A preconditioned Jacobi-Davidson method for solving large generalized eigenvalue problems

J.G.L. Booten [1], H.A. van der Vorst [2], P.M. Meijer [3], and H.J.J. te Riele [4]

[1] *CWI, P.O. Box 94079, 1090 GB Amsterdam, the Netherlands (booten@cwi.nl)*
[2] *Mathematical Institute, University of Utrecht,*
P.O. Box 80.010, 3508 TA Utrecht, the Netherlands (vorst@math.ruu.nl)
[3] *FOM Institute for Plasmaphysics,*
P.O. Box 1207, 3430 BE Nieuwegein, the Netherlands (meijer@rijnh.nl)
[4] *CWI, P.O. Box 94079, 1090 GB Amsterdam, the Netherlands (herman@cwi.nl)*

Abstract

In this paper we apply the recently proposed Jacobi-Davidson method for calculating extreme eigenvalues of large matrices to a generalized eigenproblem. This leads to an algorithm that computes the extreme eigensolutions of a matrix pencil (A, B), where A and B are general matrices. Factorization of either of them is avoided. Instead we need to solve two linear systems with sufficient, but modest accuracy. If both linear systems are solved accurately enough, an asymptotically quadratic speed of convergence can be achieved. Interior eigenvalues in the vicinity of a given complex number σ can be computed without factorization as well. We illustrate the procedure with a few numerical examples, one of them being an application in magnetohydrodynamics.

AMS Subject Classification (1991): 65F15
CR Subject Classification (1991): G.1.3
Keywords & Phrases: Eigenvalues, eigenvectors, matrix pairs, Jacobi-Davidson method, GMRES, preconditioner, magnetohydrodynamics

1. Introduction

The numerical solution of a generalized eigenvalue problem

$$Ax = \lambda Bx, \tag{1.1}$$

where A and B are general $n \times n$ matrices, $\lambda \in \mathcal{C}$ and $x \in \mathcal{C}^n$, has many applications in science and engineering. For the hermitian definite case, where both A and B are hermitian and, in addition, B is positive definite, we mention the well-known application area of structural analysis (see e.g. [17], chapter X). The solution of these generalized hermitian eigenvalue problems yields the natural vibration frequencies of large structures such as ships or aircrafts. The solution of a generalized non-hermitian eigenvalue problem is for instance required in the field of linear magnetohydrodynamics (MHD), where the interaction of hot ionized gases, so-called plasmas, with magnetic fields is studied. These phenomena play a dominant rôle in nuclear fusion experiments and in the solar atmosphere (corona) [2, 4, 10, 15].

The optimal solution method for the generalized eigenproblem (1.1) depends, apart from the types of matrices, strongly upon their size. If n is relatively small, the problem can be solved by means of similarity transformations as in the Householder-QR algorithm [9].

Suppose the matrices are too large to be treated by similarity transformations, but still of such a size that a factorization into triangular forms is feasible. In case of hermitian definite problems one can reduce the generalized problem (1.1) to a standard one by making use of the Cholesky factorization of B [8]. If $B = LL^*$, where L is lower triangular, the generalized problem (1.1) is equivalent to the standard hermitian problem

$$L^{-1}AL^{-*}y = \lambda y, \tag{1.2}$$

to which we can apply the standard Lanczos algorithm [9, 13]. As an alternative approach, noting that the matrix $B^{-1}A$ is self-adjoint with respect to the B-inner product, we can apply a Lanczos variant where the usual Euclidean inner products have been replaced by B-inner products [13, 17]. A similar approach is followed by Van der Vorst [23], who describes a generalized Lanczos scheme for the eigensystem of CB, with C symmetric (or skew-symmetric) and B symmetric positive definite. The generalized eigenvalue problem $Ax = \lambda Bx$ can be reduced to the above form $AB^{-1}y = \lambda y$, but a fast and accurate solver is required for linear systems of the form $Bu = v$. For generalized symmetric problems where B is singular, but the matrix $A - \sigma B$ is non-singular, with σ a suitably chosen shift, the so-called spectral transformation Lanczos algorithm [7, 17] can be used, i.e., the Lanczos algorithm applied to a shifted and inverted problem $(A - \sigma B)^{-1}B$. The latter algorithm is also very useful for computing interior eigenvalues. Furthermore, this shift-and-invert strategy may lead to a better separation of the desired eigenspectrum, which can result in faster convergence. The commonly followed procedure for solving generalized non-hermitian eigenvalue problems is the shift-and-invert Arnoldi method [14, 17]. Recently developed variants of this technique involve multiple shifts in one run [16]. Another option is the generalized non-hermitian Lanczos procedure [4],which is also based upon a shift-and-invert strategy.

When the order of the matrices becomes so large that factorization is very impractical, we have to search for different methods. Alternatively, in several of the algorithms mentioned above, one might replace the factorization into triangular forms and subsequently solving a lower and upper triangular system by a preferably fast, but accurate iterative linear system solver. However, such an approach can also easily become very expensive. For the generalized symmetric eigenvalue problem, where A is symmetric and B is symmetric positive definite, several algorithms have been suggested, in which factorization of A or B is avoided. In [19] Scott proposed an inner-outer iteration scheme, in which the inner iteration consists of solving a nearby standard problem by the Lanczos method. However, the algorithm turned out to be inappropriate for computing interior eigenvalues or more than a few extreme eigenvalues. Another interesting algorithm is Szyld's two-level iterative method [22]. The idea is to combine inverse and Rayleigh quotient iteration. The method finds an eigenvalue in a given interval or, if this interval is free of eigenvalues, the eigenvalue closest to this interval. The drawback is again that at each iteration step a linear system needs to be solved.

Recently, Sleijpen and Van der Vorst [20] have presented the Jacobi-Davidson method for computing the extreme eigenvalues and associated eigenvectors of a general matrix. As we will show, the method has very advantageous implications when applied to a generalized eigenvalue problem. More specifically, the method is applicable to general matrix pairs (A, B) and neither the factorization of a matrix nor the accurate solution of a linear system is necessary. Instead we need to solve two linear systems with sufficient accuracy to guarantee convergence of the eigenvalues. Furthermore, these advantages make the method very suitable for an efficient implementation on parallel computers.

This paper is organized as follows. In section 2 we briefly describe the Jacobi-Davidson method for standard eigenvalue problems. For more details on e.g. the convergence analysis or numerical experiments for standard problems, see [20]. In section 3 we discuss the new Jacobi-Davidson algorithm for matrix pairs. In section 4 we illustrate the method with a few numerical examples. In the last section we give some concluding remarks.

2. Jacobi-Davidson method for standard problems

2.1 Davidson's method

The basic idea of the classical Davidson method [5, 6] is a rather simple one. Suppose at a certain stage we have obtained an approximation (θ_j, u_j) for the largest eigenpair of a large matrix A over a subspace, of which the set of vectors $v_1, v_2, \ldots, v_j$ form an orthonormal basis. Let $V_j \in \mathcal{C}^{n\times j}$ denote the matrix whose columns are the vectors $v_1, \ldots, v_j$. The question then rises how to improve this approximation for the largest eigenpair, or, in other words, what is a useful extension of the current subspace given by V_j. Davidson suggests to solve for t the equation

$$M_j t = Au_j - \theta_j u_j = r, \tag{2.3}$$

with r the residual vector for (θ_j, u_j). The matrix M_j is considered to be a preconditioning matrix, for which Davidson proposes $(D_A - \theta_j I)$, where D_A is the main diagonal of A. After solving (2.3) for t, the vector t has to be orthogonalized against V_j, after which the resulting vector is added to V_j to obtain the subspace given by V_{j+1}. Subsequently, the orthogonal restriction of A with respect to the enlarged subspace has to be computed, i.e., the $(j+1)\times(j+1)$ matrix $H_{j+1} = V_{j+1}^* A V_{j+1}$, and then the update of (θ_j, u_j) is obtained by computing the largest eigenpair of H_{j+1}. Note that the matrices H_j are in general full matrices, in contrast to the tridiagonal (upper Hessenberg) matrices obtained with the Lanczos (Arnoldi) algorithms. If we choose $M_j = I$ in equation (2.3) Davidson's method reduces to a standard Lanczos (Arnoldi) procedure. Davidson's approach turned out to be particularly successful in Quantum Chemistry. This is due to the strongly diagonal dominance of the symmetric matrices in this application area.

In recent years Davidson's method has been generalized, in the sense that more general preconditioning matrices M_j have been used [3, 11]. For instance Crouzeix, Philippe and Sadkane [3] give some numerical examples for symmetric matrices in which $M_j = (T_A - \theta_j I)$, where T_A is the tridiagonal part of matrix A.

However, as pointed out by Sleijpen and Van der Vorst [20], considering M_j in equation (2.3) as a preconditioner for $(A - \theta_j I)$, i.e., M_j should be a good approximation for $(A - \theta_j I)$, is a clear misconception. This becomes obvious if we substitute for M_j the 'ideal' case $(A - \theta_j I)$, since then equation (2.3) reduces to $t = u_j$, or, in other words, the vector t, by which we should expand the current subspace, is already in this subspace and consequently the method breaks down.

Sleijpen and Van der Vorst realized that Davidson's method is closely related to an old method of Jacobi, designated in [20] as Jacobi's orthogonal component correction (JOCC) method. For a description of this method in modern matrix notation, see [20]. The main difference between the two methods is that Davidson exploits a complete subspace at each iteration step, while Jacobi continues with a linear combination of the current eigenvector approximation and the calculated correction vector. In fact, Davidson's method can be seen as an accelerated JOCC method. These considerations led to the Jacobi-Davidson iteration method.

2.2 The Jacobi-Davidson iteration method

The current approximation u_j for the largest eigenvector of matrix A in a subspace spanned by the orthonormal basis vectors $v_1, \ldots, v_j$ can be written as $u_j = V_j y$, with $y \in \mathcal{C}^j$. Assume that u_j is normalized: $||u_j||_2 = ||y||_2 = 1$. The update for the current approximation (θ_j, u_j) for the largest eigenpair of A has to be sought in the subspace orthogonal to u_j. The orthogonal projection of A onto this subspace $u_j^\perp$ is the matrix A_P given by

$$A_P = (I - u_j u_j^*)A(I - u_j u_j^*). \tag{2.4}$$

Using the relation

$$u_j^* A u_j = y^* V_j^* A V_j y = y^* H_j y = y^* \theta_j y = \theta_j \tag{2.5}$$

we can rewrite equation (2.4):

$$A = A_P + u_j u_j^* A + A u_j u_j^* - \theta_j u_j u_j^*. \tag{2.6}$$

We are looking for a correction vector $z \perp u_j$ such that

$$A(z + u_j) = \lambda(z + u_j), \tag{2.7}$$

where $(\lambda, z + u_j)$ is the *true* largest eigenpair of A. Substituting the expression (2.6) for A into (2.7), we obtain, noting that $A_P u_j = 0$:

$$(A_P - \lambda I)z = -r + (\lambda - \theta_j - u_j^* A z)u_j. \tag{2.8}$$

Since $A_P z \perp u_j$, $z \perp u_j$ and $r \perp u_j$, it immediately follows that the coefficient in front of u_j on the right-hand side must be zero. Hence we need to solve for z:

$$(A_P - \lambda I)z = -r. \tag{2.9}$$

Of course we can not solve this equation exactly, since we do not know the true eigenvalue λ. However, an approximation θ_j for λ is available, and therefore we can approximate the optimal solution $-(A_P - \lambda I)^{-1} r$ by solving the following equation approximately for z:

$$(A_P - \theta_j I)z = -r, \tag{2.10}$$

or, inserting equation (2.4),

$$(I - u_j u_j^*)(A - \theta_j I)(I - u_j u_j^*)z = -r. \tag{2.11}$$

This equation is solved approximately by performing a few iteration steps with a suitable linear system solver, preferably with a good preconditioner. For a recent review on currently available iterative methods for solving large linear systems, see [1]. The obtained vector is then made orthogonal against V_j by means of the Modified Gram-Schmidt (MGS) process [9, 17] and the resulting vector is added to V_j in order to obtain an enlarged subspace given by V_{j+1}. The orthogonal restriction of A with respect to the extended subspace is then given by $H_{j+1} = V_{j+1}^* A V_{j+1}$ and the update for (θ_j, u_j) is obtained by computing the largest eigenpair of H_{j+1}.

The Jacobi-Davidson algorithm for standard eigenvalue problems can then be written in the following form, restarting the iteration process after m steps:

ALGORITHM **2.1 Standard Jacobi-Davidson**

1. **Start:** Choose v_1 with $||v_1||_2 = 1$.
2. **Iterate:** Until convergence do:
3. **Inner Loop:** For $j = 1, \ldots, m$ do:
   - Compute $w := Av_j$.
   - Compute $V_j^* w$, the last column of $H_j = V_j^* A V_j$ and $v_j^* A V_{j-1}$, the last row of H_j (only if $A \neq A^*$).
   - Compute the largest eigenpair (θ_j, y) of H_j (with $||y||_2 = 1$).
   - Compute the Ritz vector $u_j := V_j y$ and its associated residual vector $r := Au_j - \theta_j u_j$.
   - Test for convergence. If satisfied return.
   - Solve approximately for z: $(I - u_j u_j^*)(A - \theta_j I)(I - u_j u_j^*) z = -r$ (skip when $j = m$).
   - Orthogonalize z against V_j via MGS, and expand V_j with this vector to V_{j+1} (skip when $j = m$).
4. **Restart:** Set $v_1 := u_m$ and go to 3.

Sleijpen and Van der Vorst [20] proved that if the linear system (2.11) is solved with sufficient accuracy, the speed of convergence of the algorithm is asymptotically quadratic. They illustrate the method for standard problems with a few numerical examples, using the generalized minimal residual algorithm GMRES of Saad and Schultz [18] to solve (2.11) approximately. They also present a variant to compute internal eigenvalues, which is based upon the harmonic Ritz value approach proposed in ref. [12].

3. JACOBI-DAVIDSON METHOD FOR GENERALIZED PROBLEMS

We now return to the generalized eigenvalue problem of the form (1.1). We make the following assumptions:

1. The matrix B is non-singular, i.e., B^{-1} exists.
2. The matrices A and B are so large that a factorization of A or B into triangular forms as well as the iterative, accurate solution of a linear system with A or B as the coefficient matrix is very undesirable.

Of course, in order to compute the extreme eigenvalues of the matrix pair (A, B), we could apply algorithm 2.1 to the equivalent standard problem $B^{-1}A$, but then the second assumption would be violated.

What happens if we project the matrix $B^{-1}A$ onto a subspace $B^* V_j$ instead of V_j as is done for standard problems? We will use the same notation as in the previous section. Again we introduce the matrix $V_j \in \mathcal{C}^{n \times j}$ as $V_j = [v_1 | v_2 | \cdots | v_j]$. Define the matrix $\tilde{W}_j \in \mathcal{C}^{n \times j}$ as:

$$\tilde{W}_j = B^* V_j = [B^* v_1 | B^* v_2 | \cdots | B^* v_j] = [\tilde{w}_1 | \tilde{w}_2 | \cdots | \tilde{w}_j]. \tag{3.12}$$

Orthogonalization via the MGS process of the set of vectors $\tilde{w}_1, \ldots, \tilde{w}_j$ yields an orthonormal basis $w_1, \ldots, w_j$ of the subspace $B^* V_j$. Again we employ the matrix notation $W_j = [w_1|w_2|\cdots|w_j]$. Then we can write:

$$\tilde{W}_j = W_j R_j, \tag{3.13}$$

where R_j is a $j \times j$ upper triangular matrix whose entries are the orthogonalization coefficients obtained during the MGS procedure. Combining equations (3.12) and (3.13) we obtain as the orthonormal basis for the subspace $B^* V_j$:

$$W_j = B^* V_j R_j^{-1}. \tag{3.14}$$

Using this orthonormal basis, we can write the orthogonal restriction of $B^{-1}A$ with respect to the subspace $B^* V_j$ as

$$W_j^* B^{-1} A W_j = R_j^{-*} V_j^* B B^{-1} A B^* V_j R_j^{-1} = R_j^{-*} V_j^* A B^* V_j R_j^{-1} = R_j^{-*} H_j R_j^{-1}, \tag{3.15}$$

where we introduced the $j \times j$ matrix $H_j = V_j^* A B^* V_j$. An approximation (θ_j, u_j) for the largest eigenpair of $B^{-1}A$, and thus for the matrix pair (A, B), can now be found by computing the largest eigenpair (θ_j, s) of the projected matrix:

$$R_j^{-*} H_j R_j^{-1} s = \theta_j s, \tag{3.16}$$

or, which is similar,

$$(R_j^* R_j)^{-1} H_j R_j^{-1} s = \theta_j R_j^{-1} s. \tag{3.17}$$

Writing $R_j^{-1} s = y$ and $R_j^* R_j = M_j^2$, the projected eigenvalue problem we have to solve is:

$$M_j^{-2} H_j y = \theta_j y. \tag{3.18}$$

The approximate eigenvector is then given by $u_j = B^* V_j y$. Note that until now assumption 2 is still satisfied, since the only matrix we have to invert is M_j^2, which is of course a very small matrix of order $j \ll n$.

The question now rises how to update the current eigenpair approximation (θ_j, u_j). For this purpose we have to solve to some accuracy equation (2.11), which for the generalized eigenvalue problem becomes:

$$(I - u_j u_j^*)(B^{-1}A - \theta_j I)(I - u_j u_j^*) z = -r, \tag{3.19}$$

where the residual vector $r = B^{-1} A u_j - \theta_j u_j$. Since we want assumption 2 to remain valid, we have to get rid of the B^{-1} on the left-hand side as well as in r. Note that a good preconditioner for solving this linear system, disregarding the u_j-direction eliminated from the matrix, should be an approximation of

$$(B^{-1}A - \theta_j I)^{-1} = (A - \theta_j B)^{-1} B. \tag{3.20}$$

Therefore, it is obvious that we should multiply equation (3.19) on both sides from the left with B. Doing this and introducing the vector $u_j' = V_j y$, we obtain from (3.19):

$$(I - B u_j u_j'^*)(A - \theta_j B)(I - u_j u_j^*) z = -d, \tag{3.21}$$

where the defect vector $d = Br = Au_j - \theta_j Bu_j$. Note that we succeeded in finding an expression for the projected system we have to solve approximately, which is free of inversion of the A as well as the B matrix. Solving (3.21) results in a vector z, by which we have to expand the subspace B^*V_j. But we also need to compute the corresponding vector $t = B^{-*}z$ to expand V_j in order to calculate the matrix $H_j = V_j^*AB^*V_j$. Since we do not want to violate assumption 2, we suggest to compute an approximation for t by solving (approximately)

$$B^*t = z. \tag{3.22}$$

We then extend V_j with t to V_{j+1}, compute B^*t – which is again an approximation of the vector z computed via (3.21) – and expand $\tilde{W}_j$ with this vector to $\tilde{W}_{j+1}$. Next we calculate $H_{j+1} = V_{j+1}^*AB^*V_{j+1}$, and orthonormalize $\tilde{W}_{j+1}$ which gives W_{j+1} and R_{j+1}. Subsequently, we compute $M_{j+1}^2 = R_{j+1}^*R_{j+1}$ and $M_{j+1}^{-2}H_{j+1}$. An update for (θ_j, u_j) is then obtained by computing the largest eigenpair of $M_{j+1}^{-2}H_{j+1}$.

Define the matrix $X_j \in \mathcal{C}^{n\times j}$ as:

$$X_j = AB^*V_j = [AB^*v_1|AB^*v_2|\cdots|AB^*v_j] = [x_1|x_2|\cdots|x_j]. \tag{3.23}$$

The Jacobi-Davidson algorithm for computing the extreme eigenvalue and corresponding eigenvector of a matrix pair (A, B) can now be written in the following form, again restarting after m steps:

Algorithm **3.1 Jacobi-Davidson for matrix pairs**

1. **Start:** Choose an initial vector v_1 and compute $\tilde{w}_1 := B^*v_1$ and $x_1 := A\tilde{w}_1$; choose a tolerance *tol.*

2. **Iterate:** Until convergence do:

3. **Inner Loop:** For $j = 1, \ldots, m$ do:

   - Compute $V_j^*x_j$, the last column of $H_j = V_j^*AB^*V_j$ and $v_j^*X_{j-1}$, the last row of H_j.
   - Orthonormalize, using MGS, the set $[W_{j-1}|\tilde{w}_j]$ to obtain W_j, i.e., compute the last column of R_j.
   - Compute $M_j^2 := R_j^*R_j$ and $M_j^{-2}H_j$.
   - Compute the largest eigenpair (θ_j, y) of $M_j^{-2}H_j$.
   - Compute the Ritz vector $u_j := \tilde{W}_jy$ (with y normalized such that $||u_j||_2 = 1$); compute the vector $u_j' := V_jy$ and the defect vector $d := X_jy - \theta_jBu_j$.
   - Test for convergence, i.e., if $||d||_2 < tol$, return.
   - Solve approximately for z:
     $(I - Bu_ju_j'^*)(A - \theta_jB)(I - u_ju_j^*)z = -d$;
     solve approximately for t:
     $B^*t = z$
     (skip when $j = m$).

- Set $v_{j+1} := t$; expand V_j with this vector to V_{j+1};
  compute $\tilde{w}_{j+1} := B^*t$; expand $\tilde{W}_j$ with this vector to $\tilde{W}_{j+1}$;
  compute $x_{j+1} := A\tilde{w}_{j+1}$; expand X_j with this vector to X_{j+1}
  (skip when $j = m$).

4. **Restart:** Set $v_1 := u'_m$, $\tilde{w}_1 := u_m$ and $x_1 := X_m y$; go to 3.

Note that from the implementation point of view four $n \times j$ matrices need to be stored: $V_j, \tilde{W}_j, W_j$ and X_j. Every iteration a new vector is added to these arrays and every iteration we only have to compute the last column and row of H_j and the last column of R_j. Further it should be remarked that the stopping criterion is not determined by the residual vector r, as in the standard Jacobi-Davidson algorithm 2.1, but by the vector $d = Br$. It is obvious that the convergence analysis presented in ref. [20] for standard problems also applies to the generalized case outlined in this paper. More precisely, the proof that the speed of convergence of algorithm 2.1 is asymptotically quadratic, provided the projected linear system (2.11) is solved with sufficient accuracy, can straightforwardly be extended to algorithm 3.1, with the exception that two linear systems, eqs. (3.21) and (3.22), have to be solved accurately enough. For this purpose it will be necessary in many applications to select suitable preconditioners for both linear systems, as we will show below in the numerical examples. These examples will clearly demonstrate that the asymptotically quadratic convergence can indeed be achieved.

Until now we have assumed that the B matrix is non-singular, i.e., assumption 1 was satisfied. If this is not the case, one can introduce a suitable shift σ such that the matrix $A-\sigma B$ is non-singular and run the algorithm for an equivalent matrix pair $(A', B') = (B, A - \sigma B)$. If this will yield eigenvalues μ for the matrix pair (A', B'), the eigenvalues λ of the original pair (A, B) can be retrieved from $\lambda = \frac{1}{\mu} + \sigma$. This procedure must also be followed if interior eigenvalues need to be computed. The algorithm will calculate the eigenvalue closest to the shift $\sigma \in \mathcal{C}$. Furthermore, this approach may speed up convergence, since the desired eigenvalues may be better separated.

Sofar, the algorithms described only compute the extreme eigenvalue and associated eigenvector. But in many practical situations a whole cluster of eigenvalues is desired. In order to compute several eigenvalues and corresponding eigenvectors at the same time, we suggest a block version. For this purpose, define the matrices $V_j, \tilde{W}_j, W_j$ and $X_j \in \mathcal{C}^{n\times jl}$ as:

$$\begin{aligned}
V_j &= [v_{1,1}|\cdots|v_{1,l}|v_{2,1}|\cdots|v_{2,l}|\cdots\cdots|v_{j,1}|\cdots|v_{j,l}],\\
\tilde{W}_j &= [\tilde{w}_{1,1}|\cdots|\tilde{w}_{1,l}|\tilde{w}_{2,1}|\cdots|\tilde{w}_{2,l}|\cdots\cdots|\tilde{w}_{j,1}|\cdots|\tilde{w}_{j,l}],\\
W_j &= [w_{1,1}|\cdots|w_{1,l}|w_{2,1}|\cdots|w_{2,l}|\cdots\cdots|w_{j,1}|\cdots|w_{j,l}],\\
X_j &= [x_{1,1}|\cdots|x_{1,l}|x_{2,1}|\cdots|x_{2,l}|\cdots\cdots|x_{j,1}|\cdots|x_{j,l}]. \qquad (3.24)
\end{aligned}$$

In the following block variant of algorithm 3.1 we compute the largest l eigenvalues and associated eigenvectors of a matrix pair (A, B). Each outer iteration we add l new vectors to the bases defined in (3.24). Also note that each outer iteration l columns and rows are added to the matrix H_j, and l columns to the matrix R_j. The order of the matrices H_j, R_j and M_j^2 is jl. Furthermore, $2l$ linear systems need to be solved approximately during every outer iteration. We restart after m iterations. Then the block version has the following form:

ALGORITHM **3.2 Block Jacobi-Davidson for matrix pairs**

1. **Start:** Choose an initial matrix of rank l: $V_1 = [v_{1,1}|\cdots|v_{1,l}]$;
   compute $\tilde{W}_1 := B^* V_1$ and $X_1 := A\tilde{W}_1$; choose a tolerance *tol.*

2. **Iterate:** Until convergence do:

3. **Inner Loop:** For $j = 1, \ldots, m$ do:

   - Compute $V_j^* x_{j,i}$ $(i = 1, \ldots, l)$, the last l columns of $H_j = V_j^* AB^* V_j$ and $v_{j,i}^* X_{j-1}$ $(i = 1, \ldots, l)$, the last l rows of H_j.
   - Orthonormalize, using MGS, the set $[W_{j-1}|\tilde{w}_{j,1}, \ldots, \tilde{w}_{j,l}]$ to obtain W_j, i.e., compute the last l columns of R_j.
   - Compute $M_j^2 := R_j^* R_j$ and $M_j^{-2} H_j$.
   - Compute the largest l eigenpairs (θ_i, y_i) of $M_j^{-2} H_j$ $(i = 1, \ldots, l)$.
   - Compute the Ritz vectors $u_i := \tilde{W}_j y_i$ $(i = 1, \ldots, l)$
     (with y_i normalized such that $||u_i||_2 = 1$);
     compute the vectors $u_i' := V_j y_i$ $(i = 1, \ldots, l)$
     and the defect vectors $d_i := X_j y_i - \theta_i B u_i$ $(i = 1, \ldots, l)$.
   - Test for convergence, i.e., if $||d_1||_2 < tol$, return.
   - For $i = 1, \ldots, l$ do:
     solve approximately for z_i:
     $(I - Bu_i u_i'^*)(A - \theta_i B)(I - u_i u_i^*) z_i = -d_i$;
     solve approximately for t_i:
     $B^* t_i = z_i$
     (skip when $j = m$).
   - Set $v_{j+1,i} := t_i$ $(i = 1, \ldots, l)$; expand V_j with these vectors to V_{j+1};
     compute $\tilde{w}_{j+1,i} := B^* t_i$ $(i = 1, \ldots, l)$; expand $\tilde{W}_j$ with these vectors to $\tilde{W}_{j+1}$;
     compute $x_{j+1,i} := A\tilde{w}_{j+1,i}$ $(i = 1, \ldots, l)$; expand X_j with these vectors to X_{j+1}
     (skip when $j = m$).

4. **Restart:** Set $V_1 := [u_1'|\cdots|u_l'], \tilde{W}_1 := [u_1|\cdots|u_l]$
   and $X_1 := [X_m y_1|\cdots|X_m y_l]$; go to 3.

4. Numerical experiments

We illustrate the algorithms presented above with two numerical examples. Both concern generalized non-hermitian eigenvalue problems, but the method applies to other types of problems as well, as said before. The first example is a small test problem, which provides good insights into the method. The second problem is a more practical one, taken from the application area of magnetohydrodynamics (MHD), see [2, 4, 10, 15]. The examples have been coded in FORTRAN and executed on a SGI workstation in about 15 decimals working precision. The projected eigenvalue problem (3.18) of order j is solved completely with the Householder-QR algorithm [9], i.e., all eigenvalues and corresponding eigenvectors have been calculated, but of course algorithm 3.1 (3.2) continues only with the largest (l) eigensolution(s).

4.1 Example 1. A small test problem.
Consider a test problem of order $n = 80$. The A matrix is tridiagonal non-symmetric and has the following non-zero entries (all other entries are zero):

$$a_{ij} = \begin{cases} i & \text{if } j = i \\ 1 & \text{if } j = i+1 \\ -1 & \text{if } j = i-1 \end{cases}, \tag{4.25}$$

and the B matrix is symmetric with the following non-zero entries (all other entries are zero):

$$b_{ij} = \begin{cases} 1 & \text{if } j = i \\ -1 & \text{if } j = i+1 \\ -1 & \text{if } j = i-1 \\ 1 & \text{if } i = 1 \text{ and } j = n \\ 1 & \text{if } i = n \text{ and } j = 1 \end{cases}. \tag{4.26}$$

The largest two eigenvalues of this matrix pencil come in a complex conjugate pair:

$$\lambda_{1,2} = 1777.5242385154\ldots \pm i\ 71.487254566584\ldots. \tag{4.27}$$

Suppose we want to obtain an approximation θ_1 for λ_1. In order to prevent algorithm 3.1 from switching between λ_1 and λ_2 we run the code with a shift $\sigma = 1700.0 + i\ 50.0$. We restart after 10 outer iterations, i.e., $m = 10$. The linear systems given in eqs. (3.21) and (3.22) are solved approximately with 30 steps of GMRES [18] (each time with initial guess 0). The starting vector is $v_1^T = (1, \ldots, 1)^T$. The stopping tolerance *tol* is set at 10^{-8}. No preconditioners have been used for eqs. (3.21) and (3.22). Algorithm 2.1 converged to the desired eigenvalue in 46 outer iterations, which corresponds to 1350 inner iteration steps for each of the linear systems. The convergence history is plotted in figures 1 and 2. Figure 1 shows $\log ||d||$ as a function of the outer iterations, while figure 2 displays the absolute distance between the eigenvalue approximation θ_1 and the true eigenvalue λ_1.

Suppose now we want to calculate the eigenvalue with the smallest absolute value. The true eigenvalue λ_n is real and is equal to 0.99578702736351.... We run algorithm 3.1 under the same conditions as before, except for σ, which equals 0 in the present case. For convergence the algorithm required 20 outer iterations (570 inner iteration steps for each of the linear systems). The convergence history for this eigenvalue is shown in figures 3 and 4.

From a comparison of figures 1 and 2 with figures 3 and 4, respectively, one might get the impression that the asymptotical convergence for the largest eigenvalue λ_1 is much slower than for the smallest eigenvalue λ_n. An explanation for this asymptotical convergence behaviour should be sought in the accuracy by which the linear systems (3.21) and (3.22) are solved. A measure for this accuracy is the norm of the residual vector r, obtained after 30 steps of GMRES. In figure 5 we show the accuracy of the inner iterations (GMRES(30)) as a function of the outer iterations for λ_1 and λ_n, for the two linear systems we have to solve approximately (denoted as l.s. 1 (corresponding to equation (3.21)) and l.s. 2 (corresponding to equation (3.22)) in the figure). From this figure we clearly see that the linear systems are solved much better for the smallest eigenvalue λ_n than for the largest eigenvalue λ_1, which explains the convergence behaviour observed in figures 1 to 4.

Sofar we did not precondition eqs. (3.21) and (3.22). A suitable preconditioner for equation (3.21) should be a good approximation for $(A - \theta_j B)^{-1}$. Note that this preconditioner depends on j and should therefore be updated every iteration in principle. However, in many

Table 1: Tridiagonal preconditioning for λ_1 (example 1).

Iteration	Approximate Eigenvalue			Linear Systems	
j	$\mathcal{R}\mathrm{e}\ \theta_1$	$\mathcal{I}\mathrm{m}\ \theta_1$	$\|\|d\|\|_2$	$\|\|r\|\|$ (l.s. 1)	$\|\|r\|\|$ (l.s. 2)
1	−38.874545	−0.011491	2.037684e−02	3.061445e−11	1.231821e−16
2	−6.362655	19.074912	1.827241e−02	2.479710e−12	3.044770e−16
3	66.823166	−173.660344	0.128156e+00	3.066878e−12	6.477340e−16
4	2462.219177	335.625493	2.580269e+00	1.411834e−14	1.020992e−15
5	1833.809678	37.310251	4.937574e+00	2.809987e−14	4.020756e−15
6	1777.542974	71.603097	0.265816e+00	3.535869e−15	3.601061e−15
7	1777.524235	71.487257	3.616086e−06	5.504482e−10	3.301667e−15
8	1777.524239	71.487255	3.768551e−10		

practical situations it might be more economical to update the preconditioner once in a few iterations, in particular towards the end, where θ_j does not change that much anymore. An appropriate preconditioner for equation (3.22) should approximate B^{-*}. For the present example we used tridiagonal preconditioning, i.e., the preconditioning matrix is the inverse of the tridiagonal part of $(A - \theta_j B)$ for equation (3.21) (updated every iteration), and the inverse of the tridiagonal part of B^* for equation (3.22). The impact of preconditioning is shown in table 1 for the largest eigenvalue λ_1. The starting parameters were the same as given above for the unpreconditioned case. We used 3 steps GMRES to obtain approximate solutions of eqs. (3.21) and (3.22). The residual norms for solving the linear systems are given in the last two columns of table 1. The approximate eigenvalue and the residual norm of the defect vector d, defined in section 3, is given in columns 2 to 4. Note that only 8 outer iterations are required, corresponding to 21 inner iteration steps for each linear system, to achieve convergence. Compare this with the unpreconditioned result shown above, where we needed 46 outer iterations, or, which is even more significant, 1350 inner iteration steps. Also note, by looking at column 4, that the speed of convergence is asymptotically quadratic, as might be expected since the linear systems are solved very accurately (columns 5 and 6), in many cases upto machine precision.

The largest six eigenvalues of this test problem are:

$$\begin{aligned} \lambda_{1,2} &= 1777.5242385154\ldots \pm i\ 71.487254566584\ldots \\ \lambda_{3,4} &= -367.02112288537\ldots \pm i\ 13.611724960533\ldots \\ \lambda_{5,6} &= 247.27064434612\ldots \pm i\ 10.523631113392\ldots. \end{aligned} \tag{4.28}$$

To obtain approximations for the eigenvalues $\lambda_{1,2}$ and $\lambda_{5,6}$ at the same time, we used the Block Jacobi-Davidson variant, algorithm 3.2, with the following starting parameters: $l = 4$, $m = 4$, tol=10^{-8}, $\sigma = 1200.0 + i\ 0.0$ and $V_1 = [e_1|e_2|e_3|e_4]$, with e_i the i-th canonical unit vector. It turned out that in the unpreconditioned case (30 steps GMRES) only 17 outer iterations were required and the residual norms of the defect vectors d_i were all below 10^{-8}. With tridiagonal preconditioning 29 outer iterations (2 steps GMRES) were necessary.

4.2 Example 2. A test problem from MHD.

The second example is taken from the application area of magnetohydrodynamics (MHD) [2, 4, 10, 15]. In these generalized non-hermitian eigenvalue problems the A matrix is non-hermitian and the B matrix is hermitian positive definite. Both matrices are of block tridiagonal shape with rather dense blocks. The interesting part of the spectrum in MHD problems is not the outer part of the spectrum, but an internal branch, known as the Alfvén spectrum. The test problem we consider here is taken from ref. [10]. The order of the matrices is 416. The blocks have dimension 16. The Alfvén spectrum is shown in fig. 6. Note the strong clustering of eigenvalues around the origin. In [10] the spectrum was calculated with a complex shift-and-invert Arnoldi method, using the implicit restart procedure of Sorensen [21]. It turned out that the Krylov dimension had to be set at 60 to reproduce the spectrum of this rather ill-conditioned test problem. In [2] we studied the same problem with an Arnoldi method for internal eigenvalues based upon the harmonic Ritz value approach outlined for the symmetric case in ref. [12]. It should be remarked that the inversion of matrices is avoided in this procedure, similar to the method presented in this paper. However, convergence turned out to be very slow: 460 outer iterations were necessary to achieve a reasonable approximations of eigenvalues close to the shift. Moreover, the dimension of the Krylov subspace, built up each iteration, had to be set at 125, i.e., 57500 inner iteration steps were required.

Here we study the same MHD problem with the Jacobi-Davidson algorithm for matrix pairs (algorithm 3.1). We try to calculate an approximation for the eigenvalue closest to the shift $\sigma = -0.3 + i\ 0.65$ (indicated in fig. 6 by a square), i.e., for the eigenvalue $\lambda = -0.294003753\ldots + i\ 0.587154648\ldots$. For the tolerance *tol* we take 10^{-6}, the starting vector $v_1^T = (1, \ldots, 1)^T$ and we restart the algorithm every 10 iterations ($m = 10$). Without preconditioning the equations (3.21) and (3.22), we were not able to reach convergence to the desired eigenvalue. The linear systems were not solved sufficiently accurate. Therefore, preconditioning seems to be an absolute necessity to solve the MHD eigenvalue problems. For the problem considered here we tried block Jacobi preconditioning, see e.g. [1]. First we tried block diagonal preconditioning matrices, where the blocks are the inverses of the diagonal blocks of the matrices $(A - \theta_j B)$ for equation (3.21) and of the matrix B^* for equation (3.22). However, still no convergence was achieved. Then we tried block diagonal preconditioning matrices, that are the inverses of the block diagonal part of $(A - \theta_j B)$ and B^*, respectively, with the blocks 2×2 block matrices (i.e., the block size is 32). It turned out that convergence to the desired eigenvalue was achieved in 26 outer iterations, provided we solved the linear systems with 70 and 125 GMRES iteration steps, respectively, which means that 1750 inner iteration steps were required for linear system (3.21) and 3125 inner iteration steps for linear system (3.22). The convergence history is plotted in figures 7 and 8.

It is obvious that we need more sophisticated preconditioners than the ones tried here in order to gain faster convergence for MHD eigenvalue problems. This will be the subject of future research; in particular, we will look for preconditioning techniques which preserve the suitability of the method for efficient parallel implementation, which is required to solve very large MHD problems (see [2, 15]).

5. Conclusion

In this final section we briefly summarize the main conclusions. We have developed a variant of the recently proposed Jacobi-Davidson method [20], that is applicable to general matrix pairs. An advantage of the new algorithm as compared with other methods, is that the inversion of a matrix, or the accurate iterative solution of a linear system, can be avoided. This should make the method very suitable to solve extremely large generalized eigenvalue problems. The speed of convergence can be asymptotically quadratic, provided two linear systems are solved with sufficient accuracy. In order to solve the two linear systems accurately enough to reach convergence, good preconditioners will be a prerequisite in many practical applications.

Acknowledgement
The authors gratefully acknowledge the many useful discussions with the other participants in the NWO-MPR project 'Parallel Computations in MHD'.

REFERENCES

1. R. BARRETT, M. BERRY, T.F. CHAN, J. DEMMEL, J. DONATO, J. DONGARRA, V. EIJKHOUT, R. POZO, C. ROMINE, AND H. VAN DER VORST, *Templates for the solution of linear systems: building blocks for iterative methods*, SIAM, Philadelphia, PA, 1994.
2. J.G.L. BOOTEN, P.M. MEIJER, H.J.J. TE RIELE, AND H.A. VAN DER VORST, *Parallel Arnoldi method for the construction of a Krylov subspace basis: an application in magnetohydrodynamics*, in Proceedings International Conference and Exhibition on High-Performance Computing and Networking, Munich, Germany, April 1994, Volume II: Networking and Tools, W. Gentzsch and U. Harms, eds., Lecture Notes in Computer Science 797, Springer-Verlag, Berlin, 1994.
3. M. CROUZEIX, B. PHILIPPE, AND M. SADKANE, *The Davidson method*, SIAM J. Sci. Comput., 15 (1994), pp. 62–76.
4. J. CULLUM, W. KERNER, AND R. WILLOUGHBY, *A generalized nonsymmetric Lanczos procedure*, Comput. Phys. Commun., 53 (1989), pp. 19–48.
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. E.R. DAVIDSON, *Monster matrices: their eigenvalues and eigenvectors*, Computers in Physics, 7 (1993), pp. 519–522.
7. T. ERICSSON AND A. RUHE, *The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems*, Math. Comp., 35 (1980), pp. 1251–1268.
8. G.H. GOLUB, R. UNDERWOOD, AND J.H. WILKINSON, *The Lanczos algorithm for the symmetric* $Ax = \lambda Bx$ *problem*, Technical Report STAN–CS–72–270, Computer Science Department, Stanford University, 1972.
9. G.H. GOLUB AND C.F. VAN LOAN, *Matrix Computations*, Second edition, The Johns Hopkins University Press, Baltimore and London, 1989.
10. M.N. KOOPER, H.A. VAN DER VORST, S. POEDTS, AND J.P. GOEDBLOED, *Application of the implicitly updated Arnoldi method with a complex shift and invert strategy in MHD*, Preprint PP 93/061, FOM Institute for Plasmaphysics 'Rijnhuizen', Nieuwegein, 1993.
11. R.B. MORGAN AND D.S. SCOTT, *Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 817–825.
12. C.C. PAIGE, B.N. PARLETT, AND H.A. VAN DER VORST, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Technical Report PAM–579, University of California at Berkeley, 1993.
13. B.N. PARLETT, *The symmetric eigenvalue problem*, Prentice-Hall, Englewood Cliffs, N.J.,

1980.

14. B.N. PARLETT AND Y. SAAD, *Complex shift and invert strategies for real matrices*, Linear Algebra Appl., 88/89 (1987), pp. 575–595.

15. S. POEDTS, P.M. MEIJER, J.P. GOEDBLOED, H.A. VAN DER VORST, AND A. JAKOBY, *Parallel magnetohydrodynamics on the CM-5*, in Proceedings International Conference and Exhibition on High-Performance Computing and Networking, Munich, Germany, April 1994, Volume I: Applications, W. Gentzsch and U. Harms, eds., Lecture Notes in Computer Science 796, Springer-Verlag, Berlin, 1994.

16. A. RUHE, *Rational Krylov algorithms for nonsymmetric eigenvalue problems. II. Matrix pairs*, Linear Algebra Appl., 197/198 (1994), pp. 283–295.

17. Y. SAAD, *Numerical methods for large eigenvalue problems*, Manchester University Press, Manchester, 1992.

18. Y. SAAD AND M.H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

19. D.S. SCOTT, *Solving sparse symmetric generalized eigenvalue problems without factorization*, SIAM J. Numer. Anal., 18 (1981), pp. 102–110.

20. G.L.G. SLEIJPEN AND H.A. VAN DER VORST, *A generalized Jacobi-Davidson iteration method for linear eigenvalue problems*, Preprint nr. 856, Department of Mathematics, University of Utrecht, 1994.

21. D.C. SORENSEN, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Mat. Anal. Appl., 13 (1992), pp. 357–385.

22. D.B. SZYLD, *Criteria for combining inverse and Rayleigh quotient iteration*, SIAM J. Numer. Anal., 25 (1988), pp. 1369–1375.

23. H.A. VAN DER VORST, *A generalized Lanczos scheme*, Math. Comp., 39 (1982), pp. 559–561.

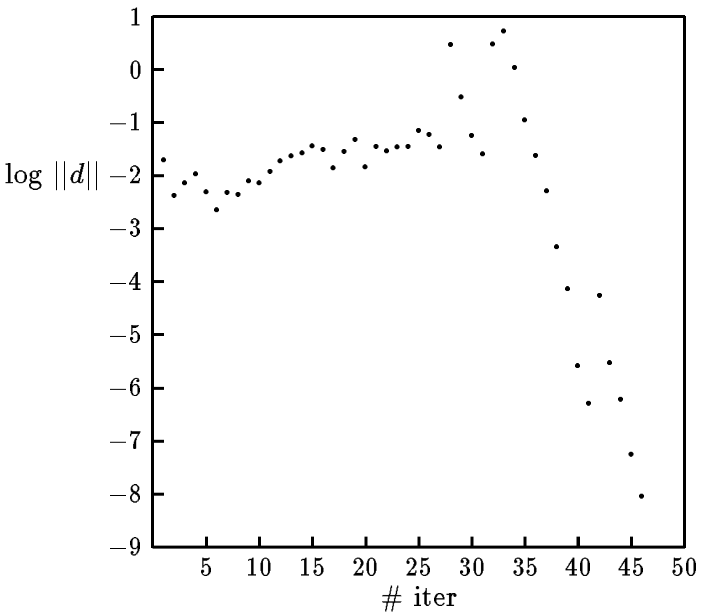

Figure 1: Convergence history for λ_1 : ${}^{10}\log||d||$. Unpreconditioned.

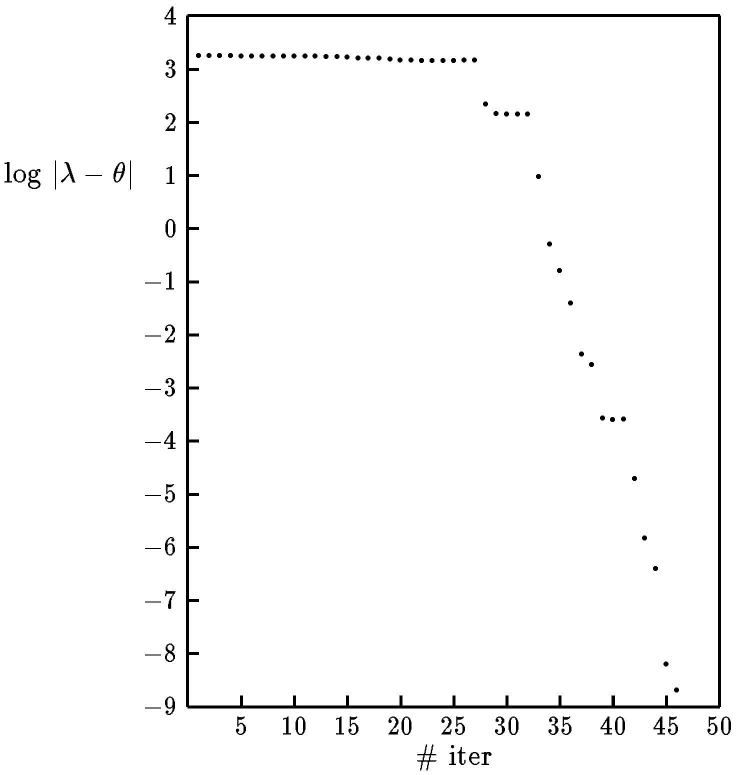

Figure 2: Convergence history for λ_1 : ${}^{10}\log|\lambda_1 - \theta_1|$. Unpreconditioned.

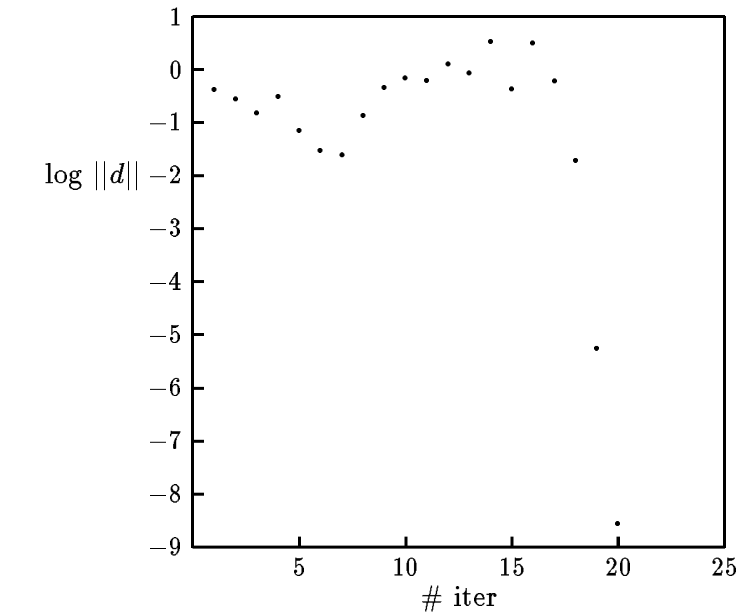

Figure 3: Convergence history for λ_n : ${}^{10}\log||d||$. Unpreconditioned.

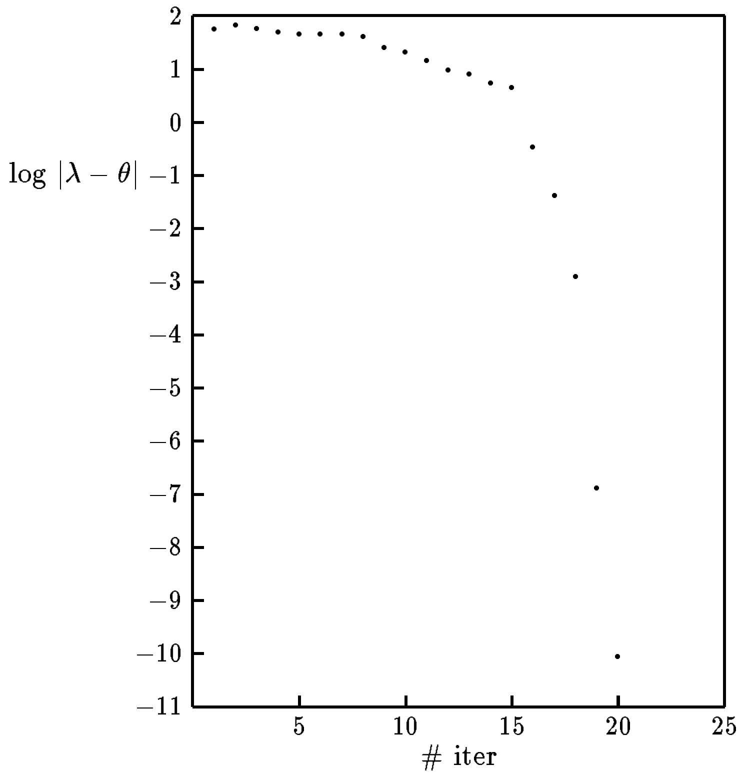

Figure 4: Convergence history for λ_n : ${}^{10}\log|\lambda_n - \theta_n|$. Unpreconditioned.

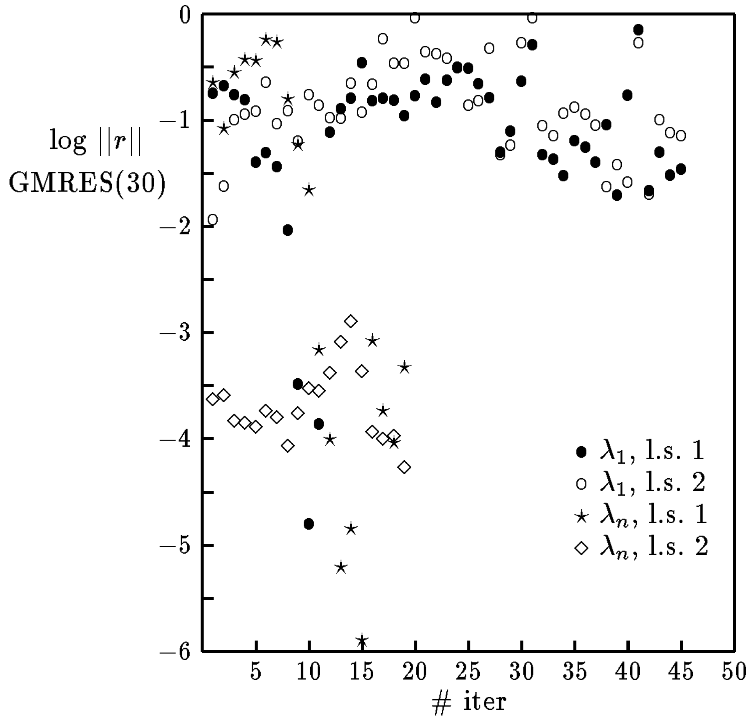

Figure 5: Residual norms inner iterations (GMRES(30)) for λ_1 and λ_n, for both linear systems (l.s. 1 and l.s. 2). Unpreconditioned.

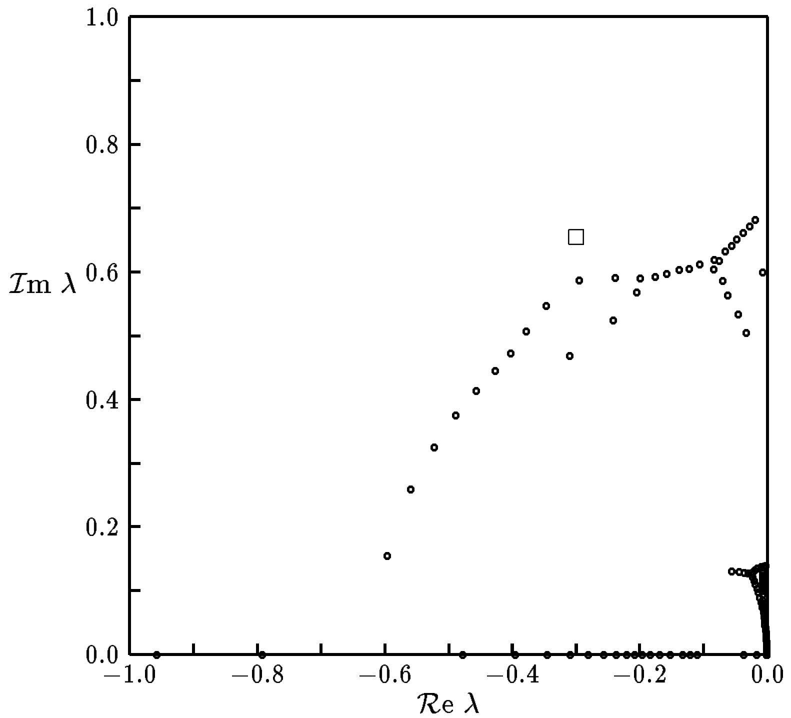

Figure 6: Alfvén spectrum MHD test problem ($n = 416$).

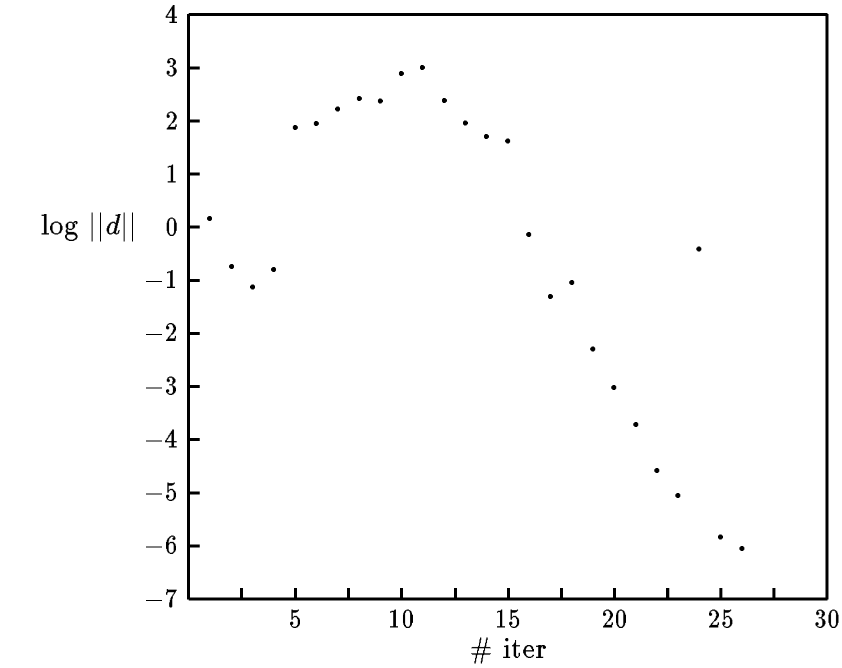

Figure 7: Convergence history MHD problem: ${}^{10}\log||d||$. Block Jacobi preconditioning.

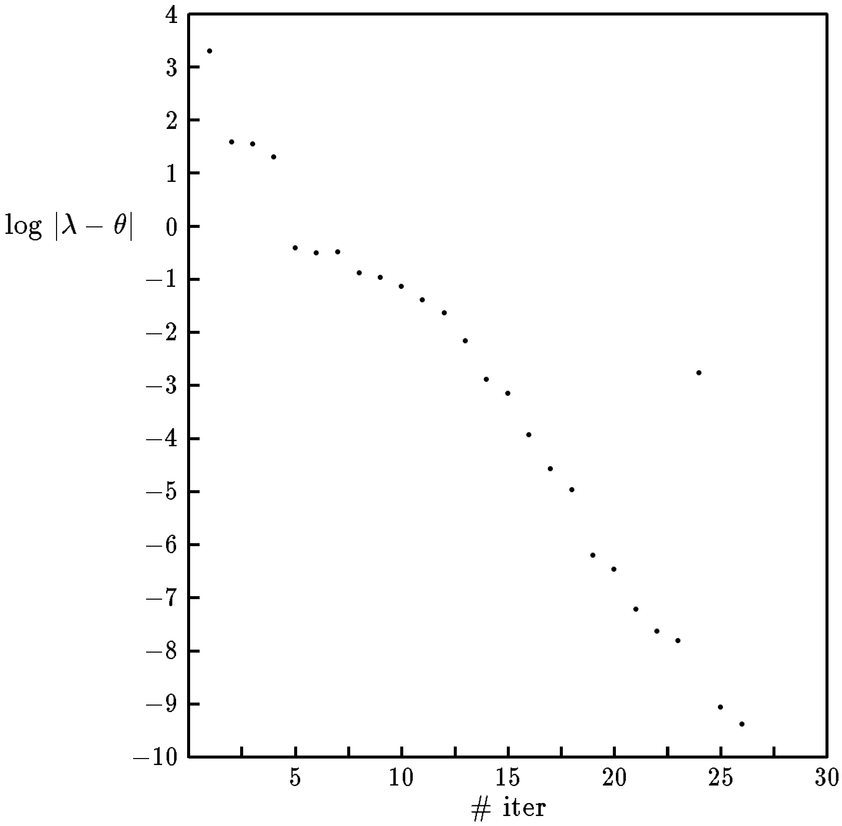

Figure 8: Convergence history MHD problem: ${}^{10}\log|\lambda - \theta|$. Block Jacobi preconditioning.