Parallel Jacobi-Davidson for Solving Generalized Eigenvalue Problems

Margreet Nool[1] and Auke van der Ploeg[2]

[1] CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Margreet.Nool@cwi.nl
[2] MARIN, P.O. Box 28, 6700 AA Wageningen, The Netherlands
A.v.d.Ploeg@marin.nl

Abstract. We study the Jacobi-Davidson method for the solution of large generalised eigenproblems as they arise in MagnetoHydroDynamics. We have combined Jacobi-Davidson (using standard Ritz values) with a shift and invert technique. We apply a complete LU decomposition in which reordering strategies based on a combination of block cyclic reduction and domain decomposition result in a well-parallelisable algorithm. Moreover, we describe a variant of Jacobi-Davidson in which harmonic Ritz values are used. In this variant the same parallel LU decomposition is used, but this time as a preconditioner to solve the 'correction' equation.

The size of the relatively small projected eigenproblems which have to be solved in the Jacobi-Davidson method is controlled by several parameters. The influence of these parameters on both the parallel performance and convergence behaviour will be studied. Numerical results of Jacobi-Davidson obtained with standard and harmonic Ritz values will be shown. Executions have been performed on a Cray T3E.

1 Introduction

Consider the generalised eigenvalue problem

$$Ax = \lambda Bx, \qquad A, B \in \mathcal{C}^{N_t \times N_t}, \tag{1}$$

in which A and B are complex block tridiagonal N_t-by-N_t matrices and B is Hermitian positive definite. The number of diagonal blocks is denoted by N and the blocks are n-by-n, so $N_t = N \times n$. In close cooperation with the FOM Institute for Plasma Physics "Rijnhuizen" in Nieuwegein, where one is interested in such generalised eigenvalue problems, we have developed a parallel code to solve (1). In particular, the physicists like to have accurate approximations of certain interior eigenvalues, called the *Alfvén spectrum*. A promising method for computing these eigenvalues is the Jacobi-Davidson (JD) method [3, 4]. With this method it is possible to find several interior eigenvalues in the neighbourhood of a given target σ and their associated eigenvectors.

J. Palma, J. Dongarra, and V. Hernández (Eds.): VECPAR'98, LNCS 1573, pp. 58–70, 1999.

In general, the sub-blocks of A are dense, those of B are rather sparse ($\approx 20\%$ nonzero elements) and N_t can be very large (realistic values are $N = 500$ and $n = 800$), so computer storage demands are very high. Therefore, we study the feasibility of parallel computers with a large distributed memory for solving (1).

In [2], Jacobi-Davidson has been combined with a parallel method to compute the action of the inverse of the block tridiagonal matrix $A - \sigma B$. In this approach, called DDCR, a block-reordering based on a combination of Domain Decomposition and Cyclic Reduction is combined with a complete block LU decomposition of $A - \sigma B$. Due to the special construction of L and U, the solution process parallelises well.

In this paper we describe two Jacobi-Davidson variants, one using standard Ritz values and one harmonic Ritz values. The first variant uses DDCR to transform the generalised eigenvalue problem into a standard eigenvalue problem. In the second one DDCR has been applied as a preconditioner to solve approximately the 'correction' equation. This approach results also into a projected standard eigenvalue problem with eigenvalues in the dominant part of the spectrum. In Section 2 both approaches are described. To avoid that the projected system becomes too large, we make use of a restarting technique. Numerical results, based on this technique, are analysed in Section 3. We end up with some conclusions and remarks in Section 4.

2 Parallel Jacobi-Davidson

2.1 Standard Ritz Values

The availability of a complete LU decomposition of the matrix $A - \sigma B$ gives us the opportunity to apply Jacobi-Davidson to a *standard* eigenvalue problem instead of a *generalised* eigenvalue problem. To that end, we rewrite (1) as

$$(A - \sigma B)x = (\lambda - \sigma)Bx. \tag{2}$$

If we define $Q := (A - \sigma B)^{-1}B$ then (2) can be written as

$$Qx = \mu x, \qquad \text{with } \mu = \frac{1}{\lambda - \sigma} \Leftrightarrow \lambda = \sigma + \frac{1}{\mu}. \tag{3}$$

The eigenvalues we are interested in form the dominant part of the spectrum of Q, which makes them relatively easy to find. The action of the operator Q consists of a matrix-vector multiplication with B, a perfectly scalable parallel operation, combined with two triangular solves with L and U.

At the k-th step of Jacobi-Davidson, an eigenvector x is approximated by a linear combination of k search vectors v_j, $j = 1, 2, \cdots, k$, where k is very small compared with N_t. Consider the N_t-by-k matrix V_k, whose columns are given by v_j. The approximation to the eigenvector can be written as $V_k s$, for some k-vector s. The search directions v_j are made orthonormal to each other, using Modified Gram-Schmidt (MGS), hence $V_k^* V_k = I$.

Let θ denote an approximation of an eigenvalue associated with the Ritz vector $u = V_k s$. The vector s and the scalar θ are constructed in such a way that the residual vector $r = QV_k s - \theta V_k s$ is orthogonal to the k search directions. From this Rayleigh-Ritz requirement it follows that

$$V_k^* Q V_k s = \theta V_k^* V_k s \Longleftrightarrow V_k^* Q V_k s = \theta s. \tag{4}$$

The order of the matrix $V_k^* Q V_k$ is k. By using a proper restart technique k stays so small that this 'projected' eigenvalue problem can be solved by a sequential method.

In order to obtain a new search direction, Jacobi-Davidson requires the solution of a system of linear equations, called the 'correction equation'. Numerical experiments show that fast convergence to selected eigenvalues can be obtained by solving the correction equation to some *modest accuracy* only, by some steps of an inner iterative method, e.g. GMRES.

Below we show the Jacobi-Davidson steps used for computing several eigenpairs of (3) using standard Ritz values.

```
step 0: initialize
    Choose an initial vector $v_1$ with $\|v_1\|_2 = 1$; set $V_1 = [v_1]$;
    $W_1 = [Qv_1]$; $k = 1$; $it = 1$; $n_{ev} = 0$
step 1: update the projected system
    Compute the last column and row of $H_k := V_k^* W_k$
step 2: solve and choose approximate eigensolution of projected system
    Compute the eigenvalues $\theta_1, \cdots, \theta_k$ of $H_k$ and choose $\theta := \theta_j$ with $|\theta_j|$ maximal
    and $\theta_j \neq \mu_i$, for $i = 1, \cdots, n_{ev}$; compute associated eigenvector $s$ with $\|s\|_2 = 1$
step 3: compute Ritz vector and check accuracy
    Let $u$ be the Ritz vector $V_k s$; compute the residual vector $r := W_k s - \theta u$;
    if $\|r\|_2 < tol_{sJD}.|\theta|$ then
      $n_{ev} := n_{ev} + 1$; $\mu_{n_{ev}} := \theta$; if $n_{ev} = N_{ev}$ stop; goto 2
    else if $it = iter$ stop
    end if
step 4: solve correction equation approximately with $it_{SOL}$ steps of GMRES
    Determine an approximate solution $\tilde{z}$ of $z$ in
    $(I - uu^*)(Q - \theta I)(I - uu^*)z = -r \;\wedge\; u^* z = 0$
step 5: restart if projected system has reached its maximum order
    if $k = m$ then
      5a: Set $k = k_{min} + n_{ev}$. Construct $C \in \mathcal{C}^{m \times k} \subset H_m$;
          Orthonormalize columns of $C$; compute $H_k := C^* H_m C$
      5b: Compute $V_k := V_m C$; $W_k := W_m C$
    end if
step 6: add new search direction
    $k := k + 1$; $it := it + 1$; call $MGS$ $[V_{k-1}, \tilde{z}]$; set $V_k = [V_{k-1}, \tilde{z}]$; $W_k = [W_{k-1}, Q\tilde{z}]$;
    goto 1
```

Steps 2 and 5a deal with the small projected system (4). Those sequential steps are performed by all processors in order to avoid communication. The basic ingredients of the other steps are matrix-vector products, vector updates and inner products. Since, for our applications, N_t is much larger than the number of processors, those steps parallelise well.

2.2 Harmonic Ritz Values

For the introduction of harmonic Ritz values we return to the original generalised eigenvalue problem (1). Assume $(\theta, V_k s)$ approximates an eigenpair (λ, x), then the residual vector r is given by

$$r = AV_k s - \theta BV_k s.$$

In case of standard Ritz values, the correction vector r has to be orthogonal to V_k; the harmonic Ritz values approach asks for vectors r to be orthogonal to $(A - \sigma B)V_k$. Let W_k denote $(A - \sigma B)V_k$, then we have

$$\begin{aligned} r &= AV_k s - \theta BV_k s \\ &= (A - \sigma B)V_k s - (\theta - \sigma)B(A - \sigma B)^{-1}(A - \sigma B)V_k s \\ &= W_k s - (\theta - \sigma)B(A - \sigma B)^{-1}W_k s. \end{aligned} \tag{5}$$

Obviously, $\nu = \frac{1}{(\theta-\sigma)}$ is a Ritz value of the matrix $B(A - \sigma B)^{-1}$ with respect to W_k. To obtain eigenvalues in the neighbourhood of σ, ν must lie in the dominant spectrum of $B(A - \sigma B)^{-1}$. The orthogonalisation requirement leads to

$$\nu W_k^* W_k s = W_k^* BV_k s. \tag{6}$$

To obtain a standard eigenvalue problem we require $W_k^* W_k = I$. By introducing $C := (A - \sigma B)^*(A - \sigma B)$ this requirement gives

$$W_k^* W_k = V_k^*(A - \sigma B)^*(A - \sigma B)V_k = V_k^* C V_k = I \tag{7}$$

and we call V_k a C-orthonormal matrix.

The new search direction $\tilde{v}_k$ must be C-orthonormal to V_{k-1}, which implies that

$$V_{k-1}^* v_k = 0 \text{ and } \tilde{v}_k = \frac{v_k}{\|v_k\|_C} = \frac{v_k}{\|w_k\|_2}, \tag{8}$$

where $w_k = (A - \sigma B)v_k$.

To move from standard to harmonic Ritz values, the adjustments in the algorithm are not radical. In comparison to the original implementation, the harmonic case requires two extra matrix-vector multiplications and in addition extra memory to store an N_t-by-k matrix. The main difference is that the LU decomposition of $A - \sigma B$ is used as a preconditioner and not as a shift and invert technique.

3 Numerical Results

In this section, we show some results obtained on both an 80 processor Cray T3E situated at the HPαC centre in Delft, The Netherlands and a 512 processor Cray T3E at Cray Research, Eagan, MN, USA. The local memory per processor is at least 128 Mbytes. On these machines, the best results were obtained by a `MESSAGE PASSING` implementation using Cray intrinsic `SHMEM` routines for data transfer and communication. For more details, we refer to [2].

3.1 Problems

We have timed five MHD problems of the form (1). The *Alfvén* spectra of Problems **1**, **2** and **3**, on the one hand, and Problems **4** and **5**, on the other hand, do not correspond because different MHD equilibria have been used. For more details we refer to CASTOR [1]. The choices of the acceptance criteria will be explained in the next section.

1 A small problem of $N = 64$ diagonal blocks of size $n = 48$. We look for eigenvalues in the neighbourhood of $\sigma = (-0.08, 0.60)$, and stop after 10 eigenpairs have been found with $tol_{sJD} = 10^{-8}$ and $tol_{hJD} = 10^{-6}$. The experiments have been performed on $p = 8$ processors.

2 The size of this problem is four times as big as that of the previous problem; $N = 128$ and $n = 96$. Again, we look for eigenvalues in the neighbourhood of $\sigma = (-0.08, 0.60)$, and stop after 10 eigenpairs have been found with $tol_{sJD} = 10^{-8}$ and $tol_{hJD} = 10^{-6}$. The experiments have been performed on $p = 8$ processors.

3 The same as Problem **2**, but performed on $p = 32$ processors.

4 The size of this large problem is: $N = 256$ and $n = 256$. We took $\sigma = (-0.15, 0.15)$ and look for $N_{ev} = 12$ eigenpairs with $tol_{sJD} = 10^{-8}$ and $tol_{hJD} = 10^{-5}$. The experiments are performed on $p = 128$ processors.

5 The size of this very large problem is: $N = 4096$ and $n = 64$, we took $\sigma = (-0.10, 0.23)$ leading to another branch in the *Alfvén spectrum.* Now, we look for $N_{ev} = 20$ eigenpairs with $tol_{sJD} = 10^{-8}$ and $\tilde{tol}_{hJD} = 10^{-5}$. For this problem a slightly different acceptance criterion has been applied:

$$\|r\|_2 < \tilde{tol}_{hJD}.|\sigma + \frac{1}{\nu}|.\|u\|_2. \tag{9}$$

For the harmonic case, the 2-norm of u can be very large, about 10^6, so the results can be compared with $tol_{hJD} = 10^{-6}$. At present, we prefer to control the residue as described in Section 3.2. Figure 1 shows the distribution of 20 eigenvalues in the neighbourhood of $\sigma = (-0.10, 0.23)$.

3.2 Acceptance Criterion

For the standard approach we accept an eigenpair $(\sigma + \frac{1}{\nu}, u)$ if the residual vector satisfies:

$$\|r\|_2 = \|(Q - \nu I)u\|_2 < tol_{sJD}.|\nu|, \quad \text{with} \quad \|u\|_2 = 1 \tag{10}$$

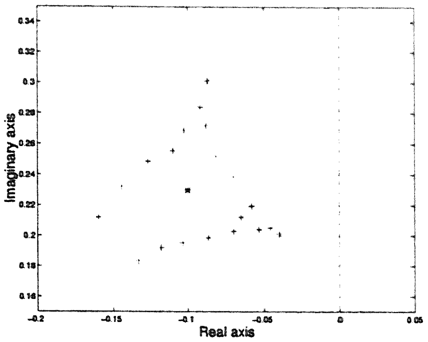

Fig. 1. The eigenvalue distribution of problem 5

and for the harmonic approach we require:

$$\|r\|_2 = \|(A - (\sigma + \frac{1}{\nu}))Bu\|_2 < tol_{hJD}.|\sigma + \frac{1}{\nu}|, \text{ with } \|u\|_C = 1. \tag{11}$$

To compare both eigenvalue solvers it is not advisable to choose the tolerance parameters tol_{sJD} equal to tol_{hJD} in (10) and (11), respectively. There are two reasons to take different values: firstly, within the same number of iterations the standard approach will result into more eigenpair solutions that satisfy (10) than into solutions that satisfy (11). Secondly, if we compute for each *accepted* eigenpair (λ, u) the true normalised residue γ defined by

$$\gamma := \frac{\|(A - \lambda B)u\|_2}{|\lambda|.\|u\|_2}, \tag{12}$$

then we see that the harmonic approach leads to much smaller γ values.

In Figure 2, the convergence behaviour of both the standard and harmonic approach is displayed, with and without restarts. A $\circ$ indicates that the eigenpair satisfies (10) or (11), a $\times$ denotes the γ value. We observe that the accuracy for the eigenpairs achieved by means of harmonic Ritz values is better than suggested by tol_{hJD}. On the other hand, tol_{sJD} seems to be too optimistic about the accuracy compared to the γ values shown in Figure 2. In our experiments we took $tol_{sJD} = 10^{-8}$ and $tol_{hJD} = 10^{-6}$ and $tol_{hJD} = 10^{-5}$. It is not yet clear to us how these parameters depend on the problem size or the choice of the target.

3.3 Restarting Strategy

The algorithm has two parameters that control the size of the projected system: k_{min} and m. During each restart, the k_{min} eigenvalues with maximal norm and not included in the set of accepted eigenvalues, that correspond to the k_{min} most promising search directions are maintained. Moreover, since an implicit deflation technique is applied in our implementation, the n_{ev} eigenpairs found so far are

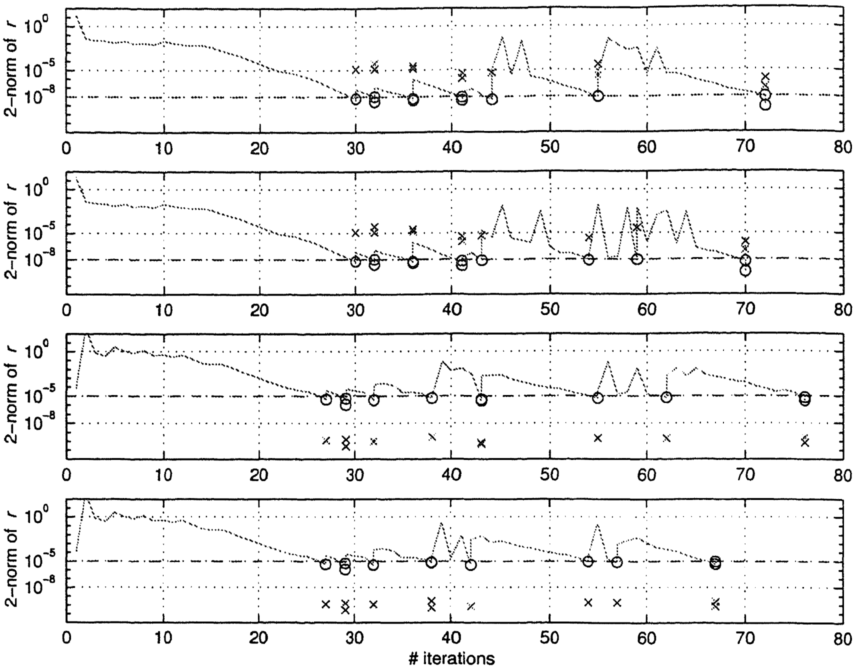

Fig. 2. The two upper plots result on problem 4 using standard Ritz values, the lower two on the same problem but using harmonic Ritz values. The first and third one show the convergence behaviour of Jacobi-Davidson restarting each time when the size of the projected system reaches $m = 37$, where $k_{min} = 25$ and $k_{min} = 20$, respectively. The second and fourth plots demonstrate the convergence in case of no restarts. The process ended when $N_{ev} = 12$ eigenvalues were found. It may happen that two eigenvalues are found within the same iteration step.

kept in the system too. The maximum size m should be larger than $k_{min} + N_{ev}$, where N_{ev} denotes the number of eigenvalues we are looking for. The influence of several (k_{min}, m) parameter combinations on both the parallel performance and convergence behaviour is studied.

3.4 Timing Results of (k_{min}, m) Parameter Combinations

For each experiment we take m constant and for k_{min} we choose the values $5, 10, \cdots, m - N_{ev}$. In Figures 4, 5, 6 and 7, the results of a single m value have been connected by a dashed or dotted line. Experiments with several m values have been performed. In the plots we only show the most interesting m values; m reaches its maximum if N_{ev} eigenpairs were found without using a restart. In the pictures this is indicated by a solid horizontal line, which is of course independent of k_{min}. If the number of iterations equals 80 and besides less than N_{ev} eigenpairs have been found, we consider the result as negative. This implies that, although the execution time is low, this experiment cannot be a candidate for the best (k_{min}, m) combination.

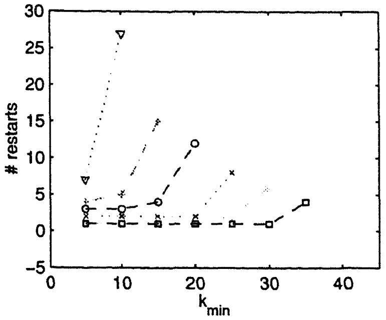

Fig. 3. The number of restarts needed to compute N_{ev} eigenvalues of Problem **2**. Results are shown for different m values: $m = 20$ ($\nabla \cdots$), $m = 25$ ($+ - \cdot$ line), $m = 30$ ($\circ - -$ line), $m = 35$ ($\times \cdots$ line), $m = 40$ ($\triangleright - \cdot$ line), $m = 45$ ($\square - -$ line).

Before we describe the experiments illustrated by Figures 4, 5, 6 and 7 we make some general remarks:

- We observed that if a (k_{min}, m) parameter combination is optimal on p processors, it is optimal on q processors too, with $p \neq q$.
- For k_{min} small, for instance $k_{min} = 5$ or 10, probably too much information is thrown away, leading to a considerable increase of iteration steps.
- For k_{min} large the number of restarts will be large at the end of the process; suppose that in the extreme case, $k_{min} = m - N_{ev}$, already $N_{ev} - 1$ eigenpairs have been found, then after a restart k becomes $k_{min} + N_{ev} - 1 = m - 1$. In other words, each step will require a restart. In Figure 3, the number of restarts is displayed corresponding to the results of Problem **2** obtained with harmonic Ritz values.
- The number of iterations is almost independent of the number of processors involved; it may happen that an increase of the number of processors causes a decrease by one or two iterations under the same conditions, because the LU decomposition becomes more accurate if the number of cyclic reduction steps increases at the cost of the domain decomposition part.

The first example (Figure 4) explicitly shows that the restarting technique can help to reduce the wall clock time for both the standard and harmonic method. The minimum number of iterations to compute 10 eigenvalues in the neighbourhood of σ is achieved in case of no restarts, viz, 53 for the standard case, 51 for the harmonic case. The least time to compute 10 eigenvalues is attained for $k_{min} = 15$ and $m = 30, 35$, but also for $k_{min} = 10$ and $m = 30, 35$ and $m = 40$ and $k_{min} = 15, 20, 25$ leads to a reduction in wall clock time of about 15 %. The harmonic approach leads to comparable results: for $(k_{min}, m) = (15, 30 : 35)$, but also $(k_{min}, m) = (10, 30 : 35)$ and $(k_{min}, m) = (15 : 25, 40)$ a reasonable reduction in time is achieved. The score for $k_{min} = 5$ in combination with $m = 35$ is striking, the unexpected small number of iterations in combination with a small k_{min} results into a fast time.

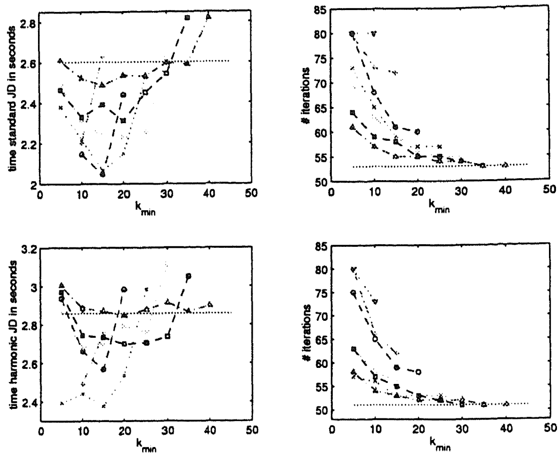

Fig. 4. The upper pictures result on problem 1 using standard Ritz values. The lower pictures result on the same problem with harmonic Ritz values. Results are shown for different m values: $m = 20$ ($\nabla\cdots$), $m = 25$ ($+ - \cdot$ line), $m = 30$ ($\circ - -$ line), $m = 35$ ($\times \cdots$ line), $m = 40$ ($\triangleright - \cdot$ line), $m = 45$ ($\square - -$ line), $m = 50$ ($\triangle - \cdot$ line). The solid lines give the value for no restart.

The plots in Figure 5 with the timing results for the Jacobi-Davidson process for Problem 2 give a totally different view. There is no doubt of benefit from restarting, although the numbers of iterations pretty well correspond with those of Problem 1. This can be explained as follows: the size of the projected system k is proportionally much smaller compared to N_t/p than in case of Problem 1; both the block size and the number of diagonal blocks is twice as big. For Problem 1 the sequential part amounts 45% and 36% of the total wall clock time, respectively, for the standard and harmonic Ritz values. For Problem 2 these values are 10.5% and 8%, respectively. These percentages hold for the most expensive sequential case of no restarts. The increase of JD iterations due to several restarts can not be compensated by a reduction of serial time by keeping the projected system small.

When we increase the number of active processors by a factor 4, as is done in Problem 4 (see Figure 6), we observe that again a reduction in wall clock time can be achieved by using a well-chosen (k_{min}, m) combination. The number of iterations slightly differ from those given in Figure 5, but the pictures with the Jacobi-Davidson times look similar to those in Figure 5. If we should have enlarged N by a factor of 4 and left the block size unchanged, we may expect execution times as in Figure 5.

For Problem 4, the limit of 80 iterations seems to be very critical. The right-hand plots of Figure 7 demonstrate that the number of iterations does not decrease monotonically when k_{min} increases for a fixed value m as holds for the previous problems. Moreover, it may happen that for some (k_{min}, m) combination, the limit of JD iterations is too strictly, while for both a smaller and larger

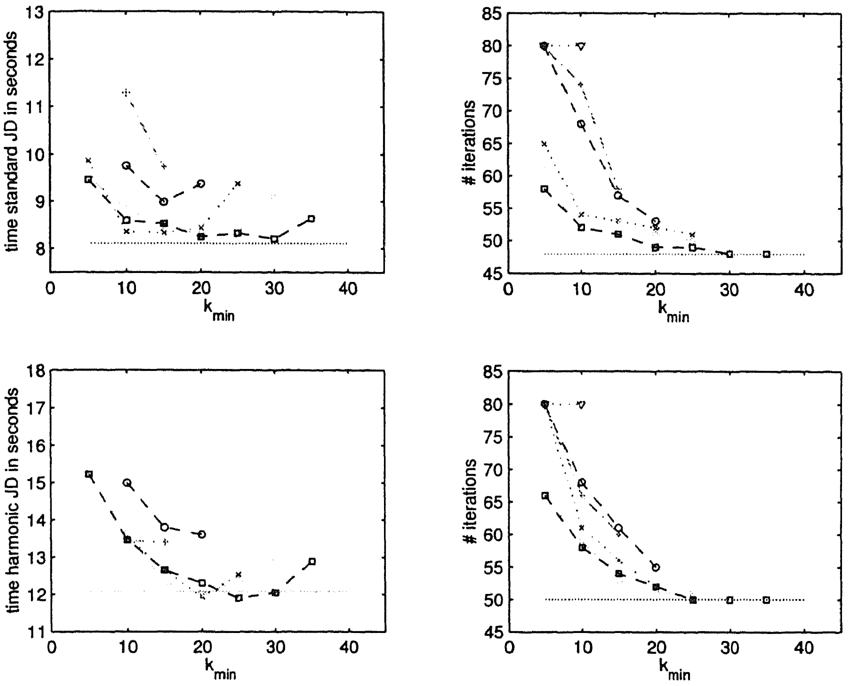

Fig. 5. The upper pictures result on problem **2** using standard Ritz values. The lower pictures result on the same problem with harmonic Ritz values. Results are shown for different m values: $m = 20$ ($\bigtriangledown \cdots$), $m = 25$ ($+ - \cdot$ line), $m = 30$ ($\circ - -$ line), $m = 35$ ($\times \cdots$ line), $m = 40$ ($\triangleright - \cdot$ line), $m = 45$ ($\square - -$ line). The solid lines give the value for no restart.

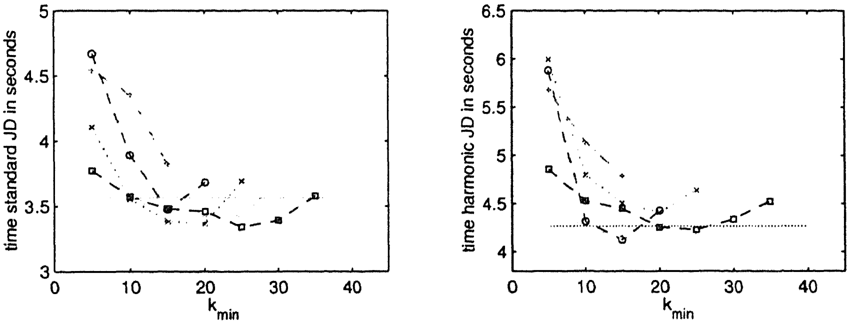

Fig. 6. The left pictures results on problem **3** using standard Ritz values. The right pictures result on the same problem with harmonic Ritz values. Results are shown for different m values: $m = 20$ ($\bigtriangledown \cdots$), $m = 25$ ($+ - \cdot$ line), $m = 30$ ($\circ - -$ line), $m = 35$ ($\times \cdots$ line), $m = 40$ ($\triangleright - \cdot$ line), $m = 45$ ($\square - -$ line). The solid lines give the value for no restart.

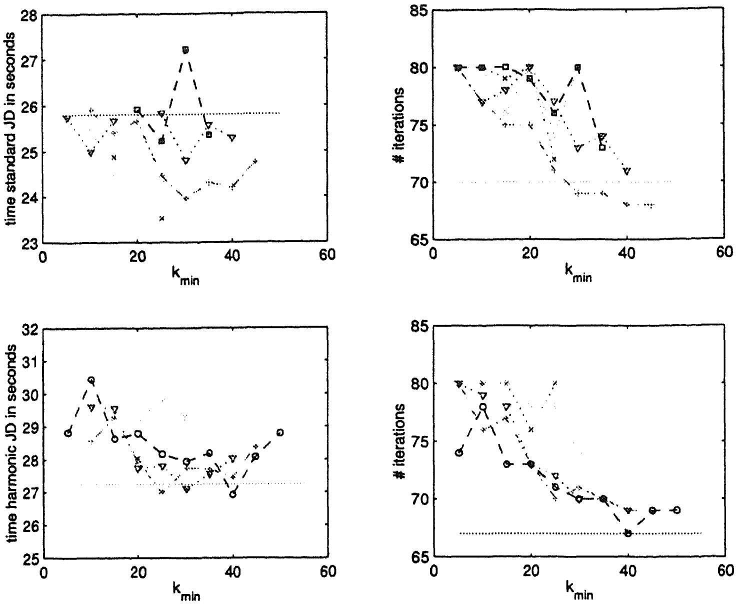

Fig. 7. The upper pictures result on problem 4 using standard Ritz values. The lower pictures result on the same Problem with harmonic Ritz values. Results are shown for different m values: $m = 37$ ($\times \cdots$ line), $m = 42$ ($\triangleright - \cdot$ line), $m = 47$ ($\square - -$ line), $m = 52$ ($\triangledown - \cdot$ line), $m = 57$ ($+ - \cdot$ line), $m = 62$ ($\circ - -$ line). The solid lines give the value for no restart.

k_{min} value the desired N_{ev} eigenpairs were easily found. In the left-hand plots only those results are included, which generate 12 eigenvalues within 80 iterations. Apparently, for the standard case with $m = 57$ and $30 \leq k_{min} \leq 45$, even less iterations are required than in case of no restarts. Of course, this will lead to a time which is far better than for the no-restart case. For the harmonic approach the behaviour of the number of JD steps is less obvious, but also here the monotonicity is lost. Execution times become unpredictable and the conclusion must be that it is better not to restart.

3.5 Parallel Execution Timing Results

Table 1 shows the execution times of several parts of the Jacobi-Davidson algorithm on the Cray T3E; the numbers in parentheses show the Gflop-rates. We took

$$N_{ev} = 20;\ tol_{sJD} = 10^{-8};\ tol_{hJD} = 10^{-5};\ k_{min} = 10;\ m = 30{+}N_{ev};\ it_{SOL} = 0.$$

The number of eigenvalues found slightly depends on the number of processors involved: about 11 for the standard and 13 for the harmonic approach within 80 iterations.

Table 1. Wall clock times in seconds for the standard and harmonic Ritz approach. $N = 4096$, $n = 64$.

p	Preprocessing	Time *standard* JD	Time *harmonic* JD	Triangular solves
32	7.90 (6.75)	64.59	88.61	25.56 (2.08)
64	4.08 (13.21)	31.70	43.78	13.28 (4.02)
128	2.19 (24.78)	15.07	21.33	7.28 (7.36)
256	1.27 (42.69)	8.55	11.48	4.36 (12.29)
512	0.84 (64.65)	5.64	7.02	3.01 (17.81)

The construction of L and U is a very time-consuming part of the algorithm. However, with a well-chosen target σ ten up to twenty eigenvalues can be found within 80 iterations. Hence, the life-time of a (L, U) pair is about 80 iterations. On account of the cyclic reduction part of the LU factorisation, a process that starts on all processors, while at each step half of the active processors becomes idle, we may not expect linear speed-up. The fact that the parallel performance of DDCR is quite good is caused by the domain decomposition part of the LU. For more details we refer to [2, 5].

About 40% of the execution time is spent by the computation of the LU factorisation (in Table 1 'Preprocessing'), which does not depend on the number of processors. The storage demands for Problem **5** are so large that at least the memories of 32 processors are necessary. DDCR is an order $\mathcal{O}(Nn^3)$ process performed by Level 3 BLAS and it needs less communication: only sub- and super diagonal blocks of size n-by-n must be transfered. As a consequence, for the construction of L and U, the communication time can be neglected also due to the fast communication between processors on the Cray T3E. The Gflop-rates attained for the construction of the LU are impressively high just like its parallel speed-up.

The application of L and U, consisting of two triangular solves, is the most expensive component of the JD process after preprocessing. It parallelises well, but its speed is much lower, because it is built up of Level 2 BLAS operations. The wall clock times for *standard* and *harmonic* JD are given including the time spent on the triangular solves. Obviously, a harmonic iteration step is more expensive than a standard step, but the overhead becomes less when more processors are used, because the extra operations parallelise very well.

4 Conclusions

We have examined the convergence behaviour of two Jacobi-Davidson variants, one using standard Ritz values, the other one harmonic Ritz values. For the kind of eigenvalue problems we are interested in, arising from Magneto-Hydrodynamics, both methods converge very fast and parallelise pretty well. With $tol_{sJD} = 10^{-8}$ and $tol_{hJD} = 10^{-5}$ in the acceptance criteria (10) and

(11), respectively, both variants give about the same amount of eigenpairs. The harmonic variant is about 20% more expensive, but results in more accurate eigenpairs. With a well-chosen target ten up to twenty eigenvalues can be found. Even for very large problems, $N_t = 65,536$ and $N_t = 262,144$, we obtain more than 10 sufficiently accurate eigenpairs in a few seconds.

Special attention has been paid to a restarting technique. The (k_{min}, m) parameter combination prescribes the amount of information that remains in the system after a restart and the maximum size of the projected system. In this paper we have demonstrated that k_{min} may not be too small. because then too much information gets lost. On the other hand, too large k_{min} values lead to many restarts and become expensive in execution time. In general, the number of iterations decreases when m increases. It depends on the N_t/p value, as we have shown, whether restarts lead to a reduction in the wall clock time for the Jacobi-Davidson process.

Acknowledgements

The authors wish to thank Herman te Riele for many stimulating discussions and suggestions for improving the presentation of the paper. They gratefully acknowledge HPαC (Delft, The Netherlands) for their technical support, and Cray Research for a sponsored account on the Cray T3E (Eagan, MN, USA), and the Dutch National Computing Facilities Foundation NCF for the provision of computer time on the Cray C90 and the Cray T3E.

References

1. W. Kerner, S. Poedts, J.P. Goedbloed, G.T.A. Huysmans, B. Keegan, and E. Schwartz. Computing the damping and destabilization of global Alfvén waves in tokamaks. In P. Bachman and D.C. Robinson, editors, *Proceedings of 18th Conference on Controlled Fusion and Plasma Physics.* EPS: Berlin, 1991. IV.89-IV.92.
2. Margreet Nool and Auke van der Ploeg. A Parallel Jacobi-Davidson Method for solving Generalized Eigenvalue Problems in linear Magnetohydrodynamics. Technical Report NM-R9733, CWI, Amsterdam, December 1997.
3. 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*, 36:595–633, 1996.
4. 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(2):401–425, april 1996.
5. A. van der Ploeg. Reordering Strategies and LU-decomposition of Block Tridiagonal Matrices for Parallel Processing. Technical Report NM-R9618, CWI, Amsterdam, October 1996.