = S_B(1,1);\ \mathbf{q} = \mathbf{V}U_R(:,1);\ \mathbf{z} = \mathbf{W}U_L(:,1);\ \mathbf{y} = \mathbf{K}^{-1}\mathbf{z};$
    $\mathbf{r} = \mathbf{A}\,\mathbf{q};\ [\,\mathbf{r},s_A] = \text{mgs}\,(\mathbf{Z},\mathbf{r});\ \mathbf{r}_B = \mathbf{B}\,\mathbf{q};\ [\,\mathbf{r}_B,s_B] = \text{mgs}\,(\mathbf{Z},\mathbf{r}_B);$
    $\mathbf{r} = \beta\,\mathbf{r} - \alpha\,\mathbf{r}_B;$
    $\widetilde{\mathbf{Q}} = [\,\mathbf{Q},\mathbf{q}\,];\ \widetilde{\mathbf{Y}} = [\,\mathbf{Y},\mathbf{y}\,];\ \widetilde{H} = [\,H, \mathbf{Q}^*\mathbf{y}; \mathbf{q}^*\mathbf{Y}, \mathbf{q}^*\mathbf{y}\,];$
      ⋮
    "found and implicit restart part", see Algorithm 2
      ⋮
  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 function mgs performs modified Gram–Schmidt and qzsort sorts the generalized Schur form. MATLAB implementations of mgs and qzsort can be found in [7, Ch. 6.A-C].

ALGORITHM 2
"Found and implicit restart part" of preconditioned JDQZ *with harmonic Petrov values.*

found = ($\|\mathbf{r}\|_2 < \epsilon$) & ($j > 1 \mid k = k_{\max} - 1$);
if found,
— *found* —
$\mathbf{Q} = \widetilde{\mathbf{Q}}$; $\mathbf{Z} = [\,\mathbf{Z}, \mathbf{z}\,]$;
$R_A = [\,R_A, s_A; \mathsf{zeros}\,(1,k), \alpha\,]$; $R_B = [\,R_B, s_B; \mathsf{zeros}\,(1,k), \beta\,]$;
$k = k + 1$; if $k = k_{\max}$, break; end
$\mathbf{Y} = \widetilde{\mathbf{Y}}$; $H = \widetilde{H}$;
$J = [\,2\!:\!j\,]$; $j = j - 1$;
$\mathbf{V} = \mathbf{V}U_R(:,J)$; $\mathbf{V}_A = \mathbf{V}_A\,U_R(:,J)$; $\mathbf{V}_B = \mathbf{V}_B\,U_R(:,J)$;
$\mathbf{W} = \mathbf{W}U_L(:,J)$; $S_A = S_A(J,J)$; $S_B = S_B(J,J)$;
$M_A = S_A$; $M_B = S_B$; $U_R = I$; $U_L = I$;
elseif $j == j_{\max}$,
— *implicit restart* —
$j = j_{\min}$; $J = [\,1\!:\!j\,]$;
$\mathbf{V} = \mathbf{V}U_R(:,J)$; $\mathbf{V}_A = \mathbf{V}_A\,U_R(:,J)$; $\mathbf{V}_B = \mathbf{V}_B\,U_R(:,J)$;
$\mathbf{W} = \mathbf{W}U_L(:,J)$; $S_A = S_A(J,J)$; $S_B = S_B(J,J)$;
$M_A = S_A$; $M_B = S_B$; $U_R = I$; $U_L = I$;
end

eigenvalue and eigenvector, respectively, with *residual* $\mathbf{r} \equiv (\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})\,\widetilde{\mathbf{q}}$ (we will assume that $\|\widetilde{\mathbf{q}}\|_2 = 1$).

For the expansion of $\mathbf{V}$, we take a vector $\mathbf{v} \perp \widetilde{\mathbf{q}}$ that solves (approximately) the *Jacobi correction equation*

$$\widetilde{\mathbf{q}}^*\mathbf{v} = 0 \quad \text{and} \quad (\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)\,\mathbf{v} = -\mathbf{r}. \tag{29}$$

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

As mentioned in the introduction, if $\widetilde{\lambda}$ is replaced in the correction equation (29) by an eigenvalue λ, then the associated eigenvector is contained in the space spanned by $\mathbf{V}$ and the exact solution of the Jacobi-correction equation. If the correction equation (29), with $\widetilde{\lambda}$, is solved exactly, then the speed of convergence for the selected Ritz values is asymptotically quadratical (cf. [26], [24]).

We reduce the projected eigenproblem (28) to Schur form by the QR algorithm [8], and then we can exploit the Schur form for the selection of a Ritz pair $(\widetilde{\mathbf{q}}, \widetilde{\lambda})$ and for restriction of the dimension of the subspace span$\{\mathbf{V}\}$ in the same way as explained in section 2.3 (note that in this case $S_B = I$).

A reordering algorithm for the Schur form can be found, for instance, in [29], [9], [20]; a Fortran implementation is available from LAPACK [1]. A simple MATLAB implementation for reordering with respect to a target value τ is given in [7, Ch. 6.B].

3.1. JDQR. For the standard eigenproblem we can use the JD algorithm for the computation of a *partial Schur form*, which can be written as (cf. [22]):

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

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 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 $\mathbf{A}$.

The resulting simplification of JDQZ will be referred to as JDQR. Although most of the simplifications are obvious, we will give the main expressions for JDQR for ease of reference.

Suppose that $k-1$ Schur pairs have detected; i.e., we already have the partial Schur form $\mathbf{A}\mathbf{Q}_{k-1} = \mathbf{Q}_{k-1}R_{k-1}$. Then the new Schur pair $(\mathbf{q}, \lambda)$ is an eigenpair of the *deflated matrix*

$$(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\mathbf{A}(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*). \tag{30}$$

We then solve the eigenproblem for the deflated matrix (30). More precisely, the JD algorithm for the deflated matrix (30) 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

$$M \equiv \mathbf{V}^*(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\mathbf{A}(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\mathbf{V} = \mathbf{V}^*\mathbf{A}\mathbf{V}. \tag{31}$$

The ordered Schur form $MU = US$ gives an approximation $(\widetilde{\mathbf{q}}, \widetilde{\lambda}) \equiv (\mathbf{V}U(:,1), S(1,1))$ for a wanted eigenpair of the deflated matrix (30). Then, according to the Jacobi–Davidson approach, the search subspace span$\{\mathbf{V}\}$ is expanded by the orthonormal complement of $\mathbf{v}$ to $\mathbf{V}$, where $\mathbf{v}$ is the (approximate) solution of the *deflated Jacobi correction equation*

$$\begin{aligned} &\mathbf{Q}_{k-1}^*\mathbf{v} = 0, \quad \widetilde{\mathbf{q}}^*\mathbf{v} = 0, \quad \text{and} \\ &(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I}) \\ &\qquad\qquad \times (\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)\,\mathbf{v} = -\mathbf{r}, \end{aligned} \tag{32}$$

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

Note that the projections in (32) 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 $\widetilde{\mathbf{q}}^*\mathbf{r} = 0$.

REMARK 5. Two deflation techniques can be found in literature for subspace methods like Arnoldi's method. They are referred to as explicit and implicit deflation (cf., e.g., [22, Ch. VI, section 2.3]). In explicit deflation, the computation is continued with a deflated matrix after detection of Schur vectors. For efficiency reasons, $\mathbf{A} - \mathbf{Q}\,R\,\mathbf{Q}^*$ is used (Schur–Wielandt deflation), rather than the 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 Jacobi correction equation we use the explicitly deflated matrix. 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 computing the deflated interaction matrix M; we compute M as $M = \mathbf{V}^*\mathbf{A}\mathbf{V}$ (cf. (31)).

Similar observations hold for the interaction matrices M_A and M_B in the generalized case (cf. (22)).

Exclusively implicit deflation is possible as well: solve the correction equation approximately with the nondeflated $\mathbf{A}$ and make 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 that leads to a faster converging process for the Jacobi correction equation (29). The decrease in 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 section 4.5).

Moreover, the explicitly deflated correction equation (32) appears 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 not only depends on the norm of the residual and on the condition number of the pair but also on the approximation errors in the previously detected Schur pairs (cf., e.g., [22, Ch. IV, section 2.5] and [13, section 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$\{\mathbf{Q}\}$.

3.2. Preconditioning. In this section we will discuss preconditioning for the Jacobi correction equation. Preconditioning is not straightforward 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 Jacobi correction equation (32) for a given $\widetilde{\mathbf{q}}$ and $\widetilde{\lambda}$ (cf. (32)). For the approximate solution of this equation we may use a Krylov subspace method, e.g., GMRES [23], or BiCGstab(ℓ) [25]. 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 [14], [10] for $\mathbf{A} - \widetilde{\lambda}\mathbf{I}$ may require much fill-in,[3] which makes the construction expensive. As we will argue in section 3.3, it may be a good strategy to compute a good (and possibly expensive) preconditioner $\mathbf{K}$ for $\mathbf{A} - \tau\mathbf{I}$ for one fixed value of τ only and to use

$$\widetilde{\mathbf{K}} \equiv (\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*)(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)\mathbf{K}(\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^*)(\mathbf{I} - \widetilde{\mathbf{q}}\,\widetilde{\mathbf{q}}^*) \tag{33}$$

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. [24]).

We will now give some more details on the derivation of the expressions for the preconditioned correction equation. We use the same notation as in Notation 1 with $\widetilde{\mathbf{Z}}_k = \widetilde{\mathbf{Q}}_k$. The typical usage of the preconditioner in a Krylov subspace method would look like

$$\begin{aligned} &\textit{solve } \mathbf{t}, \textit{ with } \widetilde{\mathbf{Q}}_k^*\mathbf{t} = 0, \textit{ from} \\ &(\mathbf{I} - \widetilde{\mathbf{Q}}_k\widetilde{\mathbf{Q}}_k^*)\,\mathbf{K}\,(\mathbf{I} - \widetilde{\mathbf{Q}}_k\widetilde{\mathbf{Q}}_k^*)\,\mathbf{t} = \mathbf{s} \\ &\textit{for } \mathbf{s}, \textit{ with } \widetilde{\mathbf{Q}}_k^*\mathbf{s} = 0. \end{aligned} \tag{34}$$

[3] 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.

The following lemma gives us an explicit expression for the solution $\mathbf{t}$ of (34) in terms of easily computable matrix-vector products with $\widetilde{H}_k^{-1}$ and $\mathbf{K}^{-1}$. The lemma generalizes Proposition 7.5 in [24], and the proof runs along the same lines (for details, see [7, Ch. 6, section 2.4]). Note that $\widetilde{H}_k$ is of small dimension, so it is cheaply inverted. There is no need to invert $\mathbf{K}$ explicitly; instead $\mathbf{v} = \mathbf{K}^{-1}\mathbf{s}$ can be computed by solving $\mathbf{v}$ from $\mathbf{K}\mathbf{v} = \mathbf{s}$.

LEMMA 1. *If $\widetilde{H}_k$ is nonsingular, then the solution $\mathbf{t}$ of equation* (34) *is given by*

$$\mathbf{t} = (\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\, \mathbf{K}^{-1}\mathbf{s}. \tag{35}$$

REMARK 7. 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 8. 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 1, it follows that the left preconditioned correction equation is equivalent with

$$\begin{aligned} &\widetilde{\mathbf{Q}}_k^*\mathbf{v} = 0 \quad \text{and} \\ &(\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\, \mathbf{K}^{-1}(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})(\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\, \mathbf{v} = -\widehat{\mathbf{r}}, \end{aligned} \tag{36}$$

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. [24, 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 (36) is equivalent to the one in (32).

Of course, right preconditioned correction equations, similar to (27), can be derived in a corresponding manner (for details, see [7, Ch. 6, section 2.4]).

REMARK 9. If one uses Krylov subspace methods for solving the second equation in (36), then one encounters matrix-vector products of the form

$$(\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\mathbf{K}^{-1}(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})(\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\, \mathbf{t}, \tag{37}$$

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 $\mathbf{v}$ we have that $\widetilde{\mathbf{Q}}_k^*\mathbf{v} = 0$, provided that this is the case for the initial guess as well. Moreover, the projection $(\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)$ in front of $\mathbf{t}$ in (37) is then redundant, and (37) reduces to

$$(\mathbf{I} - \widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \widetilde{\mathbf{Q}}_k^*)\mathbf{K}^{-1}(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})\, \mathbf{t}. \tag{38}$$

ALGORITHM 3
Preconditioned JDQR.

```
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 = A v;
    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 q − λ 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 Algorithm 4
          ⋮
    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 function mgs performs modified Gram–Schmidt, while qrsort sorts the Schur form. For MATLAB implementations of mgs and qrsort, see [7, Ch. 6.A-B].

A pseudocode for the preconditioned Jacobi–Davidson QR algorithm is given in Algorithm 3.

In Table 2 we have listed the main computational ingredients per iteration of JDQR.

ALGORITHM 4
"Found and implicit restart part" of JDQR.

found = $(\|\mathbf{r}\|_2 < \epsilon)$ & $(j > 1 \mid k = k_{\max} - 1)$;
if found,
— *found* —
$\mathbf{Q} = \widetilde{\mathbf{Q}}$; $R = [\, R, s; \mathsf{zeros}\,(1,k), \lambda\,]$;
$k = k + 1$; if $k == k_{\max}$, break; end
$\mathbf{Y} = \widetilde{\mathbf{Y}}$; $H = \widetilde{H}$;
$J = [\, 2\!:\!j\,]$; $j = j - 1$;
$\mathbf{V} = \mathbf{V}U(:, J)$; $\mathbf{V}_A = \mathbf{V}_A\, U(:, J)$;
$S = S(J, J)$; $M = S$; $U = I$;
elseif $j == j_{\max}$,
— *implicit restart* —
$j = j_{\min}$; $J = [\, 1\!:\!j\,]$;
$\mathbf{V} = \mathbf{V}U(:, J)$; $\mathbf{V}_A = \mathbf{V}_A\, U(:, J)$;
$S = S(J, J)$; $M = S$; $U = I$;
end

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

Part	DOTS	AXPYS	MVs	K
The correction equation	Variable costs			
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	$2j_{\min}$	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 $\mathbf{A}\,\widetilde{\mathbf{q}} - \widetilde{\lambda}\,\widetilde{\mathbf{q}}$, $\mathbf{r}$ may also be computed as $\mathbf{V}_A U(:,1) - \widetilde{\lambda}\,\widetilde{\mathbf{q}}$, where $\mathbf{V}_A \equiv \mathbf{AV}$ (cf. Algorithm 3); depending on the number of nonzeros in $\mathbf{A}$ and the value j, this may be more efficient.

3.3. 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, especially 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

$$\begin{aligned}(\mathbf{I} - \mathbf{Q}_k\mathbf{Q}_k^*)(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})(\mathbf{I} - \mathbf{Q}_k\mathbf{Q}_k^*) \\ = (\mathbf{I} - \mathbf{Q}_k\mathbf{Q}_k^*)\,\mathbf{K}\,(\mathbf{I} - \mathbf{Q}_k\mathbf{Q}_k^*) \\ - (\mathbf{I} - \mathbf{Q}_k\mathbf{Q}_k^*)\,\mathbf{R}\,(\mathbf{I} - \mathbf{Q}_k\mathbf{Q}_k^*) - (\tau - \widetilde{\lambda})(\mathbf{I} - \mathbf{Q}_k\mathbf{Q}_k^*),\end{aligned} \tag{39}$$

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. [32, 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 for JDQR in some of our experiments.

3.4. Notes on the speed of convergence. In this section we will make some comments with respect to the convergence behavior of JDQR. We will use well-known arguments.

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 [22] process, with a shift $\widetilde{\lambda}$ for a (deflated) eigenproblem, as can be understood as follows.

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 (29) is given by

$$\mathbf{v} = -\widetilde{\mathbf{q}} + (\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})^{-1}\widetilde{\mathbf{q}}\epsilon \tag{40}$$

with $\epsilon = 1/(\widetilde{\mathbf{q}}^*(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})^{-1}\widetilde{\mathbf{q}})$ (cf. [26, section 4.1]). Writing $\widetilde{\mathbf{q}}$ as $\sum \gamma_i\mathbf{x}_i$, it follows that

$$(\mathbf{A} - \widetilde{\lambda}\,\mathbf{I})^{-1}\widetilde{\mathbf{q}} = \sum_i \frac{\gamma_i}{\lambda_i - \widetilde{\lambda}}\,\mathbf{x}_i. \tag{41}$$

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,

$$\mathbf{v} \sim \sum_{i \neq 1} \frac{\gamma_i}{\lambda_i - \widetilde{\lambda}}\,\mathbf{x}_i. \tag{42}$$

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 $\widetilde{\lambda}$.

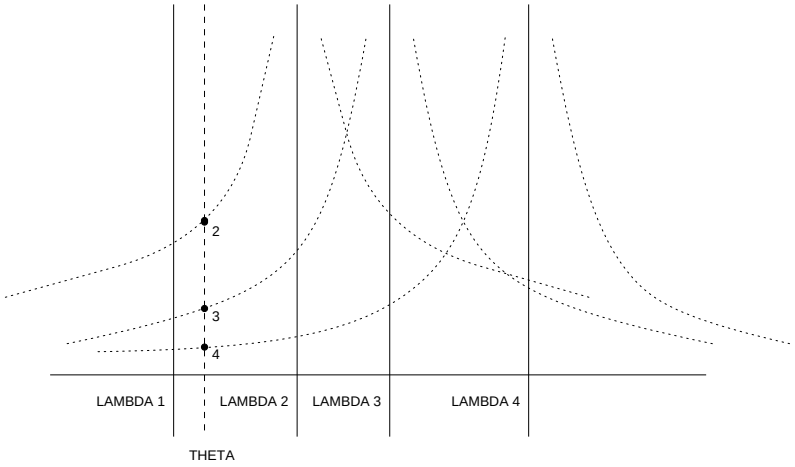

FIG. 1. *Amplification factors of eigenvectors.*

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 section 4.

4. Numerical experiments. In this section we present numerical results, obtained with JDQZ and JDQR, for several generalized eigenproblems and standard eigenproblems. The purpose of these experiments is to get an 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 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 (32), involving detected Schur vectors, are essential components of the algorithms. We will also consider eigenproblems where multiple eigenvalues are involved.

The computations were done in double complex precision (≈ 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. Algorithms 3 and 1).

As iterative solvers for the correction equation, we have considered full GMRES [23] with a maximum of m steps, denoted by GMRES$_m$, and BiCGstab(2) [25]. For BiCGstab(2) a maximum of 100 matrix multiplications was allowed. Of course, this is comparing apples with pears; our main purpose was to mimic two realis-

tic scenarios, one with a fixed number of matrix-vector multiplications (GMRES$_m$), and one with an iterative method to satisfy some stopping criterion (BiCGstab(2)). As stopping criterion for the iterative methods for the correction equation, also for GMRES$_m$, 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 [24], [27], and for Newton processes this stopping criterion may lead to efficient algorithms [6]. As the initial guess for the inner iteration method we always took the null vector.

In the figures of the convergence behavior for JDQZ and JDQR, 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 JDQZ and JDQR 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 JDQZ and JDQR.

Furthermore, we have plotted the entire convergence behavior. This means that the convergence history of the residuals of all subsequentially 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 acceptance marks the start of a new search.

4.0.1. 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 (26) and (36) 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. section 3). 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}. A similar switch was proposed in [18].

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 to those in the preceding paragraph (cf. [24, section 9.4]).

4.0.2. Stopping criterion. In our experiments, we considered an approximate eigenpair $(\widetilde{\mathbf{q}},\widetilde{\lambda})$ converged if the residual $\mathbf{r}$ is sufficiently small ($\|\mathbf{r}\| \le \epsilon$); then $(\widetilde{\mathbf{q}},\widetilde{\lambda})$

TABLE 3
Four eigenvalues of DW1024, *computed by* JDQR *(cf. section* 4.1*).*

$9.6473e-01$
$9.6551e-01$
$9.7780e-01$
$9.7880e-01$

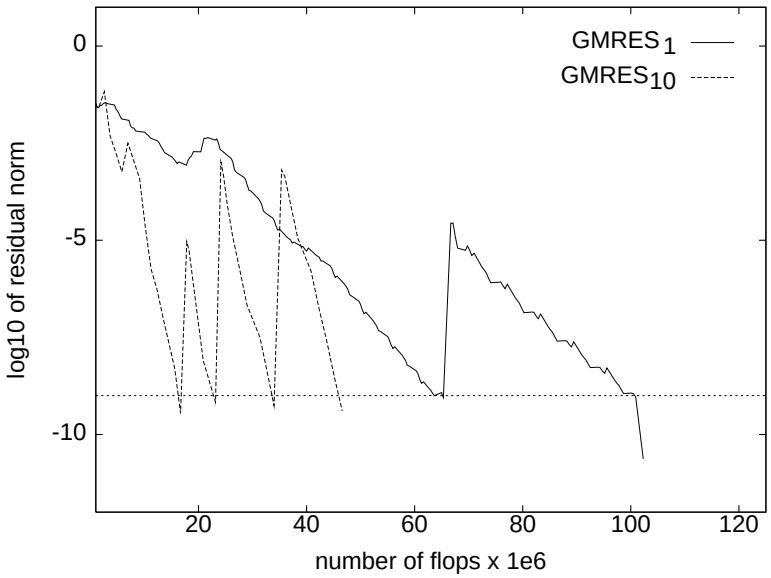

FIG. 2. *Convergence history for* DW1024, *showing the effect of solving the correction equations more accurately (cf. section* 4.1*).*

is a "detected" eigenpair. In the algorithms, weighted residuals (e.g., $\|\mathbf{r}\| \leq \epsilon\|\mathbf{A}\|$) or more sophisticated stopping criteria can be employed as well (cf. [4]).

In our experiments we varied values for some of the parameters in JDQZ and JDQR. 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
$\epsilon_{\rm tr}$	threshold used for building initial subspaces

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 DW1024 of order 2048 [2]. The problem comes from an integrated circuit application. The rightmost eigenvalues and their eigenvectors are wanted. We have used JDQR with standard Ritz values.

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

The computed eigenvalues are given in Table 3. The convergence history is plotted in Fig. 2 for JDQR, 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 Table 4.

TABLE 4
Summary of results for DW1024 *(cf. section* 4.1*).*

Method for the correction equation	JDQR iterations	MVs	flops $\times 10^6$
GMRES_1	185	185	$1.023e+02$
GMRES_{10}	43	296	$4.661e+01$

TABLE 5
Five eigenvalues of BWM2000, *computed by* JDQR *(cf. section* 4.2*).*

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

When solving the correction equation more accurately, the number of MVs is increased, but the number of outer iterations is reduced significantly (see Table 4), 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 IRA [28]. 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 phase with two small bumps, 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. section 3.4).

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. With appropriate preconditioning, we may see a dramatic improvement, as we will see in this example.

We consider the standard eigenproblem BWM2000 of order 2000 for the Brusselator wave model [2], [22]. 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 the largest real part in order to verify whether their real parts are positive or negative (corresponding to stable or unstable modes). Again we have used JDQR with standard Ritz values.

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

The computed eigenvalues are listed in Table 5. 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 Table 6.

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 expen-

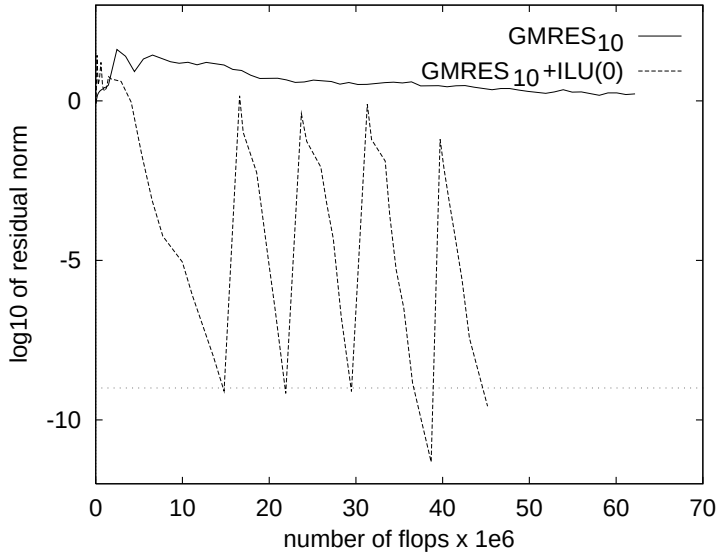

FIG. 3. *Convergence history for* BWM2000, *illustrating the effect of including preconditioning in the solver of the correction equation (cf. section* 4.2*).*

TABLE 6
Summary of results for BWM2000 *(cf. section* 4.2*).*

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

sive after each detected eigenvalue, the computational work for each eigenvalue is roughly constant, except for the first eigenvalue.

It should be noted that popular software packages for solving eigenproblems have been reported to fail for this problem BWM2000 [12].

The preconditioned correction equation (26) requires the computation and storage of the term $\widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1}$ (see Notation 1, with $\widetilde{\mathbf{Z}}_k = \widetilde{\mathbf{Q}}_k$). If $\mathbf{K}$ is a good approximation for $\mathbf{A} - \widetilde{\lambda}\mathbf{I}$, then $\widetilde{\mathbf{Y}}_k \approx (\mathbf{A} - \widetilde{\lambda}\mathbf{I})^{-1}\widetilde{\mathbf{Q}}_k$, and $\widetilde{H}_k \approx \widetilde{\mathbf{Q}}_k^*(\mathbf{A} - \widetilde{\lambda}\mathbf{I})^{-1}\widetilde{\mathbf{Q}}_k$. If $\widetilde{\mathbf{Q}}_k$ spans an invariant subspace of $\mathbf{A}$, then, in this nearly ideal case, we have that $\widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1} \approx \widetilde{\mathbf{Q}}_k$, and for this reason it is tempting to use (26) with $\widetilde{\mathbf{Q}}_k$ instead of $\widetilde{\mathbf{Y}}_k \widetilde{H}_k^{-1}$. However, this approach does not work well: it may even lead to slower convergence. In some of our experiments (as for instance with QH882 as discussed below in section 4.3) we lost convergence completely; in other experiments it had little or no effect.

4.3. Harmonic Ritz values. The JDQR algorithm computes a partial Schur form for the standard eigenproblem with standard Ritz pairs for the Schur pairs. However, with JDQZ for $\mathbf{B} = \mathbf{I}$, we can also compute a partial Schur form for the standard eigenproblem with harmonic Ritz pairs. Here we give an example that illustrates the improved convergence behavior with harmonic Ritz 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.

TABLE 7
Five eigenvalues of QH882, *computed by* JDQR *(cf. section* 4.3*).*

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

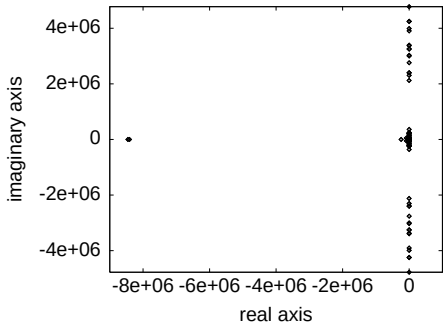

FIG. 4. *Spectrum of* QH882 *(cf. section* 4.3*).*

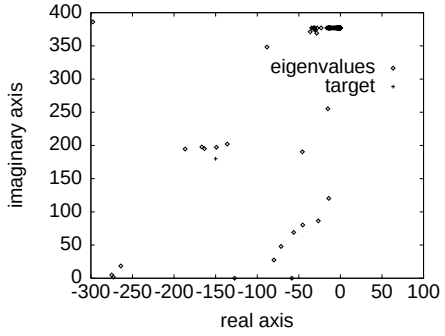

FIG. 5. *Part of the spectrum of* QH882.

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 Table 7. The convergence history is plotted in Fig. 7 for JDQR with standard Ritz values and for JDQZ with the harmonic choice (cf. section 2.4.2).

This problem is rather difficult: the eigenvalues in the neighborhood of τ are in the interior of the spectrum; see Figs. 4 and 5. For all three methods, the correction equation was solved with GMRES_{20} and was preconditioned with the exact inverse of $\mathbf{A} - \tau\,\mathbf{I}$. A summary of the results is given in Table 7.

Although the computational complexity of JDQR is less than the computational complexity of JDQZ (cf. Tables 1 and 2), it is not the most efficient method here. From the irregular convergence behavior of JDQR in Fig. 6 we may conclude that JDQR has problems in selecting the "correct" Ritz pairs and as a result the convergence is delayed. 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. As anticipated (cf. section 2.2), 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.

4.4. Identification of suitable Ritz values (tracking). From a computational point of view JDQR is less complex than JDQZ, even if the savings that standard eigenproblems problem allow are exploited in JDQZ. Therefore, it may be attractive to identify efficient strategies for avoiding "incorrect" selection of Ritz

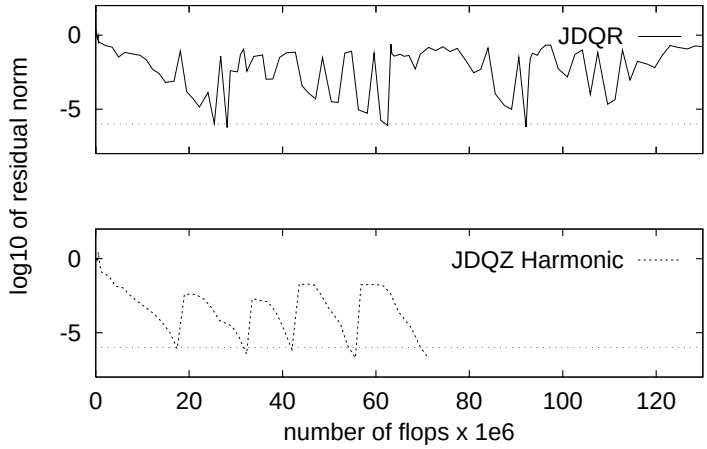

FIG. 6. *Convergence history for* QH882. *Although* QH882 *is a standard eigenproblem, for computing interior eigenvalues it is more efficient to use* JDQZ *with test subspaces that are different from the search subspaces (bottom picture). A better selection with harmonic Ritz values (bottom picture) appears to compensate for a less optimal expansion of the test subspace (cf. section* 4.3*).*

pairs. With correctly selected Ritz pairs convergence may be expected to be less irregular and stagnation may be completely avoided. We will discuss such a strategy now.

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 be 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 the experiment that we will discuss below, 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. In the example below, we took this threshold equal to $\epsilon_{\rm tr}$, the value used in the criterion for leaving the initialization stage (see section 4.0.1). 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, as we will see below.

Because of the rather regular convergence of Ritz values to exterior eigenvalues, improvement for these eigenvalues may not be expected with the above tracking strategy for standard problems.

Here we illustrate (in Fig. 7) the effects one may see by including tracking for JDQR. We applied JDQR to the example of section 4.3 with the same choice of parameters.

Nevertheless, we see that the convergence behavior of JDQZ with the harmonic choice of test subspace is still superior. See also Table 8.

4.5. 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. (24)) are very

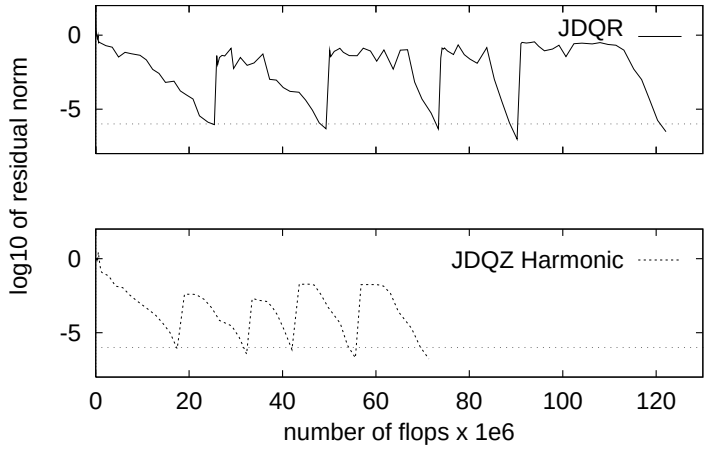

FIG. 7. *Convergence history for* QH882 *obtained with the tracking strategy (for all variants). For interior eigenvalues, the tracking strategy improves* JDQR *significantly (compare the present upper picture with the upper picture in Fig.* 7*), while there is no improvement for* JDQZ *(compare the two bottom pictures of the present figure and Fig.* 7*) (cf. section* 4.4*).*

TABLE 8
Summary of results for QH882 *(cf. sections* 4.3 *and* 4.4*).*

Method	Iterations	MVs	flops $\times 10^6$
JDQR	*	*	*
JDQR with tracking (cf. section 4.4)	99	1482	$1.221e+02$
JDQZ Harmonic	57	665	$7.133e+01$

essential in the correction equation (cf. section 3.4), and we show what happens when these projections are neglected. Note that we still take the Jacobi projections (with $\widetilde{\mathbf{q}}$ and $\widetilde{\mathbf{z}}$) into account.

We consider the bounded fineline dielectric waveguide generalized eigenproblem BFW782 [2] of order 782. This problem stems from a finite element discretization of the Maxwell equation for propagating modes and magnetic field profiles of a rectangular waveguide filled with dielectric and PEC structures. The resulting matrix $\mathbf{A}$ 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. 8. A magnification of the region of interest is plotted in Fig. 9. The computed generalized eigenvalues, represented as α/β, are given in Table 9. With JDQZ we discovered all four positive generalized eigenvalues.

The convergence history, for the harmonic version of JDQZ with GMRES_{10}, is plotted in the upper picture of Fig. 10. A summary of the results is given in Table 10. We see that JDQZ converges quite nicely.

In the bottom picture of Fig. 10 the convergence behavior of JDQZ is given for the case where the correction equation (24) is solved without taking into account the projections involving $\mathbf{Q}_k$ and $\mathbf{Z}_k$. Of course, the correction equations that are used

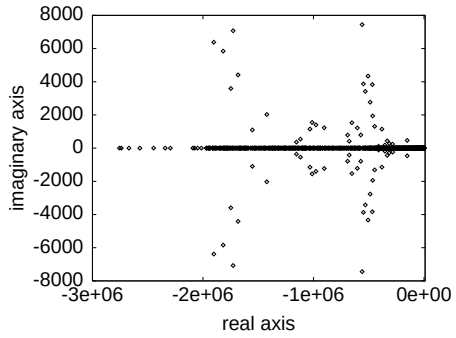

FIG. 8. *Spectrum of* BFW782 *(cf. section* 4.5*).*

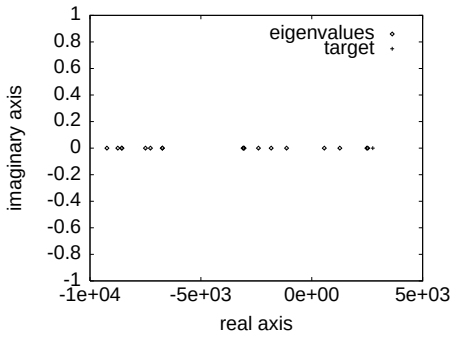

FIG. 9. *Part of the spectrum of* BFW782.

TABLE 9
Five generalized eigenvalues of BFW782, *computed by* JDQZ *(cf. section* 4.5*).*

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

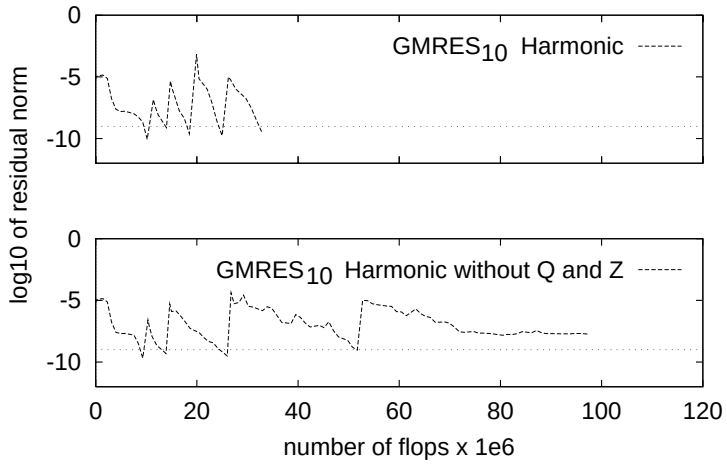

FIG. 10. *Convergence history for* BFW782 *with (upper picture) and without (bottom picture) deflating the matrices in the correction equations, with respect to the detected Schur vectors. (cf. section* 4.5*).*

TABLE 10
Summary of results for BFW782 *(cf. section* 4.5*).*

Method for the correction equation	JDQZ iterations	MVs	flops $\times 10^6$
$GMRES_{10}$	37	233	$3.17e+01$

TABLE 11
15 *generalized eigenvalues of* AC1331, *computed by* JDQZ *(cf. section* 4.6*).*

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

include the rank-one projections involving $\widetilde{\mathbf{q}}$ and $\widetilde{\mathbf{z}}$: 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 is improved significantly: in the results shown here (cf. Fig. 10) modified Gram–Schmidt is applied twice.

However, as explained in section 3.4, we do not benefit from an improved operator in the inner iteration. Although the resulting algorithm is computationally cheaper, Fig. 10 shows that this does not lead to an overall better performance: the speed of convergence becomes increasingly slower and even stagnates eventually.

4.6. 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 $\epsilon_{\text{tr}} = 10^{-4}$.

The computed 15 leftmost generalized eigenvalues represented as α/β are given in Table 11. The residual norm versus the number of flops is plotted in Fig. 11 for the harmonic version of JDQZ with GMRES_{10} and with BiCGstab(2), respectively. A summary of the results is given in Table 12.

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 (two or three peaks in the plot with GMRES and a plateau in the plot with BiCGstab(2)), after which the eigenvalues are discovered quickly one after another. This behavior is in agreement with section 3.4: 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-one 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 [31]). (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.

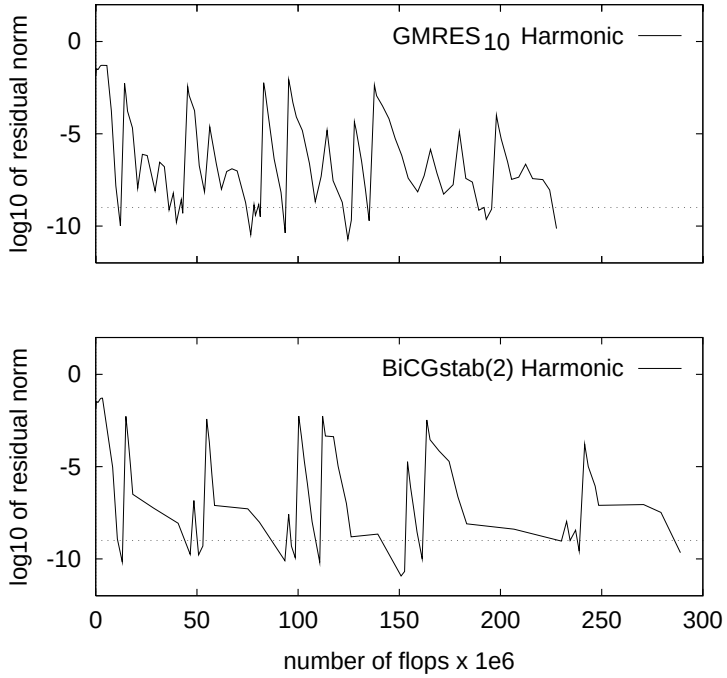

FIG. 11. *Convergence history for* AC1331: *Stagnation followed by fast detection of triple generalized eigenvalues (cf. section* 4.6*).*

TABLE 12
Summary of results for AC1331 *(cf. section* 4.6*).*

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

4.7. Harmonic Petrov values for generalized problems. Our last example shows again that for interior generalized eigenvalues the harmonic version JDQZ is quite powerful.

We consider the MHD416 generalized eigenproblem of order 416 [2], [24], [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 Figs. 12 and 13.

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. 14. The convergence history for the harmonic version of JDQZ, with GMRES_1, are plotted in Fig. 15. The exact inverse of $\mathbf{A} - \tau\,\mathbf{B}$ (for τ fixed) was used as a preconditioner for all eigenvalues.

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 the next Schur pairs as well.

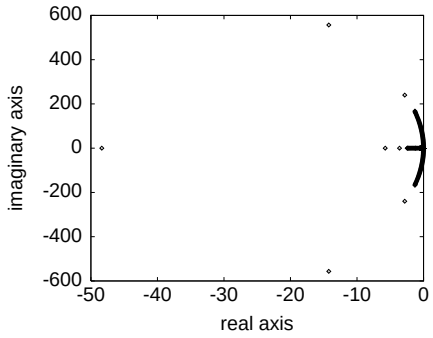

FIG. 12. *Spectrum of* MHD416 *(cf. section* 4.7*).*

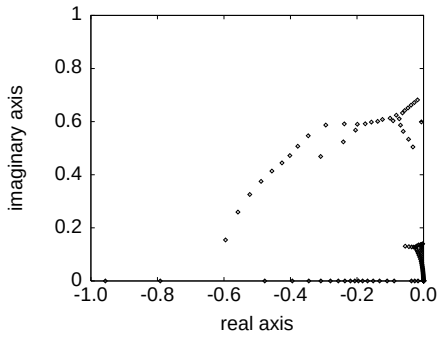

FIG. 13. *Alfvén branch of* MHD416 *(cf. section* 4.7*).*

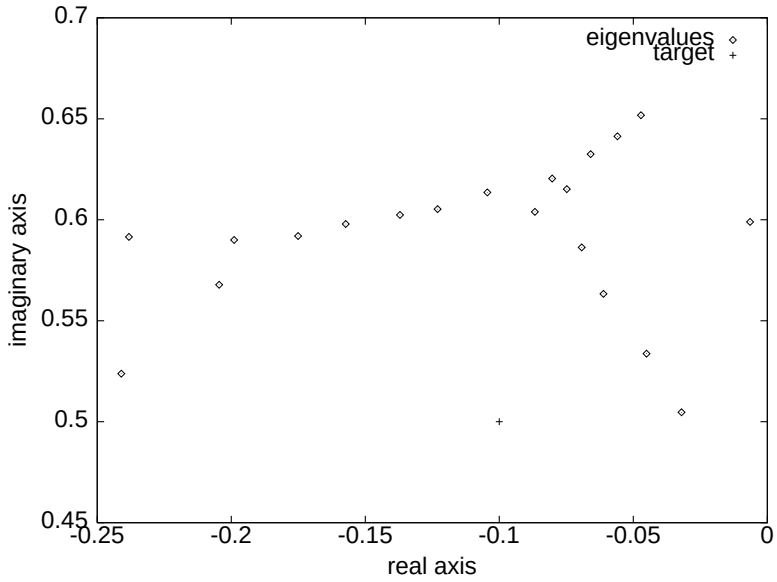

FIG. 14. 20 *generalized eigenvalues computed by* JDQZ *for* MHD416 *(cf. section* 4.7*).*

5. Conclusions. We have proposed two algorithms, JDQZ and JDQR, for computing several selected eigenpair approximations for generalized and standard 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 JDQZ and JDQR 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.

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.

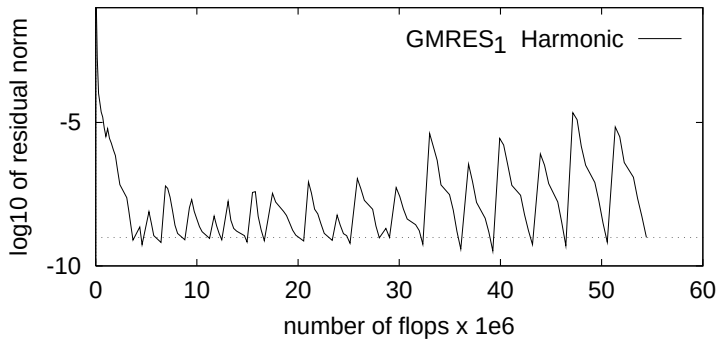

FIG. 15. *Convergence history of* JDQZ *for* MHD416. *Harmonic Petrov values allow a good selection of Petrov pairs for computing interior generalized eigenvalues also for generalized problems (cf. section* 4.7*).*

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 standard Petrov value choice and the harmonic Petrov value choice. It is argued and confirmed by our experiments that especially for interior eigenvalues the harmonic approach is also superior for generalized eigenproblems.

Acknowledgments. We acknowledge helpful discussions with Rich Lehoucq and with Beresford Parlett (on naming conventions). We also appreciate the helpful comments by the referees.

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, D. DAY, J. DEMMEL, AND J. DONGARRA, *A Test Matrix Collection for Non-Hermitian Eigenvalue Problems*, Technical Report CS-97-355, Department of Computer Science, University of Tennessee, Knoxville, TN, March, 1997.

[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 Proceeeding of ICIAM/GAMM 95, Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM), 76 (1996), pp. 131–134.

[4] F. CHAITIN-CHATELIN AND V. FRAYSSÉ, *Lectures on Finite Precision Computations*, SIAM, Philadelphia, 1996.

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

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

[7] D. R. FOKKEMA, *Subspace methods for linear, non-linear and eigen problems*, Ph.D. thesis, Utrecht University, Utrecht, The Netherlands, 1996.

[8] 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.

[9] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, London, 1989.

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

[11] C. JACOBI, *Ueber ein leichtes Verfahren, die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen*, J. Reine Angew. Math., (1846), pp. 51-94.

[12] R. B. Lehoucq and J. A. Scott, *An Evaluation of Software for Computing Eigenvalues of Sparse Nonsymmetric Matrices*, Preprint MCS-P547-1195, Argonne National Laboratory, Argonne, IL, 1996.

[13] R. B. Lehoucq and D. C. Sorensen, *Deflation techniques within an implicitly restarted iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.

[14] 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.

[15] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.

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

[17] R. B. Morgan *On restarting the Arnoldi method for large nonsymmetric eigenvalue problems*, Math. Comp., 65 (1996), pp. 1213–1230.

[18] R. B. Morgan and D. S. Scott, *Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices*, SIAM J. Sci. Statist. Comput., 7 (1989), pp. 817–825.

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

[20] A. Ruhe, *Perturbation bounds for means of eigenvalues and invariant subspaces*, BIT, 10 (1970), pp. 343–354.

[21] A. Ruhe, *Rational Krylov algorithms for nonsymmetric eigenvalue problems* II. *Matrix Pairs*, Linear Algebra Appl., 197/198 (1994), pp. 283–295.

[22] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, U.K., 1992.

[23] 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.

[24] 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*, BIT, 36 (1996), pp. 595–633.

[25] G. L. G. Sleijpen and D. R. Fokkema, *BiCGstab(ℓ) for linear equations involving matrices with complex spectrum*, Elec. Trans. Numer. Anal., 1 (1993), pp. 11–32.

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

[27] G. L. G. Sleijpen and H. A. Van der Vorst, *The Jacobi-Davidson method for eigenvalue problems and its relation to accelerated inexact Newton schemes*, in Iterative Methods in Linear Algebra, II, S. D. Margenov and P. S. Vassilevski, eds., Proceedings of the Second IMACS International Symposium on Iterative Methods in Linear Algebra, June 17–20, 1995, Blagoevgrad, Baltzer, Basel, IMACS Ann. Comput. Appl. Math. 3, 1996.

[28] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[29] 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.

[30] G. W. Stewart and Ji guang Sun, *Matrix Perturbation Theory*, Academic Press, San Diego, CA, 1990.

[31] A. van der Sluis and H. A. Van der Vorst, *The rate of convergence of conjugate gradients*, Numer. Math., 48 (1986), pp. 543–560.

[32] H. A. Van der Vorst, *Preconditioning by Incomplete Decompositions*, Ph.D. thesis, Utrecht University, Utrecht, The Netherlands, 1982.

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

[34] P. Van Dooren, *Algorithm* 590*,* DUSBSP *and* EXCHQZ*:* FORTRAN *subroutines for computing deflating subspaces with specified spectrum*, ACM Trans. Math. Software, 8 (1982), pp. 376–382.