The Jacobi–Davidson method

M.E. Hochstenbach[*] [1] and **Y. Notay**[**2]

[1] Department of Mathematics, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, OH 44106, USA.
[2] Service de Métrologie Nucléaire (C.P. 165/84), Université Libre de Bruxelles, 50, Av. F.D. Roosevelt, B-1050 Brussels, Belgium.

Received 15 November 2004, revised 30 November 2004, accepted 2 December 2004
Published online 3 December 2004

Key words Eigenvalue, eigensolver, Jacobi–Davidson, harmonic extraction, refined extraction, correction equation.
MSC (2000) 65F15

The Jacobi–Davidson method is a popular technique to compute a few eigenpairs of large sparse matrices. Its introduction, about a decade ago, was motivated by the fact that standard eigensolvers often require an expensive factorization of the matrix to compute interior eigenvalues. Such a factorization may be infeasible for large matrices as arise in today's large-scale simulations. In the Jacobi–Davidson method, one still needs to solve "inner" linear systems, but a factorization is avoided because the method is designed so as to favor the efficient use of modern iterative solution techniques, based on preconditioning and Krylov subspace acceleration. Here we review the Jacobi–Davidson method, with the emphasis on recent developments that are important in practical use.

1 Introduction

The Jacobi–Davidson (JD) method was introduced around ten years ago by Sleijpen and van der Vorst ([59], see also [61]). It is considered to be one of the best eigenvalue solvers, especially for eigenvalues in the interior of the spectrum. Given an $n \times n$ real or complex matrix A, where A is typically large and sparse, we look for some relevant eigenpairs $(\lambda, \mathbf{x})$ satisfying $A\mathbf{x} = \lambda\mathbf{x}$. Here, "relevant" means that one is interested in a specific part of the spectrum, for instance in the eigenvalues closest to some given target τ.

Before the introduction of JD, such problems were often attacked either by the Davidson method, or by a scheme from the Lanczos or Arnoldi family. Davidson's method [12] is efficient for computing a few of the smallest eigenvalues of a real symmetric (or Hermitian) matrix that is close to a diagonal one in some sense, which often arises in computational chemistry. Unfortunately, this method is not well established outside this specific context, and, in fact, it was not yet very well understood a decade ago—two decades after its introduction. (The development of the JD method had a better understanding of the Davidson method as side effect, see [59, 51].)

[*] Corresponding author: e-mail: hochsten@case.edu, Phone: +1 216 368 5015, Fax: +1 216 368 5163. The research of this author was supported in part by NSF grant DMS-0405387. Version August 2005.
[**] Supported by the "Fonds National de la Recherche Scientifique", Maître de recherches.

Lanczos [40] and Arnoldi [3] are well-known eigenvalue methods, which, however, exhibit slow convergence and tend to have difficulties finding interior eigenvalues. This is cured by "shift-and-invert" variants of these methods, but these require a factorization of the shifted matrix $A - \tau I$ (I denotes the identity matrix), which may be very expensive or even not feasible. This factorization is needed to solve "inner" linear systems of the form $(A-\tau I)\mathbf{x} = \mathbf{y}$. Therefore, a natural alternative to a factorization is offered by iterative solvers. Unfortunately, it is often unattractive to combine this approach with Lanczos and Arnoldi, mainly because these methods require accurate solutions of inner systems, implying much numerical effort if an iterative solver is used. In the JD method, this pitfall is avoided thanks to a clever combination of ideas stemming from Jacobi [34, 35] and the Davidson method.

JD belongs to the class of subspace methods, which means that approximate eigenvectors are sought in a subspace $\mathcal{U}$. Each iteration of these methods has two important phases: the subspace extraction, in which a sensible approximate eigenpair is sought, of which the approximate vector is in the search space; and the subspace expansion, in which the search space $\mathcal{U}$ is enlarged by adding a new basis vector to it, hopefully leading to better approximate eigenpairs in the next extraction phase. One of the strengths of JD is that both phases may cooperate to approximate difficult-to-find interior eigenvalues.

Here we present a short review of the JD method with some of the latest developments. For more discussions on Jacobi–Davidson and other eigenvalue methods, see, e.g., [74, 78, 41, 4, 68, 71, 64, 75, 77]. Sections 2 and 3 deal with the main two phases in a subspace method: the subspace expansion and the subspace extraction. Convergence and the solution of the correction equation are the subjects of Sections 4 and 5. Practical aspects, recent developments and variants are discussed in Sections 6 and 7. We conclude with some numerical illustrations.

2 Subspace expansion

For the subspace expansion phase, we are given an approximate eigenpair $(\theta, \mathbf{u})$ to $(\lambda, \mathbf{x})$, with $\mathbf{u} \in \mathcal{U}$, where $\mathcal{U}$ is a low k-dimensional subspace, the search space, and

$$\theta = \frac{\mathbf{u}^* A \mathbf{u}}{\mathbf{u}^* \mathbf{u}}$$

is the Rayleigh quotient of $\mathbf{u}$, taken as approximate eigenvalue because it minimizes the two-norm of the residual: $\|\mathbf{r}\| = \|A\,\mathbf{u} - \theta\,\mathbf{u}\| = \min_\eta \|A\,\mathbf{u} - \eta\,\mathbf{u}\|$. Then we are interested in an expansion of $\mathcal{U}$ by an appropriate direction to get an even better approximate eigenpair. The idea is now to look for an orthogonal correction $\mathbf{s} \perp \mathbf{u}$ such that $\mathbf{u} + \mathbf{s}$ satisfies the eigenvalue equation:

$$A(\mathbf{u} + \mathbf{s}) = \lambda(\mathbf{u} + \mathbf{s}). \tag{1}$$

Let $\widetilde{\lambda}$ be the best available approximation to the unknown eigenvalue λ. In practice, this means $\widetilde{\lambda} = \theta$ if the approximate eigenpair $(\theta, \mathbf{u})$ is already good enough (as may be checked by computing the residual norm). However, far from convergence, if one looks for the eigenvalue closest to some target τ, it may be better to simply select $\widetilde{\lambda} = \tau$. Equation (1) can now be rewritten using this approximation $\widetilde{\lambda}$ to obtain

$$(A - \widetilde{\lambda} I)\mathbf{s} = -\mathbf{r} + (\lambda - \theta)\mathbf{u} + (\lambda - \widetilde{\lambda})\mathbf{s}. \tag{2}$$

During the process, λ and hence also $\lambda - \widetilde{\lambda}$ and $\lambda - \theta$ are unknown. However, both $\|\mathbf{s}\|$ and $|\lambda - \widetilde{\lambda}|$ are expected to be small, indicating that the last term on the right-hand side is smaller than the other ones and therefore that it is sensible to neglect it. In fact, in an asymptotic situation (close to the eigenpair) this term represents a error of second order ($\mathcal{O}(\|\mathbf{s}\|^2)$) when $\widetilde{\lambda} = \theta$; for Hermitian matrices it is even of third order. The second term on the right-hand side, however, is one order of magnitude larger, and the idea is to project it out, rather than discarding it. At the same time, we would like to fix the information that is present in the residual $\mathbf{r}$.

Since $\mathbf{r} \perp \mathbf{u}$, both of these wishes are met if we multiply both sides of (2) on the left by the orthogonal projection $I - \mathbf{u}\,\mathbf{u}^*$. Remembering that we neglect the last term on the right-hand side, this results in $(I - \mathbf{u}\,\mathbf{u}^*)(A - \widetilde{\lambda}I)\mathbf{s} = -\mathbf{r}$. Since $\mathbf{s} \perp \mathbf{u}$, we arrive at the JD correction equation:

$$(I - \mathbf{u}\,\mathbf{u}^*)(A - \widetilde{\lambda}I)(I - \mathbf{u}\,\mathbf{u}^*)\mathbf{s} = -\mathbf{r} \qquad \text{where } \mathbf{s} \perp \mathbf{u}. \tag{3}$$

From this form it is clear that the shifted operator $A - \widetilde{\lambda}I$ is restricted to the orthogonal complement of $\mathbf{u}$. If we solve this equation exactly, then we find [59]

$$\mathbf{s} = -\mathbf{u} + \alpha(A - \widetilde{\lambda}I)^{-1}\mathbf{u}, \tag{4}$$

where $\alpha = (\mathbf{u}^*(A - \widetilde{\lambda}I)^{-1}\mathbf{u})^{-1}$ is such that $\mathbf{s} \perp \mathbf{u}$. The solution is used to expand the search space $\mathcal{U}$. Since $\mathbf{u}$ is already in this space, the expansion vector is effectively $(A - \widetilde{\lambda}I)^{-1}\mathbf{u}$, which is the same as for inverse iteration if $\widetilde{\lambda} = \tau$, and the same as for Rayleigh quotient iteration (RQI) if $\widetilde{\lambda} = \theta$. Therefore, JD where we solve the correction equation exactly is a "subspace accelerated" inverse iteration or RQI, depending on the phase in which we are. The subspace acceleration stems from the fact that $(A - \widetilde{\lambda}I)^{-1}\mathbf{u}$ does not directly give the next approximate eigenvector as in the standard version of these methods: a hopefully even better approximation is sought in the subspace formed by $\mathcal{U}$ expanded by this new vector.

Note that the combined use of inverse iteration and RQI is very sensible [70]. Indeed, the asymptotic convergence of RQI is fast: quadratic, or cubic for Hermitian matrices [54] (see also Section 4). However, before this asymptotic phase, it is difficult to control towards which eigenvalue RQI will converge. This motivates the selection $\widetilde{\lambda} = \tau$ during the initial phase, since inverse iteration favors unconditionally the convergence towards the eigenvalue closest to τ.

Now, the key idea in JD is to solve the linear system (3) only approximately (or "inexactly"), for example by an iterative method, see Section 5. The fact that we only neglect a second-order term (or an even third-order term for Hermitian matrices) in the derivation of the method suggests that JD where we solve the correction equation exactly is a Newton method, which is true indeed [60]. Therefore, JD combined with an iterative solver for the correction equation is also an accelerated inexact Newton method [16].

3 Subspace extraction

Now we come to the subspace extraction phase, that is, we are interested in sensible approximate eigenpairs $(\theta, \mathbf{u})$, where $\mathbf{u} \in \mathcal{U}$. For exterior eigenvalues, JD generally uses the

well-known Rayleigh–Ritz approach as follows. Let U be an $n \times k$ matrix whose columns form an orthonormal basis of $\mathcal{U}$. We can write $\mathbf{u}$ as a linear combination of these basis vectors, $\mathbf{u} = U\mathbf{c}$, where $\mathbf{c}$ is a small (k-dimensional) vector. The Rayleigh–Ritz method imposes the Ritz–Galerkin condition

$$\mathbf{r} = A\mathbf{u} - \theta\mathbf{u} = AU\mathbf{c} - \theta U\mathbf{c} \perp \mathcal{U}. \tag{5}$$

This holds if and only if $(\theta, \mathbf{c})$ is a solution of the low-dimensional projected eigenproblem

$$U^*AU\mathbf{c} = \theta\mathbf{c}.$$

An approximate eigenvector $\mathbf{u} = U\mathbf{c}$ is called a "Ritz vector", whereas $\mathbf{c}$ is a "primitive Ritz vector", and θ is a "Ritz value". Note that the Galerkin condition (5) implies $\mathbf{r} \perp \mathbf{u}$ and hence that θ is the Rayleigh quotient of $\mathbf{u}$.

However, the Rayleigh–Ritz method generally gives poor approximate eigenvectors for interior eigenvalues. Basically, the Galerkin condition (5) does not imply anything about the residual norm. The refined Rayleigh–Ritz extraction addresses this issue. Given an approximate eigenvalue $\widetilde{\lambda}$, for instance a Ritz value θ or the target τ, we solve for

$$\widehat{\mathbf{c}} = \operatorname*{argmin}_{\mathbf{c}\in\mathbb{C}^k,\, \|\mathbf{c}\|=1} \|(AU - \widetilde{\lambda}U)\mathbf{c}\|.$$

Often, the refined Ritz vector $\widehat{\mathbf{u}} := U\widehat{\mathbf{c}}$ is much better than an ordinary Ritz vector; we can take the Rayleigh quotient of this vector to determine a new approximate eigenvalue. Since the refined Ritz extraction mainly got its popularity after the publication of [59] (see [38]), the combination of JD and the refined extraction was not mentioned in the original paper; however, it was used in available software, see also [13].

A second alternative extraction method is the harmonic Rayleigh–Ritz method. The eigenvalues closest to the target τ are the eigenvalues of largest magnitude of the shifted-and-inverted matrix $(A - \tau I)^{-1}$, which is favorable for a Galerkin approach. This suggests imposing a condition of the form

$$(A - \tau I)^{-1}\widetilde{\mathbf{u}} - (\widetilde{\theta} - \tau)^{-1}\widetilde{\mathbf{u}} \perp \mathcal{V},$$

for a certain test space $\mathcal{V}$. To avoid working with the inverse of a large sparse matrix we choose $\mathcal{V} = (A - \tau I)^*(A - \tau I)\mathcal{U}$. With $\widetilde{\mathbf{u}} = U\widetilde{\mathbf{c}}$, this leads to the projected generalized eigenvalue problem

$$U^*(A - \tau I)^*(A - \tau I)U\widetilde{\mathbf{c}} = (\widetilde{\theta} - \tau)U^*(A - \tau I)^*U\widetilde{\mathbf{c}},$$

where our interest is in the pair(s) with the minimal $|\widetilde{\theta} - \tau|$; the harmonic Ritz vector $\widetilde{\mathbf{u}}$, like the refined Ritz vector, is generally a better approximate eigenvector than an ordinary Ritz vector, mainly because we have the bound $\|(A - \tau I)\widetilde{\mathbf{u}}\| \leq |\widetilde{\theta} - \tau|$ for the residual norm. Again, usually the Rayleigh quotient of an harmonic Ritz vector is taken as approximation to the eigenvalue, since this quantity is often better than the harmonic Ritz value $\widetilde{\theta}$ [62]. See [68] for an overview of the different extraction processes, and [26] for some variants.

4 Convergence

As mentioned in Section 2, with exact solution of the correction equation, the JD method behaves either as subspace accelerated inverse iteration or as subspace accelerated RQI, depending on the choice of $\widetilde{\lambda}$. In both cases, letting λ_1 be the eigenvalue closest to $\widetilde{\lambda}$, the convergence depends mainly on the ratio

$$\rho_{\widetilde{\lambda}} \;=\; \frac{|\lambda_1 - \widetilde{\lambda}|}{\min_{\lambda \in \sigma(A)\setminus\{\lambda_1\}} |\lambda - \widetilde{\lambda}|}, \tag{6}$$

where $\sigma(A)$ is the spectrum of A. Thus, the RQI approach is expected to be faster when the Rayleigh quotient θ is closer to λ_1 than the fixed target τ. The fact that ρ_θ decreases as θ converges to λ_1 already indicates a superlinear convergence of the method. As mentioned before, the convergence of RQI is quadratic in general and cubic if A is Hermitian.

Since the JD method has been introduced with an iterative solution of the correction equation in mind, it is important to know how the above convergence factor is affected when the inner iteration is solved only up to some modest accuracy. There are a few results available. Concerning the initial phase where we select $\widetilde{\lambda} = \tau$, if the matrix is Hermitian and if τ is smaller than the smallest eigenvalue, the JD method is very close to preconditioned inverse iteration, for which analysis shows that it is not necessary to solve the inner systems accurately [39].

With respect to the asymptotic phase where we select $\widetilde{\lambda} = \theta$, assuming that the matrix is Hermitian and that the eigenvalue closest to τ is the smallest one, it is proved in [50] that the convergence factor ρ_θ becomes

$$\widetilde{\rho}_{\theta,\gamma} \;=\; \frac{\rho_\theta + \gamma}{1 + \rho_\theta\,\gamma}, \tag{7}$$

where γ is the relative error left in the correction equation, measured with respect to the energy norm associated with the matrix $(I - \mathbf{u}\,\mathbf{u}^*)(A - \theta\, I)(I - \mathbf{u}\,\mathbf{u}^*)$ [1]. Thus, as long as $\gamma \gg \rho_\theta$, any progress made in the linear system solution is reflected in the outer convergence towards the desired eigenvector. On the other hand, to recover the cubic convergence of RQI, it suffices to make $\gamma = \mathcal{O}(\rho_\theta)$, so that $\widetilde{\rho}_{\theta,\gamma} = \mathcal{O}(\rho_\theta)$. However, to avoid waste of numerical effort, it is wise to stop the inner iteration as soon as the "outer" convergence (towards the desired eigenpair) slows down. This requires a careful implementation, see [49] for an example.

Although the scope of these theoretical results is limited, they reflect the behavior of the method as observed in practice in more complicated situations well. When $\widetilde{\lambda} = \tau$, a modest tolerance in the solution of the correction equation is enough and generally represents the most cost effective option; when $\widetilde{\lambda} = \theta$, it may pay off to increase the accuracy as the convergence proceeds, but this should be implemented with care.

Complementary results on the convergence of JD can be found in [73, 57, 53, 52].

[1] The energy norm associated with a matrix C is given by $\|\mathbf{x}\|_C = (\mathbf{x}\,,\, C\,\mathbf{x})^{1/2}$; it is well defined only if C positive definite; in the present case, it can be proved that if the eigenvalue closest to θ is the smallest one, then $(I - \mathbf{u}\,\mathbf{u}^*)(A - \theta\, I)(I - \mathbf{u}\,\mathbf{u}^*)$ is positive definite on $\mathbf{u}^\perp$ (that is, the subspace of vectors orthogonal to $\mathbf{u}$) [49].

5 Solving the correction equation

A basic iterative scheme to solve a linear system $C\,\mathbf{x} = \mathbf{y}$ may be sketched as follows

$$
\begin{aligned}
&\mathbf{x}_0 \quad : \quad \textit{initial approximation (default: } \mathbf{x}_0 = \mathbf{0}\textit{)} \\
&\mathbf{g}_0 \;=\; \mathbf{y} \,-\, A\,\mathbf{x}_0 \\
&\textit{For } m = 0, 1, \ldots \textit{ until convergence:} \\
&\qquad \mathbf{d}_m \;=\; B^{-1}\,\mathbf{g}_m \\
&\qquad \mathbf{x}_{m+1} \;=\; \mathbf{x}_m \,+\, \mathbf{d}_m \\
&\qquad \mathbf{g}_{m+1} \;=\; \mathbf{y} \,-\, C\,\mathbf{x}_m .
\end{aligned}
$$

In this scheme, B is the preconditioner, a matrix that approximates C and has the property that computing $B^{-1}\mathbf{g}$ for some $\mathbf{g}$ is relatively cheap. The closer $B^{-1}C$ is to the identity matrix, the faster the convergence (one iteration is enough if $B = C$); see, e.g., [6, 56] for a recent review of modern preconditioning techniques.

Nowadays, one seldom uses such a simple scheme to solve large linear systems. Instead, methods based on Krylov subspaces are used which accelerate this basic scheme by enforcing the residual $\mathbf{g}_m$ to be orthogonal to an auxiliary subspace of increasing dimension, thanks to the use of more complex relations to define the "search direction" $\mathbf{d}_m$ and the approximate solution $\mathbf{x}_m$; see, e.g., [56, 76] for more details.

If C is Hermitian positive definite, the conjugate gradient (CG) method is the method of choice, because it minimizes the energy norm of the error while requiring only little extra work per iteration compared with the basic scheme sketched above. MINRES minimizes the residual norm. It is slightly more costly per iteration than CG, but is applicable to Hermitian indefinite systems, provided that the preconditioner is positive definite and given in factored form $B = M\,M^*$, which is not always true in practice. In all other cases, one may select either a "short recurrence method", which remains cheap but loses the minimum residual property, or a method like GMRES, which minimizes the residual norm at the price of a "long recurrence". This means that the cost of each iteration increases linearly with the iteration number, implying that in practice the method has to be stopped from time to time, and restarted with the best computed approximation so far as initial guess. In the context of the solution of the correction equation this drawback is not serious because only a modest accuracy is wanted for which a limited number of iterations should hopefully be enough.

The system matrix in the correction equation (3) is

$$
C \;=\; \big(I - \mathbf{u}\,\mathbf{u}^*\big)\big(A - \widetilde{\lambda}\,I\big)\big(I - \mathbf{u}\,\mathbf{u}^*\big). \tag{8}
$$

In practice, this matrix is never formed explicitly; instead the matrix vector product $C\,\mathbf{x}$ is carried out by sequentially applying each of C's three factors. To find a good preconditioner for C, we first use any standard technique to build a preconditioner K for the shifted matrix $A - \widetilde{\lambda}I$; for instance, one may compute an incomplete factorization of the matrix, or some sparse approximate inverse [6, 56]. Then we restrict K to the same subspace, that is, we define

$$
B \;=\; \big(I - \mathbf{u}\,\mathbf{u}^*\big)\,K\,\big(I - \mathbf{u}\,\mathbf{u}^*\big). \tag{9}
$$

This preconditioner is singular, but this is not an issue because the correction equation is solved in the subspace $\mathbf{u}^{\perp}$, and the entire iterative process is carried out in this subspace. Thus,

one will have to solve systems $B\,\mathbf{d} = \mathbf{g}$ only for $\mathbf{g} \in \mathbf{u}^\perp$, which ensures the compatibility of the system if K is nonsingular. Moreover, one is only interested in the solution that belongs to $\mathbf{u}^\perp$, which is given by

$$\mathbf{d} \;=\; \left(I - \frac{K^{-1}\mathbf{u}\mathbf{u}^*}{\mathbf{u}^* K^{-1}\mathbf{u}}\right) K^{-1}\mathbf{g}\,. \tag{10}$$

This requires two applications of K^{-1}, but one of them (the computation of $K^{-1}\mathbf{u}$) can be done beforehand once for all the iterations.

Some preconditioning techniques require non-negligible numerical effort to set up the matrix K. Fortunately, numerical experiments show that it is not necessary to update the preconditioner at each JD step. For instance, one may compute a preconditioner K for the matrix $A-\tau\,I$ corresponding to the phase where one uses a fixed target, and then use the same preconditioner throughout all iterations. If A is Hermitian and one looks for its smallest eigenvalues, it is very sensible to build K from $A - \tau\,I$ with τ smaller than the smallest eigenvalue (e.g., $\tau = 0$ if the matrix is positive definite). Indeed, standard preconditioning techniques generally perform better for a positive definite matrix than for an indefinite matrix; one can also consider some more specific methods such as (algebraic) multigrid [72].

Again, available theoretical results are restricted to the case where one wants to compute the smallest eigenvalues of a Hermitian matrix, with the target τ smaller than the smallest eigenvalue. The quality of the preconditioner K for $A-\tau\,I$ may then be assessed through the spectral condition number

$$\kappa(K^{-1}(A-\tau\,I)) \;=\; \frac{\nu_{\max}(K^{-1}(A-\tau\,I))}{\nu_{\min}(K^{-1}(A-\tau\,I))}, \tag{11}$$

where $\nu_{\max}(K^{-1}(A-\tau\,I))$ and $\nu_{\min}(K^{-1}(A-\tau\,I))$ are the largest and the smallest eigenvalue of the pencil $(A-\tau\,I)-\nu\,K$, respectively. Indeed, when solving a system $(A-\tau\,I)\mathbf{x} = \mathbf{y}$, the relative error in energy norm γ after k CG iterations is bounded by

$$\gamma \;\leq\; 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \tag{12}$$

(see, e.g., [76]). When solving the correction equation, the latter bound still applies but the condition number κ has to be defined as the ratio of the extremal eigenvalues of the pencil $C - \nu\,B$ considering only nontrivial eigenvalues, that is, only those with corresponding eigenvector orthogonal to $\mathbf{u}$.

Interestingly, this quantity may be bounded as follows [50]:

$$\kappa \;\leq\; \begin{cases} \kappa(K^{-1}(A-\tau\,I)) & \text{if } \widetilde{\lambda} = \tau \\ \kappa(K^{-1}(A-\tau\,I))\,\left(1 + \frac{\theta-\tau}{\lambda_1+\lambda_2-2\,\theta}\right) & \text{if } \widetilde{\lambda} = \theta \;<\; \frac{\lambda_1+\lambda_2}{2}, \end{cases} \tag{13}$$

where λ_2 is the second smallest eigenvalue of A. Hence, solving the correction equation is essentially as fast as solving a system $(A-\tau\,I)\mathbf{x} = \mathbf{y}$ with preconditioner K (as long as the convergence is well described by the bound (12), that is, neglecting further acceleration stemming from so-called superlinear convergence effects).

This validates the use of the projected preconditioner (9) with K kept constant throughout all JD steps. Note that combining (13) with (12), one obtains a bound on the parameter γ appearing in (7). Thus, one is able to trace the convergence towards the desired eigenpair as a function of $\kappa(K^{-1}(A - \tau I))$ and of the number of *inner* iterations, see [50] for details.

We refer to [5, 4] for more details, including pseudocodes, on the implementation of iterative solvers for the correction equation. Specific preconditioners for the correction equation are discussed in [18, 63].

6 Practical aspects

In Algorithm 6.1, we give a pseudocode for the basic Jacobi–Davidson algorithm. Here, `rgs` stands for repeated Gram-Schmidt or any other numerically stable method to expand an orthonormal basis.

Input: A device to compute Au for arbitrary u, a starting vector u_1, and a tolerance ε
Output: An approximate eigenpair (θ, u) closest to the target τ

1. $s = u_1$, $U_0 = [\,]$
   for $k = 1, 2, \ldots$
2. U_k = `rgs`(U_{k-1}, s)
3. Compute kth column of AU_k
   Compute kth row and column of $H_k = U_k^* A U_k$
4. Compute eigenpair (θ, c), where $\theta \approx \tau$ of the projected system
   using standard, harmonic, or refined extraction techniques
   $u = U_k c$
5. $r = AU_k c - \theta u$
6. Stop if $\|r\| \leq \varepsilon$
7. Solve (approximately) $s \perp u$ from
   $(I - uu^*)(A - \theta I)(I - uu^*)\, s = -r$

ALGORITHM 6.1: *The basic Jacobi–Davidson method for the computation of an approximate eigenpair (θ, u) closest to the target τ.*

The practical performance of the JD method can be speeded up considerably by several practical enhancements. First consider restarts. In practice it is necessary to shrink the search space $\mathcal{U}$ every now and then to ensure modest computational and memory costs. When the dimension of $\mathcal{U}$ reaches a given maximal value `maxdim`, the `mindim` most promising approximate eigenvectors are extracted to form a new reduced basis for $\mathcal{U}$, with which the method is restarted. Experiments suggest not to restart using too few vectors; popular values are $5 \leq$ `mindim` ≤ 10 and for $20 \leq$ `maxdim` ≤ 30. Inspired by the three-term recurrence processes for CG and BiCG, Saad and Stathopoulos have launched ideas for a thick restart [67, 66], where also one or more of the approximate eigenvectors of the previous step are used.

Since the JD method can be seen as an accelerated inexact Newton process with favorable local convergence properties, it is advisable to start with a sensible approximate eigenpair. In Sleijpen's JDQR and JDQZ codes [37], this is done by first generating a Krylov search space of dimension `mindim`, before the genuine JD process sets in.

Two other techniques which often lead to a considerable performance improvement were already mentioned in Sections 2 and 5: switching $\widetilde{\lambda}$ from the target τ to the Rayleigh quotient θ at an appropriate time (for instance based on the residual norm), and carrying out a suitable number of inner iterations.

Block methods generally may perform well if there is a cluster of eigenvalues. Geus [20] and Brandts [10, 11] have experimented with a block JD variant, and given attention to its relation with the Riccati equation.

Deflation is a technique that is used in the situation in which we have detected an eigenpair $(\lambda, \mathbf{x})$ and would like to find other pairs. An orthogonal deflation approach replaces A by $(I-\mathbf{x}\mathbf{x}^H)A(I-\mathbf{x}\mathbf{x}^H)$ after finding $\mathbf{x}$. Adding these projections is both natural in the Jacobi–Davidson context as well as attractive from a numerical point of view, since this involves orthogonal transformations on $\mathbf{x}^\perp$ that are inherently stable with respect to rounding errors. The resulting JD variant, called JDQR [17], computes a partial Schur decomposition $AQ = QS$, where Q is a $n \times k$ matrix whose columns, the "Schur vectors", are orthonormal and S is $k \times k$ upper triangular with eigenvalues of A as diagonal elements. If A is Hermitian, S is diagonal and the Schur vectors are also eigenvectors of A. If A is not Hermitian, eigenvectors of A corresponding to eigenvalues appearing on the diagonal of S are easily recovered by forming $\mathbf{u} = Q\mathbf{c}$, where $\mathbf{c}$ is a (k-dimensional) eigenvector of S. A partial generalized Schur form $AQ = ZS$, $BQ = ZT$ for the generalized eigenvalue problem $A\mathbf{x} = \lambda B\mathbf{x}$ is computed by JDQZ [58, 17, 37].

7 Variants

The JD method has several interesting variants for nonsymmetric matrices and for matrices with a special structure. In several applications the matrix A is complex symmetric, $A \in \mathbb{C}^{n\times n}$, $A^T = A$; most notably in the field of electro-magnetic simulations [1, 69]. A JD variant for complex symmetric matrices is discussed in [2].

Two-sided Jacobi–Davidson (also called bi-Jacobi–Davidson) was studied in [66, 30], see also [58]. This method is particularly suitable in applications where it is favorable to have an approximation to the left eigenvector $\mathbf{y}$ and/or an approximation to the condition number $|\mathbf{x}^*\mathbf{y}|^{-1}$ of the eigenvalue during the process. Alternating Jacobi–Davidson [30] can be seen as a combination of standard and two-sided JD. For real matrices, we know that the eigenvalues and vectors come in complex conjugate pairs; this fact is used in [80].

We mention that Jacobi–Davidson type methods have also been developed for eigenvalue problems of more general type: the generalized eigenvalue problem [58, 4, 75], constrained eigenvalue problem [75], polynomial eigenvalue problem [58, 4, 43, 75, 32, 31], nonlinear eigenvalue problem [8, 81], singular value problem [24, 25], generalized singular value problem [27], and multiparameter eigenvalue problem [29, 28]. These papers show that JD type methods may be quite attractive for "more complicated" eigenproblems, also without preconditioning. For the multiparameter eigenvalue problem for instance, the JD type methods are actually the first suitable methods for even moderately sized matrices.

8 Numerical illustration

The JD method has been successfully applied in several practical contexts, see for instance [9, 79, 22, 65, 42, 46, 23, 47, 42, 83, 15, 82, 7, 19, 55, 33]. In [45], JD is compared numerically with Davidson and Lanczos.

Here we illustrate the method by computing the smallest eigenvalues and associated eigenvectors of the matrix A resulting from the 5-point finite difference approximation of the Laplacian on the unit square with Dirichlet boundary conditions on the entire boundary, using a uniform mesh with mesh size h in both directions. We consider a JDQR flavor of JD to compute up to the 8 eigenpairs corresponding to the 8 smallest eigenvalues. For $h = 1/180$, these eigenvalues are $\lambda_1 \approx 19.7$, $\lambda_2 = \lambda_3 \approx 49.34$, $\lambda_4 \approx 78.95$, $\lambda_5 = \lambda_6 \approx 98.68$, $\lambda_7 = \lambda_8 \approx 128.3$; they do not change much for smaller mesh sizes. As we look to the eigenvalues closest to zero, we select $\tau = 0$ as "target". The tolerance on the residual norm for the convergence of the eigenpair is 10^{-12}. The restart parameters are `mindim` = 7 and `maxdim` = 14. For inner iterations, we use the stopping strategy proposed in [49], that implements the ideas in Section 4, based on an estimation of the outer residual during the course of inner iterations.

For K, we consider the algebraic multilevel (AML) preconditioning defined in [48], for which the spectral condition number (11) may be bounded independently of h. In Figure 1, we have plotted the norm of the residual in the eigenvalue equation against the number of matrix vector multiplications; the diamonds $\diamond$ correspond to true residual norms as computed at each JD (outer) step, whereas at other points the plot follows the estimation of the residual as it may be computed during inner CG iterations, see [49] for details. In the pictures to the right, each peak that follows the convergence of an eigenpair corresponds to the reinitialization of the process in order to compute the next eigenpair, according to the deflation strategy sketched above. For purpose of comparison, we also plot the evolution of the residual norm when solving a linear system $A\,\mathbf{x} = \mathbf{y}$ using the CG method with preconditioner K.

One sees that the convergence in the eigenvalue computation is essentially as fast as in the linear system solution, with some overhead proportional to the number of JD steps. Furthermore, the convergence towards the desired eigenpair(s) appears to be h-independent, which is explained by the h-independency of the condition number of $K^{-1}A$. It should also be observed that no specific difficulty arises from the multiplicity of λ_2 , λ_5 and λ_7 .

Finally, we plot in Figure 2 the evolution of the relative error for the first eigenvalue, comparing Lanczos iterations [21] with the JD method, using three different techniques to define K: algebraic multilevel preconditioning (AML) as above, the standard incomplete factorization preconditioning without fill-in (ILU) [44], and the modified version of the latter for which K and A have same row-sum (MILU) [14]. One sees that the better the preconditioner, the more impressive the gains.

9 Concluding comments

The Jacobi–Davidson method is one of successes of numerical linear algebra. Although the principle behind it (Newton) is basic, the method is very flexible and easily generalizable. A Matlab implementation of JDQR and JDQZ was written by Sleijpen [37]. Notay modified this code for his JDCG [36]. Geus [20] wrote a C implementation of JD for symmetric matrices. Pseudocodes for several of the mentioned algorithms can be found in [4].

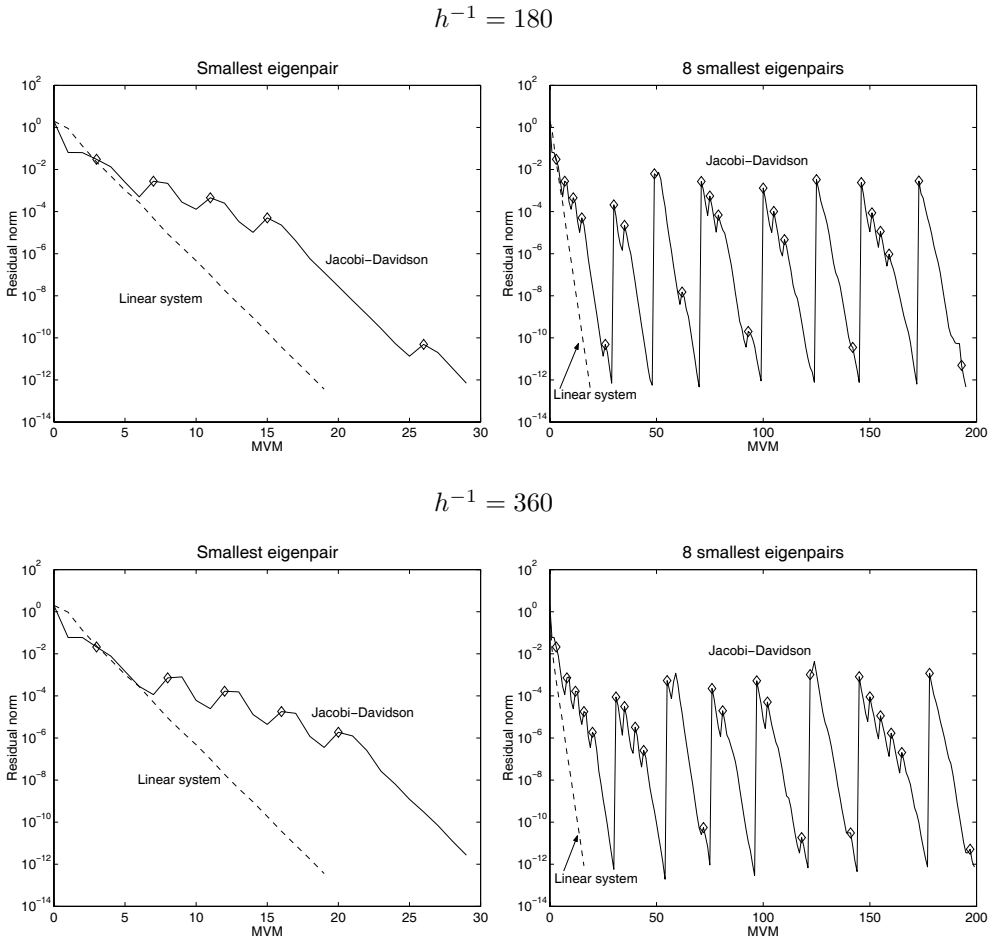

FIG. 1: Residual norm against MVM for AML preconditioning.

Acknowledgements We thank the referee for useful comments.

References

[1] ARBENZ, P., AND GEUS, R. A comparison of solvers for large eigenvalue problems occurring in the design of resonant cavities. *Numer. Linear Algebra Appl. 6*, 1 (1999), 3–16.

[2] ARBENZ, P., AND HOCHSTENBACH, M. E. A Jacobi–Davidson method for solving complex symmetric eigenvalue problems. *SIAM J. Sci. Comput. 25*, 5 (2004), 1655–1673.

[3] ARNOLDI, W. E. The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quart. Appl. Math. 9* (1951), 17–29.

[4] BAI, Z., DEMMEL, J., DONGARRA, J., RUHE, A., AND VAN DER VORST, H. A., Eds. *Templates for the Solution of Algebraic Eigenvalue Problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. A practical guide.

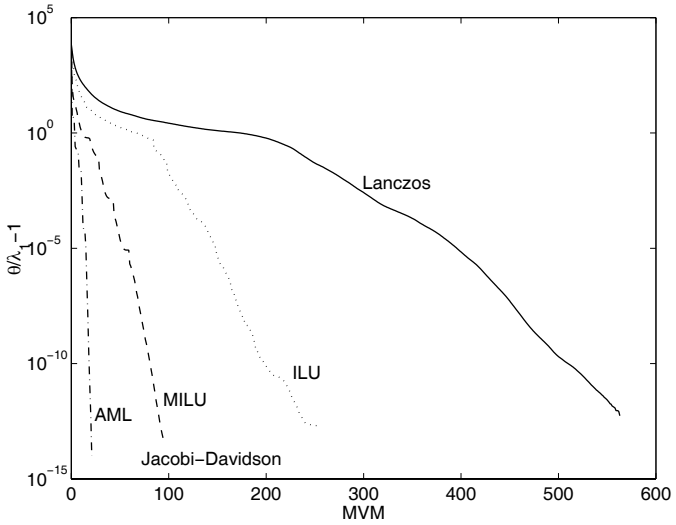

FIG. 2: Relative error for the first eigenvalue against MVM ($h^{-1} = 180$).

[5] BARRETT, R., BERRY, M., CHAN, T., DEMMEL, J., DONATO, J., DONGARRA, J., EIJKHOUT, V., POZO, R., ROMINE, C., AND VAN DER VORST, H. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[6] BENZI, M. Preconditioning techniques for large linear systems: A survey. *J. Computational Physics 182* (2002), 418–477.

[7] BERGAMASCHI, L., PINI, G., AND SARTORETTO, F. Computational experience with sequential and parallel, preconditioned Jacobi-Davidson for large, sparse symmetric matrices. *J. Comput. Phys. 188*, 1 (2003), 318–331.

[8] BETCKE, T., AND VOSS, H. A Jacobi–Davidson–type projection method for nonlinear eigenvalue problems. *Future Generation Computer Systems 20*, 3 (2004), 363–372.

[9] BOOTEN, A. G. L., FOKKEMA, D. R., SLEIJPEN, G. L. G., AND VAN DER VORST, H. A. Jacobi-Davidson methods for generalized MHD-eigenvalue problems. *Z. Angew. Math. Mech. 76*, suppl. 1 (1996), 131–134.

[10] BRANDTS, J. H. The Riccati algorithm for eigenvalues and invariant subspaces of matrices with inexpensive action. *Linear Algebra Appl. 358* (2003), 335–365.

[11] BRANDTS, J. H. Solving eigenproblems: from Arnoldi via Jacobi–Davidson to the Riccati method. In *Numerical methods and applications*, vol. 2542 of *Lecture Notes in Comput. Sci.* Springer, Berlin, 2003, pp. 167–173.

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

[13] DE STURLER, E. Improving the convergence of the Jacobi–Davidson algorithm. Preprint, Department of Computer Science, University of Illinois at Urbana-Champaign, 2002.

[14] DUPONT, T., KENDALL, R. P., AND RACHFORD, H. H. An approximate factorization procedure for solving self-adjoint elliptic difference equations. *SIAM J. Numer. Anal. 5* (1968), 559–573.

[15] FENG, Y. T. An integrated Davidson and multigrid solution approach for very large scale symmetric eigenvalue problems. *Comput. Methods Appl. Mech. Engrg. 190*, 28 (2001), 3543–3563.

[16] FOKKEMA, D. R., SLEIJPEN, G. L. G., AND VAN DER VORST, H. A. Accelerated inexact Newton schemes for large systems of nonlinear equations. *SIAM J. Sci. Comput. 19*, 2 (1998), 657–674.

[17] FOKKEMA, D. R., SLEIJPEN, G. L. G., AND VAN DER VORST, H. A. Jacobi–Davidson style QR and QZ algorithms for the reduction of matrix pencils. *SIAM J. Sci. Comput. 20*, 1 (1998), 94–125.
[18] GENSEBERGER, M., SLEIJPEN, G. L. G., AND VAN DER VORST, H. A. Using domain decomposition in the Jacobi–Davidson method. Preprint 1164, Department of Mathematics, Utrecht University, October 2000.
[19] GENSEBERGER, M., SLEIJPEN, G. L. G., AND VAN DER VORST, H. A. An optimized Schwarz method in the Jacobi-Davidson method for eigenvalue problems. In *Domain decomposition methods in science and engineering*. Natl. Auton. Univ. Mex., México, 2003, pp. 289–296.
[20] GEUS, R. *The Jacobi-Davidson algorithm for solving large sparse symmetric eigenvalue problems*. Phd thesis no. 14734, ETH Zurich, 2002. (Available at URL `http://e-collection.ethbib.ethz.ch/show?type=diss&nr=14734`).
[21] GOLUB, G. H., AND VAN LOAN, C. F. *Matrix Computations*. The John Hopkins University Press, Baltimore, Maryland, 1996. Third ed.
[22] HEEG, R. S., AND GEURTS, B. J. Spatial instabilities of the incompressible attachment-line flow using sparse matrix Jacobi-Davidson techniques. *Appl. Sci. Res. 59*, 4 (1998), 315–329.
[23] HEUVELINE, V., AND BERTSCH, C. On multigrid methods for the eigenvalue computation of nonselfadjoint elliptic operators. *East-West J. Numer. Math. 8*, 4 (2000), 275–297.
[24] HOCHSTENBACH, M. E. A Jacobi–Davidson type SVD method. *SIAM J. Sci. Comput. 23*, 2 (2001), 606–628.
[25] HOCHSTENBACH, M. E. Harmonic and refined extraction methods for the singular value problem, with applications in least squares problems. *BIT 44*, 4 (2004), 721–754.
[26] HOCHSTENBACH, M. E. Generalizations of harmonic and refined Rayleigh–Ritz. Preprint, Dept. Math., Case Western Reserve University, September 2004. Submitted.
[27] HOCHSTENBACH, M. E. A Jacobi–Davidson type method for the generalized singular value problem. Preprint, Department of Mathematics, Case Western Reserve University, September 2004. Submitted.
[28] HOCHSTENBACH, M. E., KOŠIR, T., AND PLESTENJAK, B. A Jacobi–Davidson type method for the nonsingular two-parameter eigenvalue problem. *SIAM J. Matrix Anal. Appl. 26*, 2 (2005), 477–497.
[29] HOCHSTENBACH, M. E., AND PLESTENJAK, B. A Jacobi–Davidson type method for a right definite two-parameter eigenvalue problem. *SIAM J. Matrix Anal. Appl. 24*, 2 (2002), 392–410.
[30] HOCHSTENBACH, M. E., AND SLEIJPEN, G. L. G. Two-sided and alternating Jacobi–Davidson. *Linear Algebra Appl. 358(1-3)* (2003), 145–172.
[31] HOCHSTENBACH, M. E., AND SLEIJPEN, G. L. G. Harmonic and refined extraction methods for the polynomial eigenvalue problem. Preprint, Department of Mathematics, Case Western Reserve University, September 2004. Submitted.
[32] HWANG, T.-M., LIN, W.-W., AND MEHRMANN, V. Numerical solution of quadratic eigenvalue problems with structure-preserving methods. *SIAM J. Sci. Comput. 24*, 4 (2003), 1283–1302.
[33] HWANG, T.-M., LIN, W.-W., WANG, W.-C., AND WANG, W. Numerical simulation of three dimensional pyramid quantum dot. *J. Comput. Phys. 196*, 1 (2004), 208–232.
[34] JACOBI, C. G. J. Über eine neue Auflösungsart der bei der Methode der kleinsten Quadrate vorkommende linearen Gleichungen. *Astronom. Nachr.* (1845), 297–306.
[35] JACOBI, C. G. J. Über ein leichtes Verfahren, die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen. *J. Reine und Angew. Math.* (1846), 51–94.
[36] JDCG code, available via `http://homepages.ulb.ac.be/~ynotay/`.
[37] JDQR and JDQZ codes, available via `http://www.math.uu.nl/people/sleijpen`.
[38] JIA, Z. Refined iterative algorithms based on Arnoldi's process for large unsymmetric eigenproblems. *Linear Algebra Appl. 259* (1997), 1–23.
[39] KNYAZEV, A. V., AND NEYMEYR, K. A geometric theory for preconditioned inverse iteration. III: A short and sharp convergence estimate for generalized eigenvalue problems. *Linear Algebra Appl. 358* (2003), 95–114.

[40] LANCZOS, C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Stand. 45*, 4 (1950), 255–282.

[41] LEHOUCQ, R. B., AND MEERBERGEN, K. Using generalized Cayley transformations within an inexact rational Krylov sequence method. *SIAM J. Matrix Anal. Appl. 20*, 1 (1999), 131–148.

[42] LIU, C., AND LEE, J.-F. Jacobi-Davidson algorithm and its application to modeling RF/microwave detection circuits. *Comput. Methods Appl. Mech. Engrg. 169*, 3-4 (1999), 359–375.

[43] MEERBERGEN, K. Locking and restarting quadratic eigenvalue solvers. *SIAM J. Sci. Comput. 22*, 5 (2000), 1814–1839.

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

[45] MORGAN, R. B. Preconditioning eigenvalues and some comparison of solvers. *J. Comput. Appl. Math. 123*, 1-2 (2000), 101–115. Numerical analysis 2000, Vol. III. Linear algebra.

[46] NOOL, M., AND VAN DER PLOEG, A. Parallel Jacobi-Davidson for solving generalized eigenvalue problems. In *Vector and parallel processing—VECPAR '98 (Porto)*, vol. 1573 of *Lecture Notes in Comput. Sci.* Springer, Berlin, 1999, pp. 58–70.

[47] NOOL, M., AND VAN DER PLOEG, A. A parallel Jacobi-Davidson-type method for solving large generalized eigenvalue problems in magnetohydrodynamics. *SIAM J. Sci. Comput. 22*, 1 (2000), 95–112.

[48] NOTAY, Y. Optimal order preconditioning of finite difference matrices. *SIAM J. Sci. Comput. 21* (2000), 1991–2007.

[49] NOTAY, Y. Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem. *Numer. Lin. Alg. Appl. 9* (2002), 21–44.

[50] NOTAY, Y. Convergence analysis of inexact Rayleigh quotient iteration. *SIAM J. Matrix Anal. Appl. 24* (2003), 627–644.

[51] NOTAY, Y. Is Jacobi-Davidson faster than Davidson? *SIAM J. Matrix Anal. Appl. 26*, 2 (2004/05), 522–543.

[52] OVTCHINNIKOV, E. Convergence estimates for the generalized Davidson method for symmetric eigenvalue problems. I. The preconditioning aspect. *SIAM J. Numer. Anal. 41*, 1 (2003), 258–271.

[53] OVTCHINNIKOV, E. Convergence estimates for the generalized Davidson method for symmetric eigenvalue problems. II. The subspace acceleration. *SIAM J. Numer. Anal. 41*, 1 (2003), 272–286.

[54] PARLETT, B. N. *The Symmetric Eigenvalue Problem*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. Corrected reprint of the 1980 original.

[55] ROMMES, J., VAN DER VORST, H. A., AND TER MATEN, E. J. W. Jacobi-Davidson methods and preconditioning with applications in pole-zero analysis. In *Progress in industrial mathematics at ECMI 2002*, vol. 5 of *Math. Ind.* Springer, Berlin, 2004, pp. 191–196.

[56] SAAD, Y. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2003. Second ed.

[57] SIMONCINI, V., AND ELDÉN, L. Inexact Rayleigh quotient-type methods for eigenvalue computations. *BIT 42* (2002), 159–182.

[58] SLEIJPEN, G. L. G., BOOTEN, A. G. L., FOKKEMA, D. R., AND VAN DER VORST, H. A. Jacobi–Davidson type methods for generalized eigenproblems and polynomial eigenproblems. *BIT 36*, 3 (1996), 595–633.

[59] SLEIJPEN, G. L. G., AND VAN DER VORST, H. A. A Jacobi–Davidson iteration method for linear eigenvalue problems. *SIAM J. Matrix Anal. Appl. 17*, 2 (1996), 401–425.

[60] SLEIJPEN, G. L. G., AND VAN DER VORST, H. A. The Jacobi–Davidson method for eigenvalue problems and its relation with accelerated inexact Newton schemes. In *Iterative Methods in Linear Algebra, II.* (New Brunswick, NJ, U.S.A., 1996), S. D. Margenov and P. S. Vassilevski, Eds., vol. 3 of *IMACS Series in Computational and Applied Mathematics*, IMACS, pp. 377–389.

[61] SLEIJPEN, G. L. G., AND VAN DER VORST, H. A. A Jacobi–Davidson iteration method for linear eigenvalue problems. *SIAM Review 42*, 2 (2000), 267–293.

[62] SLEIJPEN, G. L. G., VAN DER VORST, H. A., AND MEIJERINK, E. Efficient expansion of subspaces in the Jacobi–Davidson method for standard and generalized eigenproblems. *Electron. Trans. Numer. Anal. 7* (1998), 75–89.
[63] SLEIJPEN, G. L. G., AND WUBS, F. W. Exploiting multilevel preconditioning techniques in eigenvalue computations. *SIAM J. Sci. Comput. 25*, 4 (2003/04), 1249–1272.
[64] SORENSEN, D. C. Numerical methods for large eigenvalue problems. *Acta Numer. 11* (2002), 519–584.
[65] SORENSEN, D. C., AND YANG, C. A truncated RQ iteration for large scale eigenvalue calculations. *SIAM J. Matrix Anal. Appl. 19*, 4 (1998), 1045–1073.
[66] STATHOPOULOS, A. A case for a biorthogonal Jacobi-Davidson method: restarting and correction equation. *SIAM J. Matrix Anal. Appl. 24*, 1 (2002), 238–259.
[67] STATHOPOULOS, A., AND SAAD, Y. Restarting techniques for the (Jacobi–)Davidson symmetric eigenvalue methods. *Electron. Trans. Numer. Anal. 7* (1998), 163–181.
[68] STEWART, G. W. *Matrix algorithms. Vol. II*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.
[69] STREIFF, M., WITZIG, A., AND FICHTNER, W. Computing optical modes for VCSEL device simulation. *IEE Proc.-Optoelectron. 149*, 4 (2002), 166–173.
[70] SZYLD, D. B. Criteria for combining inverse and Rayleigh quotient iteration. *SIAM J. Numer. Anal.* (1988), 1369–1375.
[71] TISSEUR, F., AND MEERBERGEN, K. The quadratic eigenvalue problem. *SIAM Rev. 43*, 2 (2001), 235–286.
[72] TROTTENBERG, U., OOSTERLEE, C. W., AND SCHÜLLER, A. *Multigrid*. Academic Press, London, 2001.
[73] VAN DEN ESHOF, J. The convergence of Jacobi–Davidson iterations for Hermitian eigenproblems. *Numer. Linear Algebra Appl. 9*, 2 (2002), 163–179.
[74] VAN DER VORST, H. A. Subspace iteration for eigenproblems. *CWI Quarterly 9*, 1-2 (1996), 151–160. SMC 50 jubilee (Amsterdam, 1996).
[75] VAN DER VORST, H. A. Computational methods for large eigenvalue problems. In *Handbook of numerical analysis, Vol. VIII*. North-Holland, Amsterdam, 2002, pp. 3–179.
[76] VAN DER VORST, H. A. *Iterative Krylov Methods for Large Linear systems*. Cambridge University Press, Cambridge, 2003.
[77] VAN DER VORST, H. A. Modern methods for the iterative computation of eigenpairs of matrices of high dimension. *ZAMM Z. Angew. Math. Mech. 84*, 7 (2004), 444–451.
[78] VAN DER VORST, H. A., AND GOLUB, G. H. 150 years old and still alive: eigenproblems. In *The state of the art in numerical analysis (York, 1996)*, vol. 63 of *Inst. Math. Appl. Conf. Ser. New Ser.* Oxford Univ. Press, New York, 1997, pp. 93–119.
[79] VAN DORSSELAER, J. L. M. Computing eigenvalues occurring in continuation methods with the Jacobi-Davidson QZ method. *J. Comput. Phys. 138*, 2 (1997), 714–733.
[80] VAN NOORDEN, T. Computing a partial real ordered generalized Schur form using the Jacobi–Davidson method. Preprint 1307, Department of Mathematics, Utrecht University, September 2004.
[81] VOSS, H. A Jacobi–Davidson method for nonlinear and nonsymmetric eigenproblems. Tech. rep., Institute of Mathematics, Hamburg University of Technology, 2004. Submitted.
[82] WANG, W., HWANG, T.-M., LIN, W.-W., AND LIU, J.-L. Numerical methods for semiconductor heterostructures with band nonparabolicity. *J. Comput. Phys. 190*, 1 (2003), 141–158.
[83] YANG, C., PEYTON, B. W., NOID, D. W., SUMPTER, B. G., AND TUZUN, R. E. Large-scale normal coordinate analysis for molecular structures. *SIAM J. Sci. Comput. 23*, 2 (2001), 563–582. Copper Mountain Conference (2000).