IMPROVING THE CONVERGENCE OF THE JACOBI-DAVIDSON ALGORITHM

E. DE STURLER*

Abstract. We propose solutions to two convergence problems that may occur in the Jacobi-Davidson algorithm [13].

The first problem arises when small perturbations of the projection of the matrix to the search space (the projected matrix) create spurious eigenvalues close to the target eigenvalue. This makes the corresponding eigenvector ill-conditioned and the algorithm stagnates or converges very slowly. We discuss several causes for this problem. One potential remedy is to use refined Ritz vectors [8, 9]; however, for the Jacobi-Davidson method this solution generally is very expensive. We will propose a much cheaper solution.

The second problem occurs if the correction equation solved in the Jacobi-Davidson algorithm produces a solution that makes a small angle with the current search space. In this case we do not have an effective extension to the search space, and again the algorithm tends to converge very slowly. We propose a solution to this problem that also improves the convergence of the linear systems that must be solved in the algorithm.

Key words. large, sparse eigenvalue problems, Jacobi-Davidson method, truncation, non-Hermitian linear systems

AMS subject classifications. Primary 65F15; Secondary 65F50, 15A18

1. Introduction. The Jacobi-Davidson algorithm to compute selected eigenvalues and -vectors of a matrix was introduced in [13]. For completeness, we briefly describe the basic idea.

We want to solve the linear eigenvalue problem $Ax = \lambda x$ for one (or more) eigenvalues close to a given target $\hat{\lambda}$. Assume we have already computed a matrix V_k with k orthonormal columns that span the search space over which we approximate the desired eigenpair. Let $H_k = V_k^* A V_k$ and let (θ, s) be an eigenpair of H_k. Then the approximate eigenvalue θ and eigenvector $u = V_k s$ form a so-called Ritz pair. The residual is given by $r = Au - \theta u$, and by definition we have $r \perp V_k$. Obviously H_k has up to k eigenpairs. We select the one with the eigenvalue closest to the desired eigenvalue. In the next iteration, to improve our approximate eigenpair (θ, u), we solve the following correction equation (see [13]).

$$(I - uu^*)(A - \theta I)(I - uu^*)t = -r, \qquad t \perp u. \tag{1.1}$$

Afterwards, the solution t is orthogonalized against V_k and normalized to compute an orthogonal extension to the space range(V_k). Then we set column $k+1$ of V to t: $V_{k+1} = [V_k \ t]$. The correction equation (1.1) is typically solved only to low accuracy by an iterative method. In their paper the authors use GMRES, but other methods may be used as well. The algorithm is continued until the norm of the residual satisfies some preset tolerance ($\|r\|_2 \leq tol$).

The Jacobi-Davidson algorithm has recently gained wide popularity and the method and generalizations have been used to solve several hard eigenvalue problems, see e.g., [1, 12, 16, 15],

In this paper we focus on the two problems mentioned in the abstract. The purpose of this paper is not to compare the Jacobi-Davidson algorithm with other

*Department of Computer Science, University of Illinois at Urbana-Champaign (sturler@uiuc.edu).

eigensolvers nor to introduce another eigensolver. Therefore, to analyze the convergence problems in as simple a setting as possible we will not use preconditioners. Even though we realize that this is one of the powerful features of the algorithm, see e.g. [15]. In addition, we select specific (but mostly realistic) problems to illustrate the convergence problems and to demonstrate the usefulness of our proposed solutions.

The first problem arises when small perturbations of the (projected) matrices H_k for (many) successive k create spurious eigenvalues close to the target eigenvalue. This makes the eigenvector ill-conditioned and the algorithm stagnates or converges very slowly. One potential remedy is to use refined Ritz vectors [8, 9]; however, for the Jacobi-Davidson method this solution generally is expensive. The refined Ritz vector $\tilde{u}$ corresponding to θ is defined as the solution to

$$\tilde{u} = \arg\min_y \|(A - \theta I)V_k y\|_2. \tag{1.2}$$

If the columns of V_k span a Krylov space (but not an arbitrary subspace of a Krylov space), the residuals of all Ritz pairs are differently scaled copies of the same vector, say r. Therefore, $\text{range}((A - \theta I)V_k) \subseteq \text{range}([V_k \ r])$, and we can compute the refined Ritz vector from a small $(k+1) \times k$ matrix, cf. the Arnoldi method where we can compute it using the extended Hessenberg matrix [8]. However, the columns of V_k in the Jacobi-Davidson algorithm will generally not span a Krylov space. Note that even an arbitrary selection of vectors out of a Krylov sequence does not span a Krylov space. So we do not have the property that $\text{range}((A - \theta I)V_k) \subseteq \text{range}([V_k \ r])$. Indeed, for the Jacobi-Davidson algorithm the residuals of the Ritz pairs may all be independent. In fact, for the Jacobi-Davidson algorithm each basis vector might have been generated with a different preconditioner. So, computing the refined Ritz vector becomes expensive, because it requires the singular value decomposition (SVD) of an $N \times k$ matrix. Moreover, since θ changes from one iteration to the next, we have to compute a new SVD in each iteration.

We will propose a much cheaper solution. We will also show an example that using refined Ritz vectors in the Jacobi-Davidson algorithm may lead to slower convergence than using the Ritz vectors (Problem 3 in Section 4).

The second problem occurs if the approximate solution t to the correction equation (1.1) makes a small angle with the current search space $\text{range}(V_k)$, that is, $\|(I - V_k V_k^*)t\|_2 \ll \|t\|_2$. After orthogonalization t contains mainly noise, we do not compute an effective extension to the search space, and again the algorithm tends to converge very slowly. We propose a solution to this problem that also improves the convergence of the linear systems that must be solved in the algorithm.

In [4] we suggested the solutions to these problems. In the present paper we elaborate in detail the relevant theory, and we present several improvements and additions to the material in [4].

We discuss the relevant theory in Section 2 and our implementations in Section 3. Section 4 contains the numerical experiments, and Section 5 contains the conclusions.

2. Theory. Let $A \in \mathbb{C}^{N \times N}$, let $V_k^* V_k = I_k$, and let $H_k = V_k^* A V_k$. Further, we assume that the desired eigenvalue λ of A closest to the specified (input) target $\hat{\lambda}$ is simple, and we denote the (right) eigenvector corresponding to λ by x. Although we believe that the approaches discussed in this paper are applicable to higher dimensional invariant subspaces, we will not discuss this in the present paper. Let θ be the eigenvalue of H_k closest to λ and let (θ, s) denote the corresponding (right) eigenpair. The operator used in the Jacobi-Davidson correction equation (1.1) will be denoted

by

$$A_{u,\theta} = (I - uu^*)(A - \theta I)(I - uu^*).$$

Since the failure to converge of Ritz vectors is caused by the ill-conditioning of the eigenvectors s of the matrices H_k, for $k = 1, 2, \ldots$, we will need to look at the following matrices and singular value decompositions. Let $S_c \in \mathbb{C}^{k\times k-1}$, $S_c^* S_c = I_{k-1}$, and $S_c \perp s$. So S_c provides an orthonormal basis for the complement of range(s), and we have

$$[s\ S_c]^* H_k [s\ S_c] = \begin{bmatrix} \theta & s^* H_k S_c \\ 0 & S_c^* H_k S_c \end{bmatrix}. \tag{2.1}$$

We define the following two singular value decompositions

$$S_c^* H_k S_c - \theta I = \Phi \Omega \Psi, \tag{2.2}$$

and

$$A_{u,\theta} V_k S_c = \tilde{\Phi} \tilde{\Omega} \tilde{\Psi}. \tag{2.3}$$

We are concerned with the convergence of the Jacobi-Davidson algorithm to some given tolerance *tol*. Hence, we will discuss ill-conditioning relative to the tolerance we are trying to achieve. To compute accurate approximations to the desired eigenpair we are concerned with the conditioning of the matrix $(S_c^* H_k S_c - \theta I)$ and, in the case of small eigenvalues, with the conditioning of the matrix H_k.

2.1. Spurious Eigenvalues. As discussed in [8, 9] the Ritz vectors u may not converge to the eigenvector x until the search space spans the entire space $\mathbb{C}^n$, even though the actual eigenvector x can be approximated very accurately in smaller (previous) subspaces. When this happens, the reason for this failure to converge is that there exists a small perturbation of the projected matrix H_k that makes its eigenvalue θ a double eigenvalue. As a result the conditioning of the eigenvector s corresponding to θ becomes so poor that $u = V_k s$ fails to converge. This may happen even if A has no other eigenvalue (relatively) close to the desired eigenvalue λ and the corresponding eigenvector x is not very ill-conditioned, as we will show in Section 4.

In order to derive an alternative to refined Ritz vectors, we will analyze the problem of ill-conditioning of the eigenvector s of H_k. If s is significantly more ill-conditioned than x, this must be an artifact of the subspace over which we are trying to approximate (λ, x). Therefore, we propose to improve the search space by purging subspaces that cause this *artificial* ill-conditioning.

THEOREM 2.1. *There exists a perturbation E of H_k with $\|E\|_2 = \omega_{k-1}$ such that θ is a double eigenvalue of $H_k + E$.*

Proof. From (2.2) we have $\|(S_c^* H_k S_c - \theta I)\psi_{k-1}\|_2 = \omega_{k-1}$. In this case, the Kahan-Parlett-Jiang theorem [10, 11] states that there exists a perturbation $\widehat{E}$ of $S_c^* H_k S_c$ such that $\|\widehat{E}\|_2 = \omega_{k-1}$ and $(S_c^* H_k S_c + \widehat{E})\psi_{k-1} = \theta\psi_{k-1}$. So θ is an eigenvalue of $S_c^* H_k S_c + \widehat{E}$. Then, from (2.1) we can conclude that θ is a double eigenvalue of $H_k + E$, where

$$E = [s\ S_c] \begin{bmatrix} 0 & 0 \\ 0 & \widehat{E} \end{bmatrix} [s\ S_c]^*.$$

Obviously, $\|E\|_2 = \|\widehat{E}\|_2 = \omega_{k-1}$. $\square$

This theorem shows that if ω_{k-1} is very small, the eigenvector s corresponding to θ is ill-conditioned. In particular, if ω_{k-1}/ω_1 is very small, there exists a small relative perturbation of H_k that makes θ a double eigenvalue, and we may not expect that s will be computed to high accuracy. Clearly, this will lead to convergence problems if the ill-conditioning persists. It is important to realize that this does not necessarily imply that $S_c^* H_k S_c$ has an eigenvalue close to θ, some eigenvalues of $S_c^* H_k S_c$ may be very ill-conditioned. If we use the QR-algorithm to compute the eigenvalue decomposition of H_k, then we have for the computed Schur form of the matrix H_k that $P\widehat{T}P^* = H_k + E$, where $\|E\|_2 \approx \varepsilon_{mach}\|H_k\|_2$ [7]. Typically, $\|H_k\|_2 \gtrsim \omega_1$, so we can expect perturbations of order $\omega_1 \varepsilon_{mach}$.

The following theorem, adapted from [9], shows under which conditions the Ritz vector $u = V_k s$ does converge to x.

THEOREM 2.2. *If ε exists such that*

$$\omega_{k-1} \equiv \sigma_{min}(S_c^* H_k S_c - \theta I) \geq \varepsilon > 0, \tag{2.4}$$

for $k = 1, 2, \ldots$, then the Ritz vector $V_k s$ corresponding to θ converges.

Proof. See [9], theorem 3.2 and subsequent discussion. □

It was also shown in [9] that the Ritz value θ converges unconditionally as the angle between the vector x and the search space goes to zero. So, if we can adapt the sequence of search spaces such that they produce the same converging Ritz value θ, but the corresponding eigenvector s remains well-conditioned (ω_{k-1} uniformly bounded away from 0) the Ritz vector $u = V_k s$ will converge to x. This is exactly what we propose to do.

We want the computed Ritz vector to converge within the required tolerance *tol*. Therefore, as a matter of practical concern, we must prevent that s becomes so ill-conditioned that the inaccuracy in the computation of s prevents us from reaching this tolerance. In the following, we will derive a way to adapt the sequence of search spaces spanned by V_k, for $k = 1, 2, \ldots$, such that s remains well-conditioned. From the theorems above, we already see that we want to keep ω_1/ω_{k-1} smaller than a certain tolerance related to *tol*. We will derive this tolerance criterion below.

First, however, we will consider where the ill-conditioning of s may come from, and why the Jacobi-Davidson algorithm may be susceptible to this problem. The most important reason for the potential ill-conditioning of s is that $A_{u,\theta}$ may be ill-conditioned over range$(V_k S_c)$.

THEOREM 2.3. *If $A_{u,\theta}$ is ill-conditioned over* range$(V_k S_c)$ *and $\omega_1 \approx \tilde{\omega}_1$ then there exists a small relative perturbation of H_k such θ is a double eigenvalue.*

Proof. Since $s \perp S_c$ we have

$$(V_k S_c)^* A_{u,\theta} (V_k S_c) = S_c^* H_k S_c - \theta I = \Phi \Omega \Psi.$$

Clearly $\omega_{k-1} \leq \tilde{\omega}_{k-1}$. So from $\omega_1 \approx \tilde{\omega}_1$ we may infer $\omega_1/\omega_{k-1} \lesssim \tilde{\omega}_1/\tilde{\omega}_{k-1}$. Therefore, $S_c^* H_k S_c - \theta I$ is ill-conditioned. According to Theorem 2.1 this means a small relative perturbation exists that makes θ a double eigenvalue. □

The problem that $A_{u,\theta}$ is ill-conditioned over range$(V_k S_c)$ tends to arise for a variety of reasons. The matrix A may be very ill-conditioned (which does not mean the eigenvalue x is ill-conditioned). The eigenvalue approximation θ is much more accurate than the eigenvector approximation u (which is quite common), or vice versa. We can improve the conditioning of s by purging those directions from the search space that cause the ill-conditioning. This is rather counterintuitive, as generally the approximation improves as we extend the search space. However, the problem

has little to do with the projection of the actual eigenvector x onto the search space range(V_k); it is caused by the projection of A onto the search space. Note that $S_c^* H_k S_c - \theta I = \Phi\Omega\Psi$ may also be ill-conditioned if there exists a vector y such that almost $A_{u,\theta}(V_k S_c)y \perp V_k S_c$, even if $\|A_{u,\theta}(V_k S_c)y\|_2$ is not small. However, this problem may well disappear in subsequent iterations if V_k is expanded.

The assumption in Theorem 2.3 that $\omega_1 \approx \tilde{\omega}_1$ excludes the possibility that we are lucky and $S_c^* H_k S_c - \theta I$ is much better conditioned than $A_{u,\theta}(V_k S_c)$. This could happen if $\sigma_{max}(S_c^* V_k^* A_{u,\theta} V_k S_c)$ is much smaller than $\tilde{\omega}_1$. However, this points us in the right direction for adapting V_k such that the eigenvector s remains well-conditioned.

Before we discuss how we purge those directions that cause the ill-conditioning from the search space, we discuss an additional, related convergence problem.

The previously discussed problem holds for any eigenvalue λ. However, if we are interested in an eigenpair with a small absolute eigenvalue, we will also have problems if H_k is ill-conditioned. If the condition number of H_k is large, this may lead to large relative errors in the computed smallest eigenvalues. This can be shown as follows. We have for the computed Schur form of H_k (from the QR algorithm) that $P\widehat{T}P^* = H_k + E$, where $\|E\|_2 \approx \epsilon_{mach}\|H_k\|_2$ [7]. In addition, we have from the Bauer-Fike theorem (see [7]) that if μ is an eigenvalue of $H_k + E$, where $H_k = X\Lambda X^{-1}$, then

$$min_{\lambda \in \lambda(H_k)} |\lambda - \mu| \leq \kappa_2(X)\|E\|_2,$$

where $\kappa_2(X)$ is the spectral condition number of the eigenvector matrix of H_k. So, the ill-conditioning of H_k can lead to a large relative shift of the smallest (absolute) eigenvalues of H_k, especially if $\kappa_2(X)$ is large as well. If we are interested especially in the smallest eigenvalues, this inaccuracy may prevent us from converging, even if the eigenpair itself can be well approximated in the Krylov space. It is the inaccuracy of the computed eigenvalues of the projected matrix that keeps us from converging. Typically, $\kappa_2(H_k)$ will be large when ω_1/ω_{k-1} is large. Especially in the case that $S_c^* H_k S_c - \theta I$ is ill-conditioned because its eigenvector matrix is ill-conditioned, rather than having an eigenvalue very close to θ. Specifically, if $\omega_1 \gg \theta$ then H_k is ill-conditioned. So if we try to approximate an eigenpair with an absolute small eigenvalue, we have reason to try to purge subspaces from range(V_k) that cause large singular values of H_k.

We will now discuss how we can compute more accurate approximations to x by adapting the search space range(V_k). As stated before, we are interested in computing an eigenpair approximation accurate to some given tolerance *tol*, that is we want

$$\|r\|_2 \equiv \|AV_k s - \theta V_k s\|_2 \leq tol. \tag{2.5}$$

Given a certain inaccuracy in s we have no direct way to assess the effect on $\|r\|_2$ (other than computing it). Of course, we do know that $\|H_k s - \theta s\|_2 \leq \|r\|_2$ (see below). Hence, we want to bound the inaccuracy in the computed s to be less than tol/γ, where $\gamma \geq 1$ provides some *margin*. Note that reducing ω_1/ω_{k-1} increases the relative perturbation that makes θ a double eigenvalue and hence controls the conditioning of s. Therefore, we want to truncate the search space such that ω_1/ω_{k-1} remains bounded. In particular we want to bound ω_{k-1} away from 0, so that Theorem 2.2 guarantees convergence. We will now derive the bound we want to maintain for ω_1/ω_{k-1} relative to *tol*.

From a theorem in [17, p.236] and its specialization to eigenvectors [17, p.240] we can compute the following error bound for the computed eigenvector $\tilde{s}$ of H_k with

respect to the exact eigenvector s for a perturbation (matrix) E.

$$\tilde{s} \approx s + S_c(\theta I - S_c^* H_k S_c)^{-1} S_c^* E s. \tag{2.6}$$

Moving s to the left, taking norms on both sides and assuming $\|E\|_2 \leq \|H_k\|\varepsilon_{mach}$, we get

$$\|\tilde{s} - s\|_2 \lesssim \frac{\|H_k\|_2}{\omega_{k-1}} \varepsilon_{mach}. \tag{2.7}$$

So in order to bound the error in s from (2.7) we have to truncate the subspace range(V_k) such that

$$\frac{\|H_k\|_2}{\omega_{k-1}} < \frac{tol}{\gamma \varepsilon_{mach}}. \tag{2.8}$$

In the case that the desired eigenvalue is not the absolute largest eigenvalue we may use the approximation $\omega_1 \approx \|H_k\|_2$, in which case the previous bound becomes

$$\frac{\omega_1}{\omega_{k-1}} < \frac{tol}{\gamma \varepsilon_{mach}}. \tag{2.9}$$

So optimizing the conditioning of s and minimizing the bound on the error in the computed $\tilde{s}$ in (2.7) require the same value to be bounded.

However, the actual error in s may be much smaller than indicated by the upper bound. Moreover, truncating too often or reducing the dimension of the subspace too much is expensive and may reduce the convergence rate. Hence, we typically apply an additional criterion. We assess the actual error and hence the *effect of ill-conditioning* by the norm of the residual of the eigenpair (θ, s) of H_k. If $\|H_k s - \theta s\|_2$ is very small, say

$$\|r_s\|_2 \equiv \|H_k s - \theta s\|_2 < tol/\gamma, \tag{2.10}$$

we may assume the ill-conditioning did not play a large role (and does not affect the accuracy of u significantly), and we may choose not truncate. Specific choices are discussed in Section 3; for the effect of certain choices see Section 4. There is an another important reason for using $\|r_s\|_2$. As will be shown in the numerical examples, if ω_1/ω_{k-1} gets sufficiently large $\|r_s\|_2$ may be large too. Since $r_s = V_k^* r$ we have

$$\|r_s\|_2 \leq \|r\|_2. \tag{2.11}$$

So, if $\|r_s\|_2 > tol$ the Ritz vector u cannot converge to the required tolerance. Occasionnally we may use further criteria in addition to those mentioned above.

Summarizing, when the bound on the conditioning of s and/or the bound on $\|r_s\|_2$ are violated we truncate the search space such the bound on $\omega_{max}/\omega_{min}$ is satisfied again. Clearly, we want to discard the subspace of smallest dimension such that

$$\frac{\omega_{max}}{\omega_{min}} < \frac{tol}{\gamma \varepsilon_{mach}}. \tag{2.12}$$

We can do this as follows. Let $\{n_1 \; n_1 + 1 \; \ldots \; n_2\}$ be the largest set of consecutive indices such that $\omega_{n_1}/\omega_{n_2} < \frac{tol}{\gamma \varepsilon_{mach}}$, where $\gamma \geq 1$ can be used to create a certain

margin for the accuracy of s (a typical choice is $\gamma = 100$). Then we define the new search space using S_c, the right singular vectors $\psi_{n_1}, \ldots, \psi_{n_2}$, and s. The new search space is represented by

$$\widehat{V}_{n_2-n_1+2} = V_k[s\ S_c\psi_{n_1}\ \ldots\ S_c\psi_{n_2}]. \tag{2.13}$$

In Section 3 we will discuss efficient ways to implement this truncation. The computation of the eigenvalue decomposition of H_k and the SVD of $S_c^* H_k S_c - \theta I$ introduces a cost of $O(k^3)$, where k is very small compared to N. In the next section we will show that the truncation (2.13) has a computational cost of $O(Nk)$, in contrast to $O(Nk^2)$ for computing refined Ritz vectors for the Jacobi-Davidson algorithm. An additional advantage is that this truncation may need to be done only once or a few times (see Section 4), while in general refined Ritz vectors must be computed at every iteration.

2.2. Ineffective extension to the search space. The second problem occurs if the solution t to the correction equation (1.1) makes a small angle with the current search space, that is, $\|(I - V_k V_k^*)t\|_2 \ll \|t\|_2$. Note that only $t \perp u$ is enforced by the algorithm, not $t \perp V_k$. Hence, if $\|(I - V_k V_k^*)t\|_2 \ll \|t\|_2$, after orthogonalization, the resulting vector, $(I - V_k V_k^*)t$ will contain mainly noise, and it will not give a *useful* extension to the subspace range(V_k). The Jacobi-Davidson algorithm then stagnates or converges very slowly. This can happen for a variety of reasons. First of all, if A is very ill-conditioned and/or strongly non-symmetric. Generating Krylov subspaces with such a matrix tends to generate spaces that make very small angles with each other, unless we enforce a certain level of independence by maintaining orthogonality to selected subspaces [3]. Another typical reason why t may be close to range(V_k) is slow convergence of the Jacobi-Davidson algorithm. If the Jacobi-Davidson algorithm at some stage converges very slowly, then r and u hardly change, and we solve virtually the same correction equation several times. Clearly, the solutions will be very close as well.

Following ideas from [3], we can reduce this problem by using the already generated subspace range(V_k) in the linear solver. This serves two purposes and both will alleviate the problem. We can generate a new search space explicitly orthogonal to V_k. The component of t in this search space always provides a good extension to the search space. If we use V_k in the linear solver to improve convergence we find better approximations to t (with fewer iterations) which generally will make the Jacobi-Davidson algorithm converge faster.

The extensions to the standard algorithm bring additional costs too. However, we can easily combine the standard algorithm with the extensions in the sense that we do not perform them at every step, but only when certain criteria are met. For example, when convergence becomes very slow, when the approximate solution of the correction equation (1.1) is almost dependent with the columns of V_k, or when the algorithm seems to stall even though we compute accurate solutions to the correction equation.

Approach 1. If we mainly aim to improve the linear solver, we apply the following strategy derived from [5, 2, 3].

We first compute the QR-decomposition

$$A_{u,\theta} V_k S_c = QR, \tag{2.14}$$

with $Q^*Q = I_{k-1}$, and $Q \perp u$. We now want to solve (1.1) over the union of range(V_k) and a small Krylov space $range(C_m)$ such that $C_m \perp Q$. The optimal way to do

this was discussed in [5, 2]. Assuming we use GMRES to approximately solve the correction equation (1.1), we do this as follows.

$$\rho = \| - r + QQ^* r\|_2, \tag{2.15}$$

$$c_1 = (-r + QQ^* r)/\rho. \tag{2.16}$$

Next we use the Arnoldi iteration with additional orthogonalization on Q to compute

$$A_{u,\theta} C_m = QQ^* A_{u,\theta} C_m + C_{m+1} \underline{G}_m, \tag{2.17}$$

so that $C_m \perp Q$ and $C_m \perp u$. In the following we use B to denote $Q^* A_{u,\theta} C_m$. Now we want the solution $t = V_k S_c y_1 + C_m y_2$ that minimizes the residual norm $\| - r - A_{u,\theta} t\|_2$. We need to find the vectors y_1 and y_2 that solve

$$y_1, y_2 = \arg\min_{\tilde{y}_1, \tilde{y}_2} \| - r - A_{u,\theta}(V_k S_c \tilde{y}_1 + C_m \tilde{y}_2)\|_2. \tag{2.18}$$

Now we substitute for $-r = -QQ^* r + \rho c_1$, and using (2.14) and (2.17) we get

$$y_1, y_2 = \arg\min_{\tilde{y}_1, \tilde{y}_2} \| - QQ^* r - QR\tilde{y}_1 - QB\tilde{y}_2 + \rho c_1 - C_{m+1} \underline{G}_m \tilde{y}_2)\|_2. \tag{2.19}$$

This problem can be solved in two minimization steps [2].

$$y_2 = \arg\min_{\tilde{y}_2} \|\rho c_1 - C_{m+1} \underline{G}_m \tilde{y}_2)\|_2, \tag{2.20}$$

$$y_1 = \arg\min_{\tilde{y}_1} \| - QQ^* r - QR\tilde{y}_1 - QB\tilde{y}_2\|_2, \tag{2.21}$$

where the second minimization reduces to a nonsingular linear system of equations after the solution of the first gives y_2 and if R is nonsingular. Note that the latter condition is automatically fulfilled if we use truncations as described in subsection 2.1, since the singular values of R are $\tilde{\omega}_1, \ldots, \tilde{\omega}_{k-1}$. The first minimization is solved as in the standard GMRES iteration. We can move the (orthogonal) matrix C_{m+1} outside the norm and solve the resulting upper Hessenberg system,

$$min\|e_1 \rho - \underline{G}_m \tilde{y}_2\|_2,$$

in least squares sense using $m - 1$ Givens rotations and back substitution.

This gives the solution $t = V_k S_c y_1 + C_m y_2$ that minimizes the residual norm of the correction equation. Moreover, t is automatically orthogonal to u. Furthermore, since we are interested in t only as an extension to the subspace range(V_k), we do not even need to compute y_1; we are only interested in the part $C_m y_2$.

However, we have taken C_m orthogonal to Q, not orthogonal to V_k, and so we have to orthogonalize $C_m y_2$ against V_k. Since C_m is orthogonal to $Q = A_{u,\theta} V_k S_c$, and we use V_k to approximate an invariant subspace, we do not expect $range(C_m)$ to be close to range(V_k). Unfortunately, occasionally when we orthogonalize the normalized $C_m y_2$ against V_k we do end up with a very small vector. If this persists we should use *Approach 2*, discussed below. However, in general this is not the case, and we find vectors $C_m y_2$ that make large angles with range(V_k).

So the algorithmic extensions outlined above serve two different purposes. First, they typically produce smaller residuals for the linear system of equations (1.1) than the standard Jacobi-Davidson algorithm. Second, they generally produce a new direction vector that has a larger angle with range(V_k) than the vector produced by the standard Jacobi-Davidson algorithm.

Finally, there is an interesting link between the truncation strategy discussed in the previous subsection and the problem of poor extensions to the search space. From (2.14) and (2.3) we have that the SVD of R is given by

$$R = Q^* \tilde{\Phi} \tilde{\Omega} \tilde{\Psi}. \tag{2.22}$$

This has following implication. If $A_{u,\theta}$ is ill-conditioned over the range$(V_k S_c)$, especially if $\tilde{\omega}_{k-1}$ is very small, then

$$y_1 = R^{-1}(-Q^* r - B y_2)$$

may be very large. This emphasizes the fact that t may have a large component in range$(V_k S_c)$, especially if $A_{u,\theta}$ is ill-conditioned over range$(V_k S_c)$. It also shows that the two problems we address are not unrelated. Since we are not interested in t but only in the component $C_m y_2$, our approach prevents the new vector from being spoiled by components in range$(V_k S_c)$ that we are not interested in.

Approach 2. As an alternative we may want to compute t to be explicitly orthogonal to V_k. This prevents the possibility of computing a vector t that is close to range(V_k). A straightforward approach would be to solve (1.1) using an orthogonal residual approach (like FOM) instead of a minimal residual approach, but this can occasionally lead to poor approximations. In fact the results of some experiments turned out rather disappointing. However, with some additional work we can solve in minimal residual sense for a general solution of the form $t = V_k S_c y_1 + C_m y_2$ with $C_m \perp V_k$: To generate the C_m, instead of separately forcing orthogonality to $V_k S_c$ and to $u = V_k s$, we rather maintain orthogonality to range(V_k) and iterate with $A_\theta = (A - \theta I)$. So, given the residual $r = Au - \theta u$, we proceed as follows. Let $c_1 = r/\|r\|_2$. We use m Arnoldi iterations with additional orthogonalization on V_k to compute

$$A_\theta C_m = V_k V_k^* A_\theta C_m + C_{m+1} \underline{G}_m. \tag{2.23}$$

Note that $C_m \perp V_k$. Taking the generic solution $t = V_k S_c y_1 + C_m y_2$ and solving in least-squares sense (minimum residual) we get the equations

$$-r - A_\theta (V_k S_c y_1 + C_m y_2) \perp range(A_\theta [V_k S_c \; C_m]). \tag{2.24}$$

The system of normal equations for this problem tends to be very ill-conditioned, so these equations are best solved using a QR-decomposition.

In general, it seems the second approach produces less good results than the first. Since, in none of our experiments using the first approach we had problems with t having a small angle with range(V_k) we will not show any results with *Approach 2*.

Remark. In [6] the authors propose to compute an extension to the search space by solving a modified correction equation using

$$(I - V_k V_k^*)(A - \theta I)(I - V_k V_k^*) t = -r.$$

This turns out not to work well [6]. Although this approach seems similar to ours, there are important differences. In *Approach 1* we extend the search space orthogonal to $A_{u,\theta} V_k S_c$ which seems to be better for the linear solver. In *Approach 2* we do extend the search space orthogonal to V_k. However, in both approaches we compute *solutions to the original correction equation (1.1)* and we compute *optimal solutions over the spaces* range$(A_{u,\theta} V_k S_c) \bigoplus$ range(C_m) *and* range$(V_k S_c) \bigoplus$ range(C_m) *respectively.* So we use $A_{u,\theta} V_k S_c$ and V_k to generate more independent search spaces, but we solve the original correction equation. This seems to make a significant difference.

3. Implementation. We follow the basic algorithm as specified in [13]. However, we extend it with a few additional steps to improve convergence and with a truncation strategy to make sure that the eigenvector of the desired eigenpair of H_k remains sufficiently well-conditioned.

Let the columns of the matrix V_k span the search space that we use for computing an approximate eigenpair, and let $W_k = AV_k$ and $H_k = V_k^* W_k$. Furthermore, let (θ, s) be the eigenpair of H_k with θ closest to the desired eigenvalue. Then we have the Ritz pair (θ, u) where $u = V_k s$. To extend the search space and find a better approximation to the desired eigenpair (λ, x) we approximately solve the correction equation

$$A_{u,\theta} t = -r,$$

subject to the constraint that t be orthogonal to u.

In order to deal with spurious eigenvalues close to θ, or more generally with ill-conditioning of $S_c^* H_k S_c - \theta I$ and s we make the following tests before an iteration (if $k > 1$).

1. We compute (2.2) and we test whether (2.9) is satified.
2. We compute $\|r_s\|_2$ and we test whether (2.10) is satisfied. If not, and if our desired eigenvalue is the largest absolute eigenvalue, we may also compute $\|H_k\|_2$ and test whether (2.8) is satisfied. Note that if (2.9) and/or (2.8) are satisfied there is no need to check (2.10) unless we are willing to change the tolerance margin γ. If we do not change γ, even if (2.10) would be violated, no truncation will be done.
3. Occasionally, it appears to be advantageous to check $\|s^* H_k S_c\|_2 / \|S_c^* H_k S_c\|_2$ and not to truncate if this number is (very) small. This number indicates a relative uncoupling of θ from the eigenvalues of $S_c^* H_k S_c$, see [17, pp. 232 & ff.].

Finally, we note that maintaining (2.9) and/or (2.8) (in practice) always takes care of maintaining (2.4). Our general truncation strategy is to truncate according to (2.12) if (2.9) and (2.10) are violated. Using the third criterion above regularly improves convergence but not much.

A straightforward implementation of the truncation (2.13) is expensive if we purge a subspace of small dimension. This is generally the case; the dimension of the subspace purged is typically 1 or 2. Therefore, the truncation is implemented using a trick from [3]. We do not care about the actual columns of the new matrix $\widehat{V}_{n_2-n_1+2}$; we only need

$$\text{range}(\widehat{V}_{n_2-n_1+2}) = \text{range}([V_k s \; V_k S_c \psi_{n_1} \; V_k S_c \psi_{n_1+1} \; \ldots \; V_k S_c \psi_{n_2}]).$$

Therefore, we use Givens rotations to rotate the columns of V_k in such a way that the vectors to be purged are rotated to the last columns of V_k. Then we adapt k to reflect the dimension of the new search space, and the vectors are effectively discarded. They will be overwritten in subsequent iterations. Suppose we want to discard the vector $V_k(S_c\psi_1)$. We proceed as follows. From the vector $S_c\psi_1$ we can generate a set of Givens rotations $G_1, \ldots, G_{k-1}$ such that $G_{k-1}^* G_{k-2}^* \ldots G_1^*(S_c\psi_1) = \nu e_k$, where ν is an arbitrary unit scalar (which can be set to $\nu = 1$ if this is advantageous). Now we update V_k using these Givens rotations:

$$\widehat{V}_k = V_k G_1 \ldots G_{k-1}. \tag{3.1}$$

So, we have for the last column of $\widehat{V}_k$

$$\widehat{V}_k e_k = V_k G_1 \ldots G_{k-1} e_k = V_k (S_c \psi_1) \bar{\nu},$$

and discarding the last vector by setting $k \leftarrow k-1$ gives the desired result. We have to update H_k by computing $\widehat{H}_k = (G_1 \ldots G_{k-1})^* H_k (G_1 \ldots G_{k-1})$. The cost of this truncation is $O(Nk)$. This process is easily repeated to discard further vectors. Hence, purging a subspace of dimension much less than k has cost $O(Nk)$.

In order to deal with poor extensions of the search space we check whether $\|(I - V_k V_k^*)t\|_2 < \delta \|t\|_2$, where typical values of δ are 1/100 or 1/1000. If this is the case we continue the iteration normally, but for the next m Jacobi-Davidson iterations we will follow the algorithm outlined in Section 2 under *Approach 1*. Typical values for m are 3 to 5. When following *Approach 1*, in order to implement (2.14) we use the same trick with the Givens rotations outlined above with $G_{k-1}^* G_{k-2}^* \ldots G_1^* s = \nu e_k$. However, we do not update V_k and W_k but add the results immediately into Q using two additional temporary vectors for intermediate results. Then we compute the QR decomposition of Q. We compute $t = C_m y_2$ and orthogonalize t against V_k. If in this case $\|(I - V_k V_k^*)t\|_2 < \delta \|t\|_2$ we will follow *Approach 2* in the next iteration. For the examples in this paper, this was never necessary.

Finally, we may need to truncate when the number of columns in the matrix V_k becomes too large. We use the 'Schur vector' strategy proposed in [14]. With one exception (to make a particular point), in the experiments discussed in the next section we do not use such truncations to avoid confusion between the effects of our algorithmic extensions and the effects of this type of truncation.

4. Numerical Experiments. We will discuss three test problems. All experiments were carried out using Matlab version 5.3.

The first problem is specifically constructed to analyze the effects of ill-conditioning of the eigenvector s of H_k. The other two problems are more realistic and concern the effects of additional orthogonalization in the linear solver, following *Approach 1*, to improve the effectiveness of the extension to the search space. The third problem is also used as an example that using refined Ritz vectors at each iteration for the Jacobi-Davidson algorithm may lead to worse convergence than using Ritz vectors.

Problem 1. The first problem is derived from a test problem in [13] using a diagonal matrix. We change the test problem by a similarity transformation that will make the matrix nonnormal. Let the diagonal matrix D be given by $diag((\frac{k}{100})^2 - 0.8)$ for $k = 1 \ldots 100$. Let S be a bidiagonal matrix with diagonal elements β and upper diagonal elements equal to 1. For this experiment we use $\beta = 0.80$. We define the matrix $A = SDS^{-1}$, and we will compute the smallest absolute eigenvalue ($\lambda = -0.0079$) and the associated eigenvector. Note that this is an interior eigenvalue. The correction equation (1.1) is solved approximately using 10 steps of GMRES.

We show the convergence of the standard Jacobi-Davidson algorithm in Fig. 4.1. Note how the Jacobi-Davidson algorithm converges slowly because $\|(I - V_k V_k^*)t\|_2 / \|t\|_2$ is fairly small. We see that right from the start $S_c^* H_k S_c - \theta I$ is rather ill-conditioned. As the ill-conditioning becomes severe, $\|r_s\|_2$ becomes large, and the algorithm stagnates. The restart parameter is set to 80. After the restart (using Schur vector truncation) $S_c^* H_k S_c - \theta I$ is better conditioned (drop of ω_1/ω_{k-1}) and the algorithm converges in 10 more iterations. This emphasizes that the stagnation was not due to the fact that the projection of the desired eigenvector x on range(V_k) was small, but due entirely to the effects of ill-conditioning.

In Fig. 4.2 we show the convergence of the Jacobi-Davidson algorithm using refined Ritz vectors. This is expensive, but we show the results to illustrate the effect on the convergence. We use the refined Ritz vectors also to extend the search space, just as in [8]. When we use refined Ritz vectors, for this problem, $\|r_s\|_2$ (not shown)

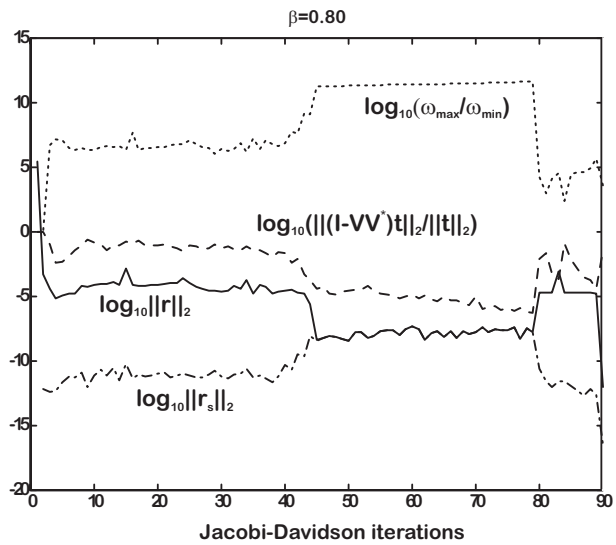

FIG. 4.1. *Convergence of the standard Jacobi-Davidson algorithm for Problem 1 and its relation to our various criteria.*

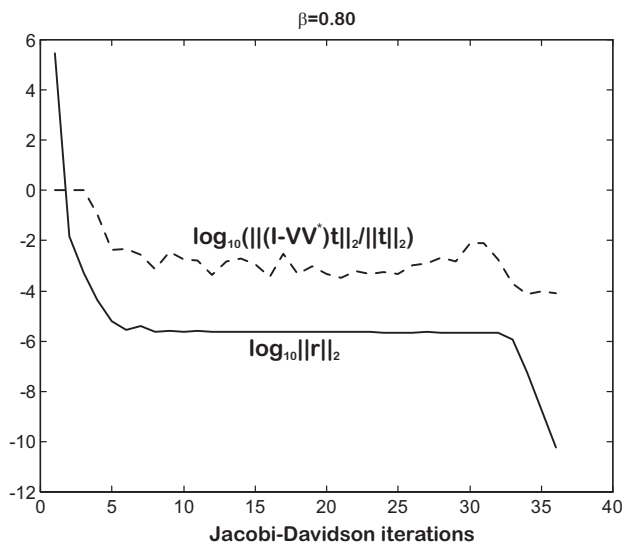

FIG. 4.2. *Convergence of the Jacobi-Davidson algorithm using refined Ritz vectors for Problem 1 and its relation to* $\|(I - V_k V_k^*)t\|_2/\|t\|_2$.

remains small. Note that $\|(I - V_k V_k^*)t\|_2/\|t\|_2$ is fairly small (but better than for the standard Jacobi-Davidson algorithm) and the algorithm converges slowly.

In Fig. 4.3 we show the convergence of the Jacobi-Davidson algorithm using truncation to maintain a well-conditioned eigenvector s. After a few iterations the algorithm does one single truncation, which significantly improves the conditioning of s; note the drop of ω_1/ω_{k-1} and of $\|r_s\|_2$. This shows our truncation algorithm is very effective. After this ω_1/ω_{k-1} increases slowly but without affecting $\|r_s\|_2$, except at the very end. At the end ω_1/ω_{k-1} increases drastically, but since $\|s^* H_k S_c\|_2/\|S_c^* H_k S_c\|_2$

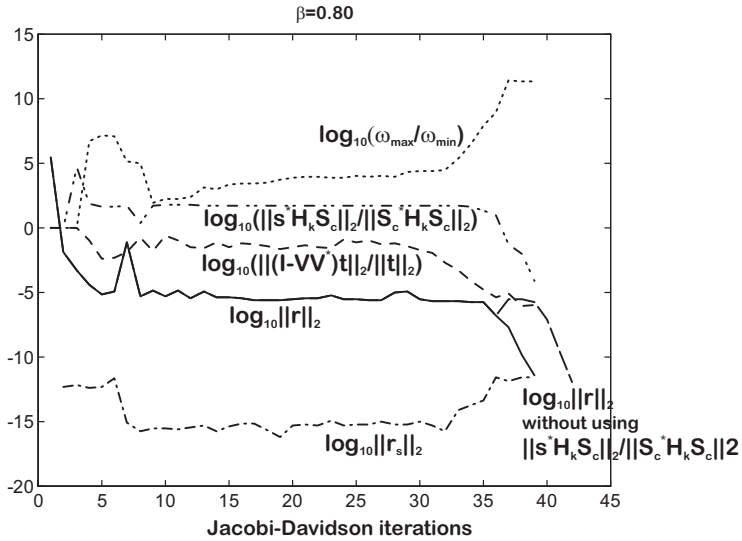

FIG. 4.3. *Convergence of the standard Jacobi-Davidson algorithm with truncation for Problem 1 and its relation to our various criteria.*

becomes small we do not truncate. The Jacobi-Davidson algorithm with truncation converges in a few more iterations than the Jacobi-Davidson algorithm using refined Ritz vectors; however, the Jacobi-Davidson algorithm with truncation does significantly less work (flops). Moreover, the Jacobi-Davidson algorithm with truncation achieves this convergence with only a single truncation (of a subspace of dimension 1).

We also provide the convergence for the case that we do not use the ratio $\|s^* H_k S_c\|_2 / \|S_c^* H_k S_c\|_2$ to avoid truncation when this ratio is small. In this case the algorithm performs a truncation at iteration 35. This improves $\|r_s\|_2$ and ω_1/ω_{k-1} again drastically (not shown in the figure), but the unnecessary truncation delays convergence by a few iterations.

In Fig. 4.4 we show the convergence of the Jacobi-Davidson algorithm using truncation to maintain a well-conditioned eigenvector s and using *Approach 1* in the linear solver when $\|(I - V_k V_k^*)t\|_2 / \|t\|_2$ becomes small. It performs three truncations in the first few iterations. This algorithm has the fastest convergence of the four Jacobi-Davidson variants.

Problem 2. The second example involves a small convection-diffusion problem with strong convection, leading to a strongly nonsymmetric (non-normal) matrix. The matrix A is derived from the finite volume discretization of the partial differential equation

$$-u_{xx} - u_{yy} + 20u_x - 30u_y = 0,$$

on $[0,1] \times [0,1]$ with Dirichlet boundary conditions, $u = 1$ for $x = 0$ and $y = 1$, and $u = 0$ for $x = 1$ and $y = 0$. We discretize the system using 22 mesh points in each direction. We solve again for the (absolute) smallest eigenvalue and the associated eigenvector. In this case the smallest eigenvalue is on the boundary of the spectrum. The correction equation is solved approximately using 20 steps of GMRES.

For this problem there is no need for truncation, and $\|(I - V_k V_k^*)t\|_2 / \|t\|_2$ remains

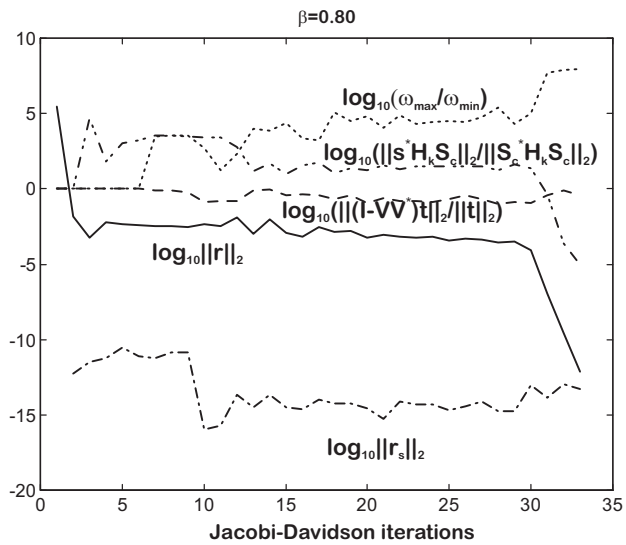

FIG. 4.4. *Convergence of the Jacobi-Davidson algorithm with truncation and* Approach 1 *for the linear solver for Problem 1 and its relation to our various criteria.*

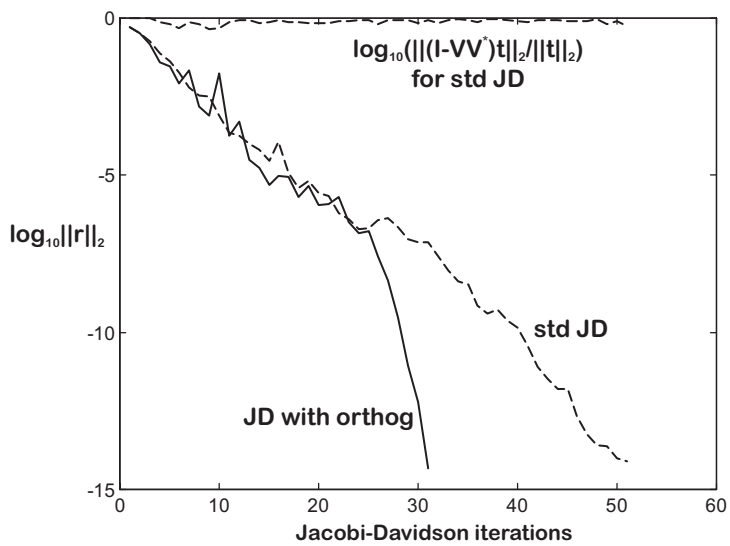

FIG. 4.5. *Convergence of the standard Jacobi-Davidson algorithm and the Jacobi-Davidson algorithm using* Approach 1 *for the linear solver for Problem 2.*

close to 1, as shown for the standard Jacobi-Davidson algorithm; see Fig. 4.5. For the Jacobi-Davidson algorithm using *Approach 1* in the linear solver, we forced the algorithm to do the additional orthogonalization in the linear solver at each iteration. This leads to much faster convergence in the linear solver and hence in the eigensolver. The standard Jacobi-Davidson algorithm takes about 70% more Jacobi-Davidson iterations than the version using *Approach 1*. This shows that extra work in the linear solver may be worth while to improve convergence.

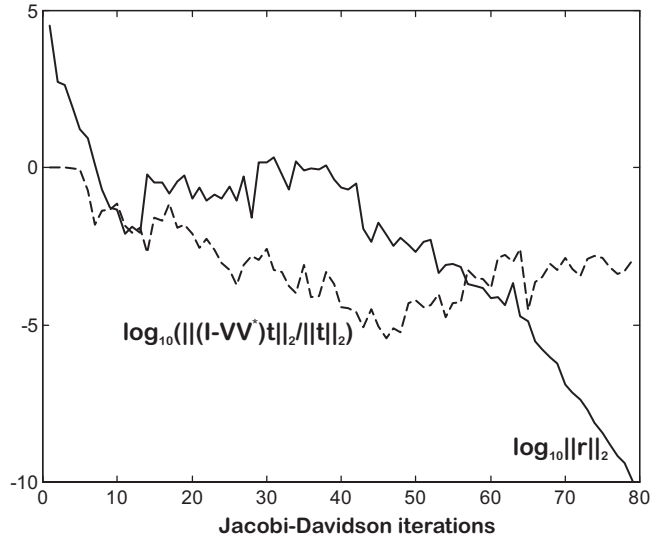

FIG. 4.6. *Convergence of the standard Jacobi-Davidson algorithm for Problem 3.*

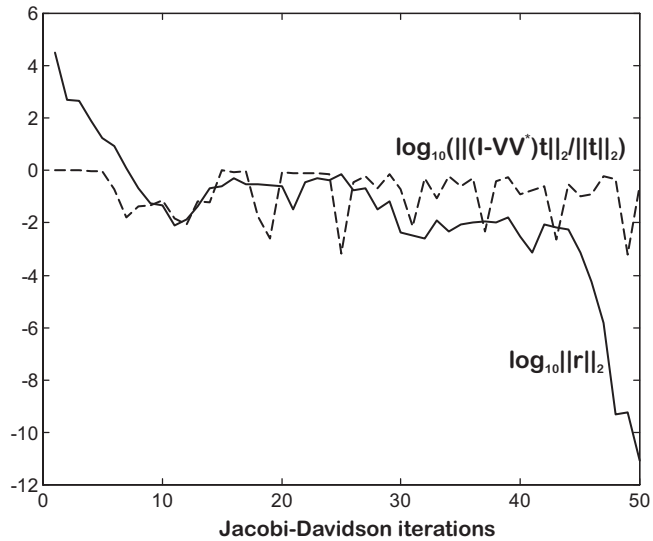

FIG. 4.7. *Convergence of the Jacobi-Davidson algorithm using* Approach 1 *in the linear solver for Problem 3.*

Example 3. In our last example we compute an interior eigenpair of the matrix West0479 from the public domain Harwell-Boeing collection. We solve for an interior eigenvalue close to the (complex) value $(-17.825, -4.6376)$ and for the associated eigenvector. The correction equation is solved approximately using 20 steps of GMRES.

We compare the standard Jacobi-Davidson algorithm with the Jacobi-Davidson algorithm using *Approach 1* for the linear solver, and with the Jacobi-Davidson algorithm using refined Ritz vectors (also for extending the search space); see Figs. 4.6, 4.7, and 4.8. For this problem there was no need for truncation. The Jacobi-Davidson algo-

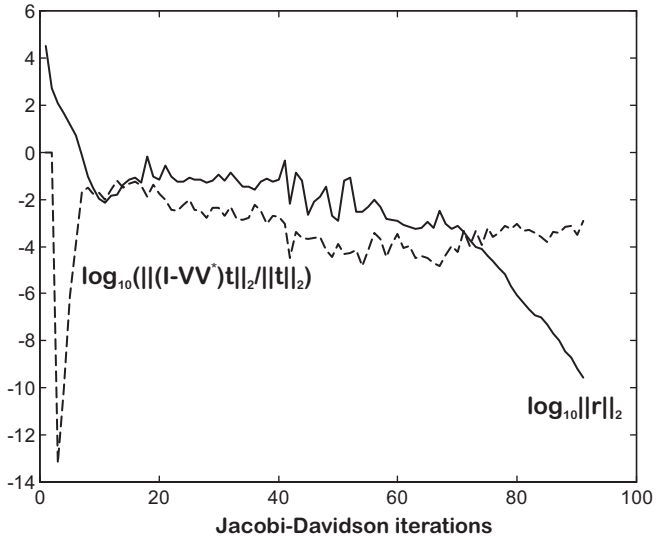

FIG. 4.8. *Convergence of the Jacobi-Davidson algorithm using refined Ritz vectors for Problem 3.*

rithm using *Approach 1* for the linear solver uses parameters $\delta = 0.01$ and $m = 5$, the number of Jacobi-Davidson iterations using *Approach 1* after $\|(I - V_k V_k^*)t\|_2 < \delta\|t\|_2$ occurs. Occasionally, $\|(I - V_k V_k^*)t\|_2/\|t\|_2$ is small, but the inner orthogonalization quickly brings the ratio close to 1. The standard Jacobi-Davidson algorithm and the Jacobi-Davidson algorithm with refined Ritz vectors suffer significantly from small $\|(I - V_k V_k^*)t\|_2/\|t\|_2$ ratios and converge much slower than the Jacobi-Davidson algorithm using *Approach 1*. Note that the Jacobi-Davidson algorithm with refined Ritz vectors converges slowest (and is the most expensive per iteration).

5. Conclusions. We have explained and analyzed two problems that can seriously deteriorate the convergence of the Jacobi-Davidson algorithm and even prevent it from converging. Based on our analysis we have proposed solutions to these problems, and we have demonstrated their usefulness in numerical experiments. In addition, we have shown how the convergence of the Jacobi-Davidson algorithm can be improved by including the existing search space for the eigenproblem in the search space for solving the (linear) correction equation.

6. Acknowledgements. The author would like to thank Andreas Stathopoulos for many useful discussions.

REFERENCES

[1] A. Booten, D. Fokkema, G. Sleijpen, and H. Van der Vorst, *Jacobi-Davidson methods for generalized MHD-eigenvalue problems*, Z. Angew. Math. Mech., 76 (1996), pp. 131–134. ICIAM/GAMM 95 (Hamburg, 1995).

[2] E. De Sturler, *Nested Krylov methods based on GCR*, J. Comput. Appl. Math., 67 (1996), pp. 15–41.

[3] ———, *Truncation strategies for optimal Krylov subspace methods*, SIAM J. Numer. Anal., 36 (1999), pp. 864–889. electronically available from http://epubs.siam.org.

[4] ———, *Variations on the Jacobi-Davidson theme*, in Iterative methods in Scientific Computation IV: Proceedings of the Fourth IMACS International Symposium on Iterative Methods in Scientific Computation, D. Kincaid and A. Elster, eds., IMACS series in Computational and Applied Mathematics - volume 5, IMACS, 1999, pp. 313–323.

[5] E. De Sturler and D. R. Fokkema, *Nested Krylov methods and preserving the orthogonality*, in Sixth Copper Mountain Conference on Multigrid Methods, N. D. Melson, T. A. Manteuffel, and S. F. McCormick, eds., NASA Conference Publication 3224, Part 1, Hampton, VA, USA, 1993, NASA Langley Research Center, pp. 111–125.

[6] M. Genseberger and G. L. Sleijpen, *Alternative corection equations in the Jacobi-Davidson method*, Numer. Linear Algebra Appl., 6 (1999), pp. 235–253.

[7] G. Golub and C. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, second ed., 1989.

[8] Z. Jia, *Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems*, Linear Algebra and Its Applications, 259 (1997), pp. 1–23.

[9] Z. Jia and G. W. Stewart, *On the convergence of Ritz values, Ritz vectors, and refined Ritz vectors*, Tech. Report TR-99-038, Institute for Advanced Computer Studies, Department of Computer Science, University of Maryland, College Park, College Park, MD 20742, 1999. electronically available from http://www.cs.umd.edu/stewart/.

[10] W. Kahan, B. N. Parlett, and E. Jiang, *Residual bounds on approximate eigensystems of nonnormal matrices*, SIAM J. Numer. Anal., 19 (1982), pp. 470–484.

[11] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, United Kingdom, 1992.

[12] G. L. G. Sleijpen, A. 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. International Linear Algebra Year (Toulouse, 1995).

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

[14] G. L. G. Sleijpen and H. A. Van der Vorst, *The Jacobi-Davidson method for eigenvalue problems and its relation with accelerated inexact Newton schemes*, tech. report, Department of Mathematics, Utrecht University, Utrecht, The Netherlands, 1995.

[15] A. Stathopoulos and J. McCombs, *A parallel, block, jacobi-davidson implementation for solving large eigenproblems on coarse grain environments*, in Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, Vol. VI, Atlanta, 1999, CSREA Press, 1999, pp. 2920–2926.

[16] A. Stathopoulos and Y. Saad, *Restarting techniques for the (Jacobi-)davidson symmetric eigenvalue methods*, Elec. Trans. on Numer. Anal. (ETNA), 7 (1998), pp. 163–181.

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