ELSEVIER Linear Algebra and its Applications 286 (1999) 45-68

LINEAR ALGEBRA AND ITS APPLICATIONS

Restrictions on implicit filtering techniques for orthogonal projection methods

G. De Samblanx *, A. Bultheel

Department of Computer Science, Katholeike Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium

Received 24 April 1997; accepted 13 July 1998

Submitted by P. Van Dooren

Abstract

We consider the class of the Orthogonal Projection Methods (OPM) to solve iteratively large eigenvalue problems. An OPM is a method that projects a large eigenvalue problem on a smaller subspace. In this subspace, an approximation of the eigenvalue spectrum can be computed from a small eigenvalue problem using a direct method. Examples of OPMs are the Arnoldi and the Davidson method. We show how an OPM can be restarted – implicitly and explicitly. This restart can be used to remove a specific subset of vectors from the approximation subspace. This is called explicit filtering. An implicit restart can also be combined with an implicit filtering step, i.e. the application of a polynomial or rational function on the subspace, even if inaccurate arithmetic is assumed. However, the condition for the implicit application of a filter is that the rank of the residual matrix must be small.

AMS classification: 65F15

Keywords: Davidson; Implicitly restarted Arnoldi; Standard eigenvalue problem; Shift-invert

1. Introduction

Consider the problem of finding a limited set of solutions to the eigenvalue problem

* Corresponding author. E-mail: Gorik.DeSamblanx@cs.kuleuven.ac.be.

0024-3795/99/$ – see front matter
PII: S0024-3795(98)10153-2

$$Ax = \lambda x, \qquad A \in \mathbb{C}^{n\times n}.$$

If the dimension of this problem is very large or if A is sparse, then it cannot be tackled using a dense or direct method, such as the QR method, due to time and memory limitations. Therefore, a large family of iterative solvers has been derived that are able to find one or more specific eigenvalues of A, e.g. the rightmost eigenvalues. Most of these methods build iteratively a subspace basis V_k of dimension k on which the eigenvalue problem is projected. The eigenvalues are then approximated from this small projected problem, using a direct solver. We call these methods Orthogonal Projection Methods (OPM).

However, if the size of the subspace V_k becomes too large, then the eigenvalue solver slows down (or it can reach a memory limit). The method then has to be restarted. The restart can be worked out explicitly, deriving new starting conditions from the unsatisfying solution. It can also be done implicitly, by reducing the size k of the subspace basis to $k - p$ and thus removing a subset of the information in V_k. A well known example of an implicitly restarted eigenvalue solver is the Implicitly Restarted Arnoldi method (IRA) of [1].

Generally, an implicit restart is cheaper and more stable than an explicit restart. It has the additional advantage that it often implements an *implicit filter* on the subspace basis. If an implicit filter is applied, then the new basis can be seen as a copy of the old basis, multiplied by a polynomial (or rational) function in A. Hence, by implementing implicitly a filter, a restart procedure implements a form of subspace iteration. Spence and Meerbergen [2] noticed that the implicit filter can be used in order to remove spurious eigenvalues from the projected problem.

The idea of implicitly filtering is based on the assumption that exact arithmetic is used. For example, if the Arnoldi method is used on a shifted an inverted matrix $(A - \sigma I)^{-1}$, then the accuracy of the implicit filter that is implemented by the implicitly restarted Arnoldi algorithm depends on the accuracy of the solved systems with $A - \sigma I$.. There are many methods, such as Davidson, Jacobi–Davidson or even the Rational Krylov Sequence method (RKS), that do not require the use of accurate system solutions. For these methods, a restarting procedure cannot implement in general an implicit filter.

It is the aim of this text to show how a general eigenvalue solver can be filtered implicitly in combination with a restart. In fact, we explain why it is hard to do so if the eigenvalue solver uses an inexact method to solve the linear systems involved. It is hard because it is expensive or even impossible in some cases.

The problem is that implicit filtering always comes with the significant cost of the computational work of one or more iteration steps, i.e. the loss of some computed basis vectors. If the purpose of the restart is to reduce the size of the approximating subspace basis V_k, then this cost is the key behind the reduction, and the loss of basis vectors is not considered disadvantageous. But if e.g. spurious eigenvalues must be filtered away, then the cost of more than one iteration step can not be considered as a 'fair deal'.

Let us give an example. Each step of the Arnoldi method requires essentially a matrix vector multiplication (plus some overhead that we neglect here). Each of these steps adds a vector to the basis V_k. If we reduce the size of V_k with p, using the IRA algorithm, then we throw away the work of p iteration steps. The indirect cost of the restart is thus p matrix vector products. Suppose that the IRA algorithm reduces the subspace V_k to a subspace V_{k-p}^+ of size $k-p$ such that it implements implicitly a polynomial filter $\phi_p(A) = \prod_{i=1}^{p}(A-\sigma_i I)$

$$\mathcal{R}\left(V_{k-p}^+\right) = \mathcal{R}\left(\phi_p(A)V_{k-p}\right).$$

The zeros σ_i of the filter polynomial are provided by the user. They are chosen on the basis of on the knowledge that is present in the basis V_k. They allow us to filter out less relevant information and to keep the relevant subspace. Clearly, the p matrix vector products that are lost by removing p basis vectors are implicitly recovered by the filter polynomial (of degree p). Since the polynomial consists of p multiplications with A, we can say that as many matrix vector products are recovered as lost.

One could also apply the function $\phi_p(A)$ *explicitly* on the subspace basis, but this would cost $p(k-p)$ matrix vector products (plus the orthogonalisation of V_{k-p}^+). Therefore an explicit application of the filter is often ruled out as being too expensive.

For more general methods, the cost of an implicit product with a matrix polynomial (rational) function of degree p, comes with a cost of lp basis vectors, with $l > 1$, instead of the p basis vectors for IRA. These methods loose lp vectors but only recover the work of p iteration steps: so there is more lost than recovered.

We prove in this text that the factor l in the cost of one implicit filtering step corresponds to the rank of the residual matrix. This matrix spans the subspace of the residuals of all approximate eigenvectors in the column range of V_k. For Arnoldi's method and for the RKS method, this rank is one. For block variants of these methods, the rank equals the block size.

The paper consists of two parts. First, we prove a simple recurrence relation between the residual matrices of subsequent steps of an eigenvalue solver. This recursion is used to define the concept of a *rank conservative eigenvalue solver*. That is a solver for which the rank of the residual does not grow when the size of V_k grows. It can be shown that all methods that use a spectral transformation (i.e. the solution of a linear system with a matrix $A-\sigma I$) within their iteration, are rank conservative, and vice versa.

Then it is shown how rank conservative solvers can be filtered implicitly. This implicit filtering comes with the removal of l basis vectors, where l is equal to the residual rank. An algorithm is described that does so, based on a shifted QR decomposition of the projection matrix. The analysis assumes exact

arithmetic, but generalisations to a floating point context with rounding errors are made at each step.

Plan of the paper: The paper is structured as follows. In Section 1, we recall some basic facts on projectors and on orthogonal projection methods. Section 2 derives a recurrence relation for the residual matrix. The concept of rank conservative eigenvalue solvers is introduced. Section 3 is concerned with the restarting of the eigenvalue solver. We show how solvers can be filtered explicitly and implicitly. Some conclusions are presented in Section 4.

Notation: In this text, matrices are denoted by upper case characters, vectors by lower case characters and complex numbers by Greek characters. The Hermitian transpose of a matrix or vector is denoted by V^* or v^* and $\|\cdot\|$ denotes the 2-norm. The identity matrix is denoted by I and the kth unit vector by e_k. The column range of a matrix V is denoted by $\mathcal{R}(V)$.

1.1. Preliminary definitions

The analysis of an orthogonal projection method makes use of projection matrices. Let us recall some facts about projection matrices that are employed in the following sections. These properties are used to show that the difference between the solutions of an eigenvalue solver in exact arithmetic and the solutions in floating point arithmetic, are acceptably small.

Definition 1.1. Given a matrix $\mathcal{P} \in \mathbb{C}^{n\times n}$. $\mathcal{P}$ is called a projection matrix if $\mathcal{P}\mathcal{P} = \mathcal{P}$. If $\mathcal{P} = \mathcal{P}^*$, then $\mathcal{P}$ is called an orthogonal projection matrix. Otherwise, $\mathcal{P}$ is called an oblique projection matrix.

If we apply $\mathcal{P}$ to the columns of a matrix $V_k \in \mathbb{C}^{n\times k}$, then $\mathcal{P}V_k$ is the projection of V_k on the column space of $\mathcal{P}$. For an orthogonal projection matrix, $\mathcal{P}V_k$ is orthogonal to $(I - \mathcal{P})V_k$. An orthogonal projection matrix can, using its singular value decomposition, always be written as $\mathcal{P} = QQ^*$, with $Q^*Q = I_m$. The orthogonal matrix Q then forms an orthogonal basis for the column space of $\mathcal{P}$. Inversely, we will denote the dual projection matrices that are generated by a matrix W by

$$\mathcal{P}_W \equiv QQ^*, \qquad \mathcal{P}_W^{\perp} \equiv I - \mathcal{P}_W,$$

where $\mathcal{R}(Q) = \mathcal{R}(W)$ and $Q^*Q = I$.

The goal of an iterative eigenvalue solver is to produce an invariant subspace for a given matrix A. For the iterative eigenvalue solver to be competitive with the 'direct' eigenvalue solvers, such as the QR method, the dimension of this subspace must be much smaller than the dimension of the eigenproblem itself. In general, the algorithm generates an orthogonal matrix $V_k = [v_1, \ldots, v_k]$. The columns of this matrix span the subspace on which A is then projected.

In general, the projection is described by a pair of $k \times k$ matrices (K_k, L_k) and a residual matrix F_k. It is summarised in an equation

$$AV_kL_k = V_kK_k + (\tau_kA - \sigma_kI)F_k, \quad \text{with } V_k^*F_k = 0, \tag{1}$$

where τ_k, $\sigma_k \in \mathbb{C}$. This equation is called the *projection equation*. We assume that σ_k/τ_k is not an eigenvalue of A, so $(\tau_kA - \sigma_kI)$ is non-singular. In practice, a projection equation will emerge in one of two special forms. The first one corresponds to Eq. (1) with $L_k = I$, $K_k = G_k$, $\tau_k = 0$ and $\sigma_k = -1$, such that

$$AV_k = V_kG_k + F_k. \tag{2}$$

The matrix $G_k \equiv V_k^*AV_k$ describes explicitly the projection of A on $\mathcal{R}(V_k)$. The eigenvalues of G_k are the Ritz values and they approximate eigenvalues of A.

For some methods, the projection equation is written by use of a pair of matrices (K_k, L_k), a shift $\sigma_k \in \mathbb{C}$ and normalising $\tau_k = 1$ in the following form

$$AV_kL_k = V_kK_k + (A - \sigma_kI)F_k. \tag{3}$$

This notation useful for methods where the shift σ_k is different for different k, e.g. for the RKS algorithm. Eq. (3) can be rewritten in the form of Eq. (2), but for a matrix $(A - \sigma_kI)^{-1}$:

$$(A - \sigma_kI)^{-1}V_k = V_kL_k(K_k - \sigma_kL_k)^{-1} + F_k(K_k - \sigma_kL_k)^{-1}. \tag{4}$$

One can prove that $(K_k - \sigma_kL_k)$ is non-singular unless V_k contains an exact eigenvalue of A or unless L_k is singular.

The Ritz values of $(A - \sigma_kI)^{-1}$ are then given by the eigenvalues of the small generalised eigenvalue problem $(L_k, K_k - \sigma_kL_k)$. These values are the *Harmonic Ritz values* of A [3]. In this text, we will derive most of the properties on the residual matrix assuming that Eq. (2) holds and then generalise them to Eq. (3).

The framework in which we will view the iterative eigenvalue solvers in this text is the framework of the *orthogonal projection methods*. Algorithm 1.1 defines a template that covers the projection methods that generate the relation (1). Notice that the most important and distinctive step, the computation of w_k, is *not* specified by this template. Therefore, we underline that this template lies far from an algorithm in pseudo code that would be ready-to-implement. It only shows which entries must be computed to find a solution and at what stage of the algorithm they can be computed. For many methods, these matrices come for 'free' and they must not be computed explicitly.

Algorithm 1.1. *Orthogonal Projection Method (template)*

1. *Given* A, $V_1 = [v_1]$, $\|v_1\| = 1$
2. *For* $k = 1, 2, 3, \ldots$
   2.1. *Compute* $G_k = V_k^*AV_k$ *or compute* L_k *and* K_k.

2.2. *Compute* F_k *(if needed).*
2.3. *Compute* (θ_k, z_k) *from*
$G_k z_k = \theta_k z_k$ *or*
$L_k z_k = \theta_k (K_k - \sigma_k L_k) z_k$.
2.4. *Set* $y_k \leftarrow V_k z_k$.
2.5. *If convergence then exit.*
2.6. *Compute* w_k.
2.7. *Set* $v_{k+1} \leftarrow \mathscr{P}_{V_k}^{\perp} w_k / \|\mathscr{P}_{V_k}^{\perp} w_k\|$.
2.8. *Set* $V_{k+1} \leftarrow [V_k, v_{k+1}]$.

Since the orthogonality of V_k is used implicitly, we must take care that this property is always true – to working precision. Therefore, reorthogonalisation must be considered [4].

The residual of an arbitrary unit vector and the residial of an orthogonal matrix is defined as follows.

Definition 1.2. Given a matrix $A \in \mathbb{C}^{n\times n}$ and a vector $u \in \mathbb{C}^n$ with $\|u\| = 1$. The residual $r(A,u)$ of u is given by $r(A,u) \equiv Au - (u^*Au)u = \mathscr{P}_u^{\perp} Au$. If $V \in \mathbb{C}^{n\times k}$ is orthogonal, then $R(A,V) \equiv \mathscr{P}_V^{\perp} AV$.

We can write for Eq. (2) that $R(A,V_k) = \mathscr{P}_{V_k}^{\perp} AV_k = AV_k - V_k V_k^* AV_k = AV_k - V_k G_k = F_k$. For Eq. (3) however, we apply $\mathscr{P}_{V_k}^{\perp}$ and see that $R(A,V_k) = \mathscr{P}_{V_k}^{\perp} AV_k = \mathscr{P}_{V_k}^{\perp}(A-\sigma_k I)F_k L_k^{-1}$. In the sequel, we shall use $\tilde{F}_k$ to denote $R(A,V_k)$ in general. Thus in case of Eq. (2), $\tilde{F}_k = F_k$, while in case of Eq. (3), $\tilde{F}_k$ and F_k are related by

$$\tilde{F}_k = \mathscr{P}_{V_k}^{\perp}(A-\sigma_k I)F_k L_k^{-1}.$$

By using Eq. (4), we could obtain an explicit expression for F_k, namely

$$F_k = R((A-\sigma_k I)^{-1}, V_k)(K_k - \sigma_k L_k),$$

which is less attractive to deal with than $R(A,V_k)$. If $y = V_k z$ is a Ritz vector in V_k, then it is easy to see that $r(A,y) = \tilde{F}_k z$.

There are different possibilities to check the convergence at step 2.5., depending on the individual method. Most commonly, some measure for the residual norm $\|r(A,y_k)\| = \|Ay_k - \theta y_k\|$ is used, e.g. $|e_k^* z_k| \|\mathscr{P}_{V_k}^{\perp} w_k\| / \|w_k\|$.

Example 1.1. Many well known iterative methods for solving eigenvalue problems can be fitted into this scheme.

- *Arnoldi's method* [5]: If we choose $w_k = Av_k$, then we get Arnoldi's method. The matrix G_k is upper Hessenberg and its kth column g_k contains the orthogonalisation coefficients of the kth iteration step: $w_k = V_{k+1} g_k$. Moreover, $F_k = [0 \cdots 0\ f_k]$, with $f_k = \|f_k\| v_{k+1}$. The next vector in the Arnoldi iteration is equal to the residual of Eq. (2) – which is called the Arnoldi

equation. If A is symmetric, then G_k is a tridiagonal matrix and the method is the symmetric Lanczos procedure [6].

- *RKS* [7–9]: If we choose $w_k = (A - \sigma_k I)^{-1} V_k t_k$, with $t_k \in \mathbb{C}^k$ some *continuation vector*, we get the (RKS) sequence method. RKS corresponds to an OPM that builds Eq. (3), where L_k contains the orthogonalisation coefficients and $K_k = L_k \operatorname{diag}(\sigma_i) + T_k$ is an upper Hessenberg matrix. T_k is the upper triangular matrix that collects the continuation vectors. As for the Arnoldi algorithm, F_k only contains one non-zero column. For RKS, the subdiagonal elements of L_k will always be non-zero – unless the method has converged to some solution.
- *Davidson* [10,11]: If we choose $w_k = (A - \sigma_k I)^{-1}(A - \theta_k I)y_k = (A - \sigma_k I)^{-1} r(A, y_k)$, then Algorithm 1.1 corresponds to Davidsons method. The Davidson method is well suited for use with an inexact linear system solver. If the system $(A - \sigma_k I)w_k = r(A, y_k)$ is solved exactly, then this method corresponds to an extended RKS method and $\operatorname{rank}(F_k) = 1$. If the linear system is solved inexactly, e.g. with an iterative method, then $\operatorname{rank}(F_k) > 1$.
- *Jacobi–Davidson* [12]: If w_k is computed as $\mathscr{P}_{y_k}^{\perp}(A - \theta_k I)\mathscr{P}_{y_k}^{\perp} w_k = -(A - \theta_k I) \times y_k$, by use of an iterative system solver, then the OPM is the Jacobi–Davidson algorithm. This method is a.o. an extention of the Davidson method. A short comparison between both methods can be found in [12,13]. Notice that for both Davidson variants, G_k is no longer an upper Hessenberg matrix. As for Davidson's method, it holds that if the system is solved in exactly, then $\operatorname{rank}(F_k) > 1$.

We define the (numerical) rank of a matrix as follows.

Definition 1.3. Given a matrix $F \in \mathbb{C}^{m \times n}$, with singular values $\sigma_1, \sigma_2, \ldots$ then define the rank of F and its numerical ε-rank as

$$\operatorname{rank}(F) = \#\{\sigma_i \mid \sigma_i \neq 0\} \text{ and } \operatorname{rank}(F, \varepsilon) = \#\{\sigma_i \mid \sigma_i > \varepsilon\}. \tag{5}$$

In order to understand the correspondence of an OPM to its numerical implementation, we show how a projection matrix acts on the (numerical) rank of a matrix and on its singular values.

Lemma 1.1. *Given a matrix $F \in \mathbb{C}^{n \times k}$ of rank k, an orthogonal projection matrix $\mathscr{P}$ and a vector $f \in \mathbb{C}^n$. Let $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_k$ be the singular values of F.*

1. *If $\sigma'_1 \geqslant \sigma'_2 \geqslant \cdots \geqslant \sigma'_k$ are the singular values of $\mathscr{P}F$, then $\sigma'_1 \leqslant \sigma_1$, $\sigma'_2 \leqslant \sigma_2, \ldots, \sigma'_k \leqslant \sigma_k$.*
2. *Given a vector $v = Ft + e$ for some $t \in \mathbb{C}^k$, with $\|t\| = 1$ and $\mathscr{P}v = 0$. If $\sigma_p < \|e\|$, then $\sigma'_{p-1} < \|e\|$. Moreover, $\operatorname{rank}(\mathscr{P}F, \|e\|) \leqslant p$.*

3. *If* $\sigma_1'' \geqslant \cdots \geqslant \sigma_{k+1}''$ *denote the singular values of* $[F,\ f]$, *then*

$$\sigma_{k+1}'' \leqslant \frac{1}{\sqrt{1+h^*h}}\|\mathscr{P}_F^{\perp} f\|,$$

$$\sigma_i'' \leqslant \kappa\sigma_i + \|\mathscr{P}_F^{\perp} f\|,$$

$$\sigma_i'' \leqslant \sqrt{\sigma_i^2 + \|f\|^2},\ i = 1,\ldots,k,$$

where $h \in \mathbb{C}^k$ *is defined by* $\mathscr{P}_F^{\perp} f = f - Fh$ *and* $\kappa = 1 + \|h\|$.

Proof. For this proof, we use the fact that the singular values of a $k \times n$ matrix F are

$$\sigma_i = \max_{X \in \mathscr{X}_i} \min_{x \in X} \frac{\|Fx\|}{\|x\|} = \min_{X \in \mathscr{X}_{k-i+1}} \max_{x \in X} \frac{\|Fx\|}{\|x\|},$$

where $\mathscr{X}_i$ is the set of subspaces of dimension i [14,15]. Therefore,

$$\sigma_1 = \max_x \frac{\|Fx\|}{\|x\|} \text{ and } \sigma_k = \min_x \frac{\|Fx\|}{\|x\|}.$$

We also rely on the fact that

$$\max_{X \in \mathscr{X}_i} \min_{x \in X} \{\cdot\} \leqslant \max_{X \in \mathscr{X}_i} \min_{x \in X, x \neq 1} \{\cdot\} = \max_{X \in \mathscr{X}_i, 1 \notin X} \min_{x \in X} \{\cdot\} \leqslant \max_{X \in \mathscr{X}_{i-1}} \min_{x \in X} \{\cdot\},$$

i.e. the maximum over a 'smaller' set $\mathscr{X}_{i-1}$ is larger than the maximum over the original set $\mathscr{X}_i$.

1. The first property is very well known:

$$\sigma_i' = \max_{X \in \mathscr{X}_i} \min_{x \in X} \frac{\|\mathscr{P}Fx\|}{\|x\|} \leqslant \max_{X \in \mathscr{X}_i} \min_{x \in X} \frac{\|Fx\|}{\|x\|} = \sigma_i.$$

2. For the $p-1$-th singular value, it holds that

$$\begin{aligned}\sigma_{p-1}' &= \min_{X \in \mathscr{X}_{k-p+2}} \left\{ \max_{x \in X, \|x\|=1} \|\mathscr{P}Fx\| \right\} \\ &\leqslant \min_{X \in \mathscr{X}_{k-p+2}} \left\{ \max_{x \in X\{t\}, \|x\|=1} \{\|\mathscr{P}Fx\|\}, \|\mathscr{P}Ft\| \right\} \\ &\leqslant \min_{X \in \mathscr{X}_{k-p+1}} \left\{ \max_{x \in X, \|x\|=1} \|\mathscr{P}Fx\|, \|e\| \right\} \leqslant \min\{\sigma_p, \|e\|\}.\end{aligned}$$

3. Say $f = Fh + g$ and let $y = [x^* \ \alpha^*]^*$, where $x \in \mathbb{C}^k$. We normalise $\|y\| = 1$, thus $|\alpha|^2 = 1 - x^*x$. There exists a $y_0 = [x_0^* \ \alpha_0^*]^*$, such that $x_0 + \alpha_0 h = 0$. Without loss of generality, we may assume that the α are real and positive. Hence, $\alpha_0^2 = 1 - x_0^* x_0 \leqslant 1$ and

$$\sigma''_{k+1} = \min_{x^*x+\alpha^2=1} \|Fx + f\alpha\| = \min_{x^*x+\alpha^2=1} \|F(x+\alpha h) + \alpha \mathscr{P}_F^{\perp} f\|$$
$$\leqslant \min\left\{\alpha_0 \|\mathscr{P}_F^{\perp} f\|, \min_{x+\alpha h \neq 0} \|F(x+\alpha h)\| + \alpha \|\mathscr{P}_F^{\perp} f\|\right\} \leqslant \alpha_0 \|\mathscr{P}_F^{\perp} f\|.$$

For the other singular values, we can write

$$\sigma''_i = \max_{X \in \mathscr{T}_i} \min_{x \in X,\ x} \|Fx + \alpha f\|$$
$$\leqslant \max_{X,\ x} \min \|F(x+\alpha h)\| + \alpha \|\mathscr{P}_F^{\perp} f\|$$
$$\leqslant \max\left\{\min\left\{\alpha_0 \|\mathscr{P}_F^{\perp} f\|, \min_{x \in X \setminus \{h\}} \frac{\|F(x+\alpha h)\|}{\|x+\alpha h\|} \|x+\alpha h\| + \alpha \|\mathscr{P}_F^{\perp} f\|\right\}\right\}$$
$$\leqslant \max\left\{\min\left\{\alpha_0 \|\mathscr{P}_F^{\perp} f\|, \min_{x \in X \setminus \{h\}} \frac{\|F(x+\alpha h)\|}{\|x+\alpha h\|} \kappa + \|\mathscr{P}_F^{\perp} f\|\right\}\right\}$$
$$\leqslant \sigma_i \kappa + \|\mathscr{P}_F^{\perp} f\|.$$

Finally,

$$\sigma''_i \leqslant \min_X \max_{x,\ x} \|Fx\| + \alpha \|f\| \leqslant \min_X \max_x \frac{\|Fx\|}{\|x\|} \sqrt{1-\alpha^2} + \alpha \|f\|$$
$$\leqslant \max_x \sigma_i \sqrt{1-\alpha^2} + \alpha \|f\|.$$

If $\alpha = 0$, then this value is equal to $\|f\|$; if $\alpha = 1$, then it is σ_i. The extremum inside the interval $[0,1]$ is found at $\alpha^2 = \|f\|^2/(\sigma_i^2 + \|f\|^2)$. The value of the maximum is then $\sqrt{\sigma_i^2 + \|f\|^2}$. □

Note that the conclusions of this lemma only hold for orthogonal projection matrices. When oblique projectors $\mathscr{P}_{\text{obq}}$ are used, it is possible that $\|\mathscr{P}_{\text{obq}} f\| \gg \|f\|$. In practice, this will be merely an exception, but it keeps us from drawing broad conclusions about the generalised eigenvalue problem, because the generalised eigenvalue problem makes use of oblique projector matrices. The standard eigenvalue problem only needs orthogonal projectors. Consider some extremal cases of Lemma 1.1(3):

1. If $f \in \mathscr{R}(F)$ then $\|\mathscr{P}_F^{\perp} f\| = 0$ and the matrix $[F, f]$ will have a new singular value that is equal to zero. If $\|\mathscr{P}_F^{\perp} f\| = \varepsilon$ is small, then the new singular value will be at most equal to ε. The other singular values will grow in proportion to $\|f\|$.
2. If $f \perp \mathscr{R}(F)$ then $h = 0$. We then expect a new singular value that is equal to $\|f\|$. The other singular values do not change. If $\|h\|$ is small, then the new singular value will be approximately equal to $\|\mathscr{P}_F^{\perp} f\| \simeq \|f\|$. The other singular values will not change much.
3. If $f \perp \mathscr{R}(F)$, but $\|f\|$ is small, then the singular values will not grow much either. The new singular value will be approximately equal to $\|f\|$.

2. Monitoring the residual matrix

We prove in this section a recurrence relation between the residual matrices of different steps of the eigenvalue solver. This relation is then used to show which orthogonal projection methods are rank conservative and thus may be considered for implicit filtering.

2.1. A recursion on the residual matrix

The subspace V_k depends directly on the computation of the vectors $w_1, \ldots, w_{k-1}$. As for the Arnoldi algorithm, these computations can often be written as a recurrence relation. The same can be done for the residual matrix.

Lemma 2.1. *Let* $V_{k+1} = [V_k \; v_{k+1}]$ *and* $V_{k+1}^* V_{k+1} = I$, *then*

$$R(A, V_{k+1}) = \mathscr{P}_{V_{k-1}}^{\perp} [R(A, V_k) \quad r(A, v_{k-1})]. \tag{6}$$

Proof. The proof is quite trivial. Since $R(A, V_{k+1}) = \mathscr{P}_{V_{k-1}}^{\perp} A V_{k+1}$ $= [\mathscr{P}_{V_{k-1}}^{\perp} A V_k \; \mathscr{P}_{V_{k-1}}^{\perp} A v_{k+1}]$ and since $\mathscr{P}_{V_{k-1}}^{\perp} = \mathscr{P}_{V_k}^{\perp} \mathscr{P}_{v_{k-1}}^{\perp} = \mathscr{P}_{v_{k-1}}^{\perp} \mathscr{P}_{V_k}^{\perp}$, we have

$$\begin{aligned} R(A, V_{k+1}) &= \left[\mathscr{P}_{V_{k-1}}^{\perp} R(A, V_k) \quad \mathscr{P}_{V_k}^{\perp} r(A, v_{k-1}) \right] \\ &= \mathscr{P}_{V_{k-1}}^{\perp} [R(A, V_k) \quad r(A, v_{k-1})]. \end{aligned}$$

From this lemma, we can immediately derive a recurrence relation for the residual matrices.

Theorem 2.2. *If* $\tilde{F}_k = R(A, V_k)$ *is the matrix of residuals in Algorithm* 1.1, *then it satisfies the recursion*

$$\tilde{F}_{k+1} = \mathscr{P}_{V_{k-1}}^{\perp} [\tilde{F}_k \quad r(A, v_{k-1})]. \tag{7}$$

Recall that in the case of Eq. (2), $\tilde{F}_i = F_i$, $i = k, k+1$. *If* F_k *emerges in Eq.* (3), *then Eq.* (7) *says that*

$$\mathscr{P}_{V_{k-1}}^{\perp} (A - \sigma_{k+1} I) F_{k+1} = \mathscr{P}_{V_{k-1}}^{\perp} [\tilde{F}_k \quad r(A, v_{k+1})] L_{k+1}.$$

Proof. The recurrence (7) is given in the previous lemma. The relation for F_k follows from combining the relation $\tilde{F}_i L_i = \mathscr{P}_{V_i}^{\perp} (A - \sigma_i I) F_i$, $i = k, k+1$ with Eq. (7). □

The interpretation of Theorem 2.2 reveals information about the possible convergence properties of the algorithm. Indeed, by (7) the new residual matrix consists of the old residual orthogonalised to v_{k+1} and of the residual of the new vector orthogonalised to V_k. If the residual norm $\|F_k\|$ or $\|\tilde{F}_k\|$ is small, then the

method converges. In order to get a small residual norm $\|F_k\|$, one should try to satisfy as much as possible these conditions: $\mathscr{R}(\tilde{F}_k) \subset \mathscr{R}(V_{k+1})$ and $r(A, v_{k+1}) \in \mathscr{R}(V_{k+1})$. The first condition is optimally fulfilled by the Arnoldi method, the second by a Davidson type approach.

Concerning the rank of the residual matrix, we first observe that in case Eq. (2) $F_k = \tilde{F}_k = R(A, V_k)$, so $\operatorname{rank}(F_k) = \operatorname{rank}(\tilde{F}_k)$. In case Eq. (3) however, it may happen that $\operatorname{rank}(F_k) \geqslant \operatorname{rank}(\tilde{F}_k)$, because $\tilde{F}_k = \mathscr{P}_{V_k}^{\perp}(A - \sigma_k I)F_k L_k^{-1}$ (and $\operatorname{rank}(F_k) = \operatorname{rank}((A - \sigma_k I)F_k L_k^{-1})$). The strict inequality will only hold in very specific situations. However, generically we also have $\operatorname{rank}(F_k) = \operatorname{rank}(\tilde{F}_k)$ in case of Eq. (3) and thus for the general case Eq. (1). We will assume further in this paper that $\operatorname{rank}(F_k) = \operatorname{rank}(\tilde{F}_k)$ holds for the general case.

2.2. Rank conservative eigensolvers

If algorithms have the property that the rank of their residual matrix does not grow, i.e. $\operatorname{rank}(F_k) \geqslant \operatorname{rank}(F_{k+p})$, then they are called *rank conservative*. If the numerical rank is constant for the method, then they are called *ε-rank conservative*. However, due to rounding errors, the numerical rank of the residual can increase slightly, even for a theoretically ε-rank conservative method.

Definition 2.1. If for an OPM, based on Algorithm 1.1, it holds generically (i.e. for almost any matrix A) that $\operatorname{rank}(F_{k+1}) \leqslant \operatorname{rank}(F_k)$, $k \geqslant 1$, then the method is called rank-conservative.

If $\operatorname{rank}(F_{k+1}, \varepsilon) \leqslant \operatorname{rank}(F_k, \varepsilon')$, with $\varepsilon' = O(\varepsilon)$, then the method is called ε-rank conservative.

From Theorem 2.2, we derive a condition that a rank conservative solver must fulfill.

Lemma 2.3. *Let* $\tilde{F}_k = R(A, V_k)$. *In exact arithmetic, it holds that* $\operatorname{rank}(\tilde{F}_{k+1}) \leqslant \operatorname{rank}(\tilde{F}_k)$ *iff*

$$r(A, v_{k+1}) \in \mathscr{R}(\tilde{F}_k) \cup \mathscr{R}(V_{k+1}) \quad \text{or} \quad v_{k+1} \in \mathscr{R}(\tilde{F}_k),$$

where $\tilde{F}_k = R(A, V_k)$. *Numerically, we can say that if*

$$\|\mathscr{P}_{\tilde{F}_k}^{\perp}\mathscr{P}_{V_{k+1}}^{\perp} r(A, v_{k+1})\| < \varepsilon \text{ or } \|\mathscr{P}_{\tilde{F}_k}^{\perp} v_{k+1}\| < \varepsilon,$$

then there exists a $\varepsilon' = O(\varepsilon)$ *such that* $\operatorname{rank}(\tilde{F}_{k+1}, \varepsilon') < \operatorname{rank}(\tilde{F}_k, \varepsilon)$.

Proof. Since Theorem 2.2 shows that $\tilde{F}_{k+1} = [\mathscr{P}_{v_{k+1}}^{\perp}\tilde{F}_k \;\; \mathscr{P}_{\tilde{F}_k}^{\perp} r(A, v_{k+1})]$, we have $\operatorname{rank}(\tilde{F}_{k+1}) \leqslant \operatorname{rank}(\tilde{F}_k)$ if $\mathscr{P}_{\tilde{F}_k}^{\perp} r(A, v_{k+1}) \in \mathscr{R}(\tilde{F}_k)$ or if $\operatorname{rank}(\mathscr{P}_{v_{k+1}}^{\perp}\tilde{F}_k) = \operatorname{rank}(\tilde{F}_k) - 1$. The first condition is true iff $r(A, v_{k+1}) \in \mathscr{R}(\tilde{F}_k) \cup \mathscr{R}(V_{k+1})$, the second one

holds iff $v_{k+1} \in \mathscr{R}(\tilde{F}_k)$. Using Lemma 1.1, we can prove the second part. Say $\sigma_1 \geqslant \sigma_2 \geqslant \ldots$ are the singular values of $\tilde{F}_k$ and $\sigma'_1 \geqslant \sigma'_2 \geqslant \cdots$ are the singular values of $[\tilde{F}_k \ \mathscr{P}_{\tilde{V}_k}^{\perp} r(A, v_{k+1})]$. Combining the first condition with Lemma 1.1, gives

$$\sigma'_{k+1} \leqslant \|\mathscr{P}_{\tilde{F}_k}^{\perp} \mathscr{P}_{\tilde{V}_k}^{\perp} r(A, v_{k+1})\| < \varepsilon$$

and if $\sigma_i < \varepsilon$ for $i = 1, \ldots, k$, then

$$\sigma_i \leqslant \kappa\sigma_i + \|\mathscr{P}_{\tilde{F}_k}^{\perp} \mathscr{P}_{\tilde{V}_k}^{\perp} r(A, v_{k+1})\| \leqslant (\kappa + 1)\varepsilon = \varepsilon'.$$

Thus, $\operatorname{rank}(\tilde{F}_{k+1}, \varepsilon') \leqslant \operatorname{rank}([\tilde{F}_k \ \mathscr{P}_{\tilde{V}_k}^{\perp} r(A, v_{k+1})], \varepsilon) \leqslant \operatorname{rank}(\tilde{F}_k, \varepsilon)$. The proof is completed by noticing that if $v_{k+1} = \tilde{F}_k h + \varepsilon u$, with $\|u\| = 1$ and $\|h\| = \mathrm{O}(1)$, then Lemma 1.1 can be applied: if we set $\varepsilon' = \|h\|\varepsilon$, then $\operatorname{rank}(\mathscr{P}_{v_{k+1}}^{\perp} \tilde{F}_k, \varepsilon') \leqslant \operatorname{rank}(\tilde{F}_k, \varepsilon) - 1$. □

Lemma 2.3 divides the set of rank conservative solvers into two different types. First, there is the Arnoldi type algorithm for which $v_{k+1} \in \mathscr{R}(\tilde{F}_k)$. This condition corresponds to setting $w_k = AV_k t_k$, for some vector $t_k \in \mathbb{C}^k$.

The second type is an RKS or Davidson type algorithm. These methods compute v_{k+1} such that $r(A, v_{k+1}) \in \mathscr{R}(AV_k) \cup \mathscr{R}(V_k) = \mathscr{R}(\tilde{F}_k) \cup \mathscr{R}(V_k)$. In order to compute v_{k+1} (even implicitly) from $r(A, v_{k+1})$, these methods will need to solve a linear system. Both methods are rank conservative, as we show in the following theorem.

Lemma 2.4. *Let $b_{k+1} = V_k p_k + AV_k q_k$ be an arbitrary vector in $\mathscr{R}(V_k) \cup \mathscr{R}(AV_k)$. Given some α_k, $\beta_k \in \mathbb{C}$, then at step k of an orthogonal projection method, the vector w_k satisfies*

$$(\alpha_k A - \beta_k I) w_k = b_{k+1} \tag{8}$$

iff $\operatorname{rank}(\tilde{F}_{k+1}) \leqslant \operatorname{rank}(\tilde{F}_k)$.

Say that $\alpha_k = 1$ and that the vector w_k is approximated by $\hat{w}_k$ such that $s_{k+1} = b_{k+1} - (A - \beta_k I)\hat{w}_k \neq 0$ and set $\tilde{F}_k = R(A, V_k)$. Then $\|\mathscr{P}_{\tilde{V}_k}^{\perp} \mathscr{P}_{\tilde{V}_{k+1}}^{\perp} s_k\| \leqslant \varepsilon \|\mathscr{P}_{\tilde{V}_k}^{\perp} w_k\|$, implies that $\operatorname{rank}(\tilde{F}_{k+1}, \varepsilon') \leqslant \operatorname{rank}(\tilde{F}_k, \varepsilon)$, with $\varepsilon' = \mathrm{O}(\varepsilon)$. Hence, the method is ε'-rank conservative.

Proof. Since $v_{k+1} = \mathscr{P}_{\tilde{V}_k}^{\perp} w_k / \|\mathscr{P}_{\tilde{V}_k}^{\perp} w_k\|$, we can write that $w_k = \eta v_{k+1} + V_k h_k$, ($\eta \neq 0$). If we plug this in Eq. (8), then we obtain

$$\begin{aligned} V_k p_k + AV_k q_k - (\alpha_k A - \beta_k I)(\eta v_{k+1} + V_k h_k) \\ = \alpha_k \eta A v_{k+1} - \eta \beta_k v_{k+1} + \alpha_k AV_k h_k - \beta_k V_k h_k. \end{aligned}$$

Since $r(A, v_{k+1}) = Av_{k+1} - \gamma v_{k+1}$ with $\gamma = v_{k+1}^* Av_{k+1}$, this becomes

$$\alpha_k \eta r(A, v_{k+1}) = V_k(p_k + \beta_k h_k) + AV_k(q_k - \alpha_k h_k) - \xi v_{k+1},$$

with $\xi = -(\alpha_k\gamma + \beta_k)\eta$. If $\alpha_k \neq 0$, then this means that $r(A, v_{k-1}) \in \mathscr{R}(V_{k-1}) \cup \mathscr{R}(AV_k) = \mathscr{R}(V_{k-1}) \cup \mathscr{R}(R(A, V_k))$ so that $r(A, v_{k-1}) \in \mathscr{R}(V_{k-1}) \cup \mathscr{R}(\tilde{F}_k)$. Using Lemma 2.3, this is equivalent to saying $\operatorname{rank}(\tilde{F}_{k-1}) \leqslant \operatorname{rank}(\tilde{F}_k)$.

If $\alpha_k = 0$, then

$$\xi v_{k-1} = V_k(p_k + \beta_k h_k) + AV_k q_k.$$

Applying $\mathscr{P}_{V_k}^{\perp}$ gives

$$\xi v_{k-1} = \mathscr{P}_{V_k}^{\perp} AV_k q_k = \tilde{F}_k q_k.$$

By Lemma 2.3 this implies again that the rank does not increase.

Let us now prove the second part. Using $r(A, v_{k-1}) = Av_{k-1} - \gamma v_{k-1} = (A - \beta_k I)v_{k-1} + (\beta_k - \gamma)v_{k-1}$, we can derive

$$\begin{aligned} r(A, v_{k-1}) &= \frac{1}{\eta}((A - \beta_k I)w_k - (A - \beta_k I)V_k h_k) + (\beta_k - \gamma)v_{k-1} \\ &= \frac{1}{\eta}(V_k(p_k - \beta_k h_k) + AV_k(q_k - h_k) - s_k) + (\beta_k - \gamma)v_{k-1} \end{aligned}$$

and hence

$$\begin{aligned} \mathscr{P}_{\tilde{V}_{k-1}}^{\perp} r(A, v_{k-1}) = \mathscr{P}_{\tilde{V}_k}^{\perp} r(A, v_{k-1}) &= \frac{1}{\eta}(R(A, V_k)(q_k - h_k) - \mathscr{P}_{\tilde{V}_k}^{\perp} s_k) \\ &= \frac{1}{\eta}(\tilde{F}_k(q_k - h_k) - \mathscr{P}_{\tilde{V}_k}^{\perp} s_k) \| \cdot \mathscr{P}_{\tilde{V}_k}^{\perp} \mathscr{P}_{\tilde{V}_k}^{\perp} r(A, v_{k-1}) \| \\ &= \frac{1}{|\eta|} \| \cdot \mathscr{P}_{\tilde{V}_k}^{\perp} \mathscr{P}_{\tilde{V}_{k-1}}^{\perp} s_k \| \leqslant \varepsilon. \end{aligned}$$

reminding that $|\eta| = \| \cdot \mathscr{P}_{\tilde{V}_k}^{\perp} w_k \|$. Using Lemma 2.3, this proves the theorem. □

Basically, Lemma 2.4 says that in exact arithmetic there are only three rank conservative eigenvalue solvers: Arnoldi, RKS and a Davidson algorithm that used an direct linear system solver. Inversely, any eigenvalue solver that has a residual of rank 1 can be interpreted as a generalisation of an Arnoldi or an RKS process.

Example 2.1. We illustrate the difference between rank-conservative eigenvalue solvers and ε-rank-conservative solvers with a small example of the RKS method.

We constructed a 100×100 bidiagonal matrix A, setting $(A)_{i,i} = -i$ and $(A)_{i,i+1} = 1$. We compute the rightmost eigenvalue $\lambda = -1$ using Algorithm 1.1. The starting vector is $v_1 = [0.1, 0.1, \ldots, 0.1]^*$ and v_{k+1} is computed from $w_k \leftarrow (A - \mu_k I)v_k$, where $\mu_1 = 1$ and $\mu_i = 0_i$ for $i > 1$. The approximation ϑ_i is computed as the rightmost eigenvalue of G_k. The linear systems are solved with Gaussian elimination. It is well known that the systems will only be solved with

a (relative) error proportional to the condition number of the matrix, i.e. relative to $\|A-\mu_k I\|\|(A-\mu_k I)^{-1}\|$. The error will be large when the method converges. This effect is illustrated in Table 1. The table shows for iteration step 4–8 the error on the eigenvalue, the residual norm and the absolute error on the solved system. It also displays the three largest singular values of F_k. In theory, this RKS method is rank conservative. In this example, the second singular value of the residual can not be neglected at the point of convergence, in this case the sixth step.

The second part of the table shows what happens when we fix the shift $\mu_k = \theta_5$ for $k \geq 5$. The quadratic convergence is lost, but the convergence rate is very high. The second singular value of F_k remains of order 1e−12. If we set e.g. $\varepsilon = 1\text{e}{-}11$, then the ε-rank of F_k is one.

3. Restarting and filtering

In practice, we cannot assume that Algorithm 1.1 will converge in a predictable amount of steps. Furthermore, if several eigenvalues must be found, then the number of iteration steps that the algorithm needs in order to find them all, will likely exceed an acceptable amount. A restarting procedure for relation (1) must be considered. A different reason for restarting the Arnoldi relation (for the generalised eigenvalue problem) was formulated in [2]: if the problem has an infinite eigenvalue, then spurious approximations of this infinite eigenvalue can pop up and bring about wrong results. The filtering property of the Implicitly restarted Arnoldi [2] algorithm can be used to filter away these spurious eigenvalues. In [16], it is shown how the restarting of the Arnoldi equation can be generalised to the RKS equation. A different restarting procedure, based on the Schur decomposition of G_k for the Jacobi–

Table 1
Using RKS with shift $\sigma_k = \theta_k$ gives quadratic convergence, but a residual with large rank. The second part shows RKS with a fixed shift ($\sigma_k = \theta_5$ for $k > 5$). The convergence is slower but the residual rank remains one

k	$\lvert\theta_k - \lambda\rvert$	$\|Ay_k - \theta_k y\|$	$\|(A-\mu_k I)w_k - r_k\|$	σ_1	σ_2	σ_3
4	3.8e−1	7.7e−1	1.5e−16	2.7e+1	5.5e−14	1.0e−14
5	1.2e−3	7.5e−2	5.9e−14	2.6e+1	8.3e−14	1.4e−14
6	2.3e−7	1.1e−5	3.1e−11	2.5e+1	3.0e−12	1.1e−14
7	2.2e−15	2.2e−13	2.2e−3	2.5e+1	1.8e−9	9.4e−13
8	1.2e−16	1.0e−14	3.0e−2	2.5e+1	7.1e−2	5.2e−10
6	2.3e+1	1.0e−5	4.3e−15	2.5e+1	3.0e−12	1.0e−14
7	9.4e−12	1.2e−9	1.3e−13	2.5e+1	5.6e−12	5.7e−14
8	2.1e−15	1.1e−13	4.6e−15	2.5e+1	8.0e−12	1.2e−13

Davidson algorithm is proposed in [17]. In this section, we will consider these two related procedures for an arbitrary OPM.

When we use the word *restarting*, we mean the reduction of Eq. (1) to an equation

$$AV_{k-p}^{+}L_{k-p}^{+} = V_{k-p}^{+}K_{k-p}^{+} + (\tau_{k-p}^{+} - \sigma_{k-p}^{+}I)F_{k-p}^{+}, \tag{9}$$

with V_{k-p}^{+}, $F_{k-p}^{+} \in \mathbb{C}^{n\times k-p}$ and K_{k-p}^{+}, $L_{k-p}^{+} \in \mathbb{C}^{k-p\times k-p}$. If these are matrices that could have been generated by the same OPM, using a new starting vector v_1^{+}, then we call this operation an *implicit restart*. Otherwise, the equation is restarted *explicitly*. Often, an implicitly restart procedure can be seen as a filtering procedure. After the filtering, the new basis V_{k-p}^{+} contains a filtered version of the old basis V_{k-p},

$$\mathscr{R}(V_{k-p}^{+}) = \mathscr{R}(\phi_p(A)V_{k-p}),$$

where $\phi_p(A)$ is a polynomial or a rational filter function that depends on the restart algorithm.

3.1. Reducing the projection equation

Consider the reduction of Eq. (1) to an Eq. (9) of lower dimension. It is clear that a transformation of V_k into V_{k-p}^{+} must be orthogonal, i.e. there exists an orthogonal matrix Q such that $V_{k-p}^{+} = V_kQ$. A corresponding transformation will then be applied on G_k. The following lemma shows this for the general case.

Lemma 3.1. *Given a set of matrices V_k, K_k, L_k and F_k that fulfil $AV_kL_k = V_kK_k + (\tau_kA - \sigma_kI)F_k$. Say* $\operatorname{rank}(F_k) = l$ *and* $F_k = gr^*$, $g \in \mathbb{C}^{n\times l}$, $r \in \mathbb{C}^{k\times l}$. *If $[Q\ q]$ is a unitary matrix, with $Q \in \mathbb{C}^{k\times k-p}$ and $q \in \mathbb{C}^{k\times p}$, and if $Z \in \mathbb{C}^{k\times k-p}$ is a full rank matrix such that*

(a) $q^*K_kZ = 0$ *or* $q^*K_k = g_Kr^*$, $g_K \in \mathbb{C}^{1\times l}$,

(b) $q^*L_kZ = 0$ *or* $q^*L_k = g_Lr^*$, $g_L \in \mathbb{C}^{1\times l}$

then with $V_{k-p}^{+} \equiv V_kQ$, $K_{k-p}^{+} \equiv Q^*K_kZ$, $L_{k-p}^{+} \equiv Q^*L_kZ$ and $F_{k-p}^{+} = F_kZ + wr^*Z$, for some $w \in \mathbb{C}^{n\times l}$, we get

$$AV_{k-p}^{+}L_{k-p}^{+} = V_{k-p}^{+}K_{k-p}^{+} + (\tau_kA - \sigma_kI)F_{k-p}^{+}.$$

Moreover, $\operatorname{rank}(F_{k-p}^{+}) \leqslant l$.

Proof. Since $QQ^* + qq^* = I$, multiplication of $AV_kL_k = V_kK_k + (\tau_kA - \sigma_kI)F_k$ with Z gives

$$AV_kQQ^*L_kZ + \underbrace{AV_kqq^*L_kZ}_{0=AV_kg_Lr^*Z} = V_kQQ^*K_kZ + \underbrace{V_kqq^*K_kZ}_{0=V_kg_Kr^*Z} + (T_kA - \sigma_kI)\underbrace{F_kZ}_{gr^*Z}.$$

Thus, $F_k^+ = F_kZ + wr^*Z = (g + w)r^*Z$ and thus $\operatorname{rank}(F_k^+{}_p) \leqslant l$. □

If we reduce the size of the subspace V_k by the rank of the residual, i.e. $l = p$, then the distinction between (a) and (b) in Lemma 3.1 may become obsolete. Indeed, if $Z \in \mathbb{C}^{k\times k-l}$, then the null space of Z has dimension l, which is equal to $\operatorname{rank}(r)$. So there will always exist a Z such that $r^*Z = 0$. In that case, supposing that $q^*K_k = g_Kr^*$ and $q^*L_k = g_Lr^*$ implies that there exists a Z such that $q^*K_kZ = 0 = q^*L_kZ$. Notice also that if $w \neq 0$, then the property that $V_k \perp F_k$ is not inherited automatically by $V_k^+{}_p$ and $F_k^+{}_p$.

The two ways to restart an OPM are clear from the proof of Lemma 3.1. The first solution is constructed such that the additional residual term $(AV_kqq^*L_k - V_kqq^*K_k)Z$ is equal to zero. The second solution makes this residual of the same form as the original residual. Both solutions result in a new residual $F_k^+{}_p$ that has the same rank as the residual before the restart.

From a different point of view, there are also two different possibilities for reducing the size of the projection equation. Reducing the size of the subspace V_k means the removal of a subset of its vectors. We call this filtering of the subspace. These vectors can be selected in an *explicit way*, e.g. as the Ritz vectors that approximate 'uninteresting' eigenvectors. The filtering can also be worked out in an implicit manner.

3.2. *Reduction with Schur vectors*

In the explicit approach, we can make use of the generalised Schur decomposition of the matrices K_k and L_k. This corresponds to the restarting approach in [17]. This decomposition computes a set of orthogonal matrices $Q, Z \in \mathbb{C}^{k\times k}$, such that $Q^*K_kZ = T_K$ and $Q^*L_kZ = T_L$, with upper triangular matrices T_K, $T_L \in \mathbb{C}^{k\times k}$. If we multiply the projection equation on the right by Z, then

$$AV_kQQ^*L_kZ = V_kQQ^*K_kZ + (\tau_kA - \sigma_kI)F_kZ,$$
$$AV_kQT_L = V_kQT_K + (\tau_kA - \sigma_kI)F_kZ.$$

If $F_kZ = 0$, then this would be a partial Schur decomposition of A. We set $V_kQ = V_k^+$ and denote by $V_k^+{}_p$ the same matrix, but restricted to its first $k - p$ columns. Because of the upper triangular form of T_K and T_L, the relation can be reduced to Eq. (9) by simply deleting the last p columns. If Q can be computed efficiently such that the wanted information is collected in the first $k - p$ columns of V_k^+, then the restart may be expected to give good results. We expect also that $\operatorname{rank}(F_k) = \operatorname{rank}(F_k^+{}_p)$.

The matrices involved in this restarting procedure can also be found without using a Schur decomposition. This is shown in the following lemma, that says how a set of Ritz values can be removed explicitly from the approximation. Therefore, we call these *explicit filtering* procedures, in contrast with the implicit filtering in the next section.

Lemma 3.2. *Given matrices V_k, K_k, L_k, F_k as in Lemma 3.1 with $V_k^* F_k = 0$. Let $(\theta_i, y_i = V_k L_k z_i)$, $i = 1, \ldots, k$ be the (Harmonic) Ritz pairs of the problem, i.e. $K_k z_i = \theta_i L_k z_i$. Define the matrices Q and $Z \in \mathbb{C}^{k \times k-p}$ such that Q is orthogonal and*

$$\mathscr{R}(Z) = \mathscr{R}(z_1, \ldots, z_{k-p}) \quad \text{and} \quad \mathscr{R}(Q) = \mathscr{R}(L_k Z).$$

Then $V_{k-p}^+ = V_k Q$, $K_{k-p}^+ = Q^* K_k Z$, $L_{k-p}^+ = Q^* L_k Z$ and $F_{k-p}^+ = F_k Z$ define a projection equation that generates the same (Harmonic) Ritz pairs, except for $(\theta_{k-p+1}, y_{k-p+1}), \ldots, (\theta_k, y_k)$.

Proof. We only have to prove that if $q^* Q = 0$, then $q^* K_k Z = 0 = q^* L_k Z$, so we can apply Lemma 3.1. Since there exist a non-singular matrix P for which $QP = L_k Z$, $q^* L_k Z$ must be zero. Also, for all $i < k - p$, $q^* K_k z_i = \theta_i q^* L_k z_i = 0$, which proves $q^* K_k Z = 0$. Since $V_{k-p}^+ = F_k Z$, it follows directly from $V_k^* F_k = 0$ that $(V_{k-p}^+)^* F_{k-p}^+ = Q^* V_k^* F_k Z = 0$. □

This lemma illustrates the difference between implicitly restarted Arnoldi [1] and implicitly restarted RKS [16]. For Arnoldi, $L_k = I$ and thus one may choose $Q = Z$, whereas for RKS, $L_k \neq I$.

3.3. Implicit filtering

IRA gives us another way to look at a restarting algorithm. Given a shift μ, a step of IRA combines the basis vectors V_k of a Krylov subspace $\mathscr{K}_k(A, v_1)$ to a new orthogonal basis for $\mathscr{K}_{k-1}(A, (A - \mu I)v_1)$. Thus, it computes the results V_{k-1}^+, G_{k-1}^+ of an Arnoldi iteration with a new starting vector $v_1^+ = (A - \mu I)v_1$. The vector that is removed from the subspace V_k is not selected explicitly, but it is an implicit result of the choice of μ. One can prove that if the shift is chosen equal to a Ritz value $\mu = \theta_i$, then the corresponding Ritz vector is removed from V_k, so *implicitly* Lemma 3.2 is executed.

Let us recall briefly the mechanism behind IRA. Suppose that we ran the Arnoldi algorithm to obtain $AV_k = V_k G_k + F_k$, with G_k upper Hessenberg and $F_k = f_k e_k^*$ the rank one residual. Given a shift μ, we compute the QR decomposition $QR = G_k - \mu I$, with Q unitary Hessenberg and R upper triangular. Hence, we can write

$$(A-\mu I)V_k = V_k(G_k-\mu I) + F_k = V_kQR + F_k = V_k[Q_1\ q_2]\begin{bmatrix} R_1 & r_2 \\ 0 & r_3 \end{bmatrix} + f_ke_k^*.$$

Deleting the last column of both sides of this equation makes F_k vanish, so

$$(A-\mu I)V_{k-1} = V_kQ_1R_1.$$

If we define $V_{k-1}^+ \equiv V_kQ_1$, then this means that $V_{k-1}^+ = (A-\mu I)V_{k-1}R_1^{-1}$ and thus

$$\mathscr{R}(V_{k-1}^+) = \mathscr{R}((A-\mu I)V_{k-1}). \tag{10}$$

supposing that R_1 has full rank (one can see that this is true if μ is no eigenvalue of A). Eq. (10) proves the *implicit filtering property* of the IRA algorithm, since V_{k-1}^+ equals the old subspace V_{k-1} filtered with a polynomial of degree 1 in A. The zero of the polynomial is μ. Repeating the restarting algorithm builds a polynomial of higher degree.

On the other hand, if we multiply the equation from the right by Q_1, then we get

$$(A-\mu I)V_kQ_1 = V_kQRQ_1 + F_kQ_1.$$
$$AV_{k-1}^+ = V_{k-1}^+([R_1\ r_2]Q_1 + \mu Q_1) + (V_kq_2r_3e_k^* + f_ke_k^*)Q_1 = V_{k-1}^+G_{k-1}^+ + F_{k-1}^+.$$

The matrix Q_1 fulfils the conditions of Lemma 3.1. It is easy to see that $G_{k-1}^+ = Q_1^*G_kQ_1$ is upper Hessenberg and

$$F_{k-1}^+ = [0\ \cdots\ 0\ f_{k-1}^+] \quad \Rightarrow \quad \operatorname{rank}(F_{k-1}^+) \leqslant 1.$$

Lemma 2.4 showed that we cannot filter the subspace using a restarting procedure that decrements the size of the subspace V_{k+1} by one if the rank of the residual is larger than one. However, we can implicitly apply a filter, but then we will need to delete more than one vector.

Proposition 3.3. *A restarting algorithm that reduces the size of the subspace V_{k+1} with one, can only be a polynomial (or rational) filter of degree one, if the rank of the residual matrix equals one.*

Proof. Due to the correspondence with an Arnoldi or RKS process (Lemma 2.4), a relation with $\operatorname{rank}(F_k) = 1$ can be filtered while restarting. This is shown in [16,1]. Inversely, for a rational filter, given a full rank matrix $P \in \mathbb{C}^{k\times k}$ (set $\bar{P} = [P^*\ 0]^*$), given any vector $t_{k+1} \in \mathbb{C}^{k+1}$ and a pair $\alpha,\ \beta \in \mathbb{C}$. If V_k^+ spans a filtered version of V_k, then

$$V_k^+ = V_{k+1}Q = (\alpha A - \beta I)^{-1}(A-\mu I)V_kP,$$

so

$$AV_{k+1}(\alpha Q + \bar{P}) = V_{k+1}(\beta Q - \mu\bar{P}),$$

and

$$AV_{k+1}[\alpha Q + \bar{P}t_{k+1}] = V_{k+1}[(\beta Q - \mu\bar{P})(V_{k+1}^* A V_{k+1} t_{k+1})] + [0 \cdots \mathscr{P}_{k+1}^{\perp} A V k + 1 t_{k+1}]$$

Thus, $\operatorname{rank}(F_k) = \operatorname{rank}(\mathscr{P}_{V_{k+1}}^{\perp} A v_{k+1}) = 1$. Setting $\alpha = 0$ and $\beta = 1$, gives the result for a polynomial filter. □

The following theorem shows how in general an eigenvalue solver can be restarted in combination with an implicit filtering step.

Theorem 3.4. *Given V_k, K_k, L_k and F_k that fulfil the projection Eq.* (1). *Suppose that $F_k = [0\ F_p]$, $F_p \in \mathbb{C}^{n\times p}$ and* $\operatorname{rank}(F_k) = p$ *and suppose that $\alpha, \beta \in \mathbb{C}$ are given such that β/α is not an eigenvalue of A and $\beta\tau_k \neq \sigma_k\alpha$. If $V_k^* F_k = 0$, then with the QR factorisation $\alpha K_k - \beta L_k = [Q_1\ Q_2]R$ and $Z \equiv (\tau_k K_k - \sigma_k L_k)^{-1} Q_1$, we define $L_{k-p}^+ \equiv Q_1^* L_k Z$, $K_{k-p}^+ \equiv Q_1^* K_k Z$ and $V_{k-p}^+ \equiv V_k Q_1$. The relation*

$$AV_{k-p}^+ L_{k-p}^+ = V_{k-p}^+ K_{k-p}^+ + (\tau_k A - \sigma_k I)F_{k-p}^+$$

is a restarted projection equation, so $\sigma_{k-p}^+ = \sigma_k$ and $\tau_{k-p}^+ = \tau_k$. Moreover, $(V_{k-p}^+)^* F_{k-p}^+ = 0$ and

$$\mathscr{R}\left(V_{k-p}^+\right) = \mathscr{R}\left((\tau_k A - \sigma_k I)^{-1}(\alpha A - \beta I)V_{k-p}^+\right).$$

Proof. It can easily be derived that by shifting the projection equation,

$$(\tau_k A - \sigma_k I)V_k(\alpha K_k - \beta L_k) = (\alpha A - \beta I)V_k(\tau_k K_k - \sigma_k L_k) - (\beta\tau_k - \alpha\sigma_k)(\tau_k A - \sigma_k I)F_k.$$

If we apply Lemma 3.1 on this equation, then r is defined by $F_k = F_p[0\ I_p] = F_p r^*$. On the one hand, $Q_2^*(\alpha K_k - \beta L_k) = Q_2^*[Q_1\ Q_2]R = \tilde{R}[0\ I_p] = \tilde{R}r^*$, where $\tilde{R}$ is the $p \times p$ lower triangular submatrix of R. On the other hand, $Q_2^*(\tau_k K_k - \sigma_k L_k)Z = Q_2^*(\tau_k K_k - \sigma_k L_k)(\tau_k K_k - \sigma_k L_k)^{-1} Q_1 = 0$, so Lemma 3.1 may be applied. Hence,

$$(\tau_k A - \sigma_k I)V_{k-p}^+(\alpha K_{k-p}^+ - \beta L_{k-p}^+) = (\alpha A - \beta I)V_{k-p}^+(\tau_k K_{k-p}^+ - \sigma_k L_{k-p}^+) - ((\beta\tau_k + \alpha\sigma_k)F_p + (\tau_k A - \sigma_k I)V_k Q_2 \tilde{R})[0\ I_p]Z$$

and shifting back the new projection equation proves the first result. We find that $F_{k-p}^+ = F_k Z + (\beta\tau_k + \alpha\sigma_k)^{-1}(\tau_k A - \sigma_k I)V_k Q_2 \tilde{R}[0\ I_p]Z$. So if $V_k^* F_k = 0$, then $(V_{k-p}^+)^* F_{k-p}^+ = 0$ because $Q_1^* Q_2 = 0$. □

This theorem can also be applied in the special case where $L_k = I$, $\tau_k = 0$ and $\sigma_k = -1$ and this gives:

Theorem 3.5 *Given V_k, G_k, F_k, such that $AV_k = V_kG_k + F_k$. Suppose that $F_k = [0\ F_p]$, $F_p \in \mathbb{C}^{n\times p}$ and $\operatorname{rank}(F_k) = p$. If, given the parameters $\alpha, \beta \in \mathbb{C}$ such that β/α is no eigenvalue of A, the matrix Q_1 contains the first k columns of the QR decomposition of $\alpha G_k - \beta I = [Q_1\ Q_2]R$, then $G^+_{k-p} \equiv Q_1^*G_kQ_1$ and $V^+_{k-p} \equiv V_kQ_1$ define a restarted relation*

$$AV^+_{k-p} = V^+_{k-p}G^+_{k-p} + F^+_{k-p},$$

with $\operatorname{rank}(F^+_{k-p}) \leqslant p$. Moreover, if $V_k^*F_k = 0$, then

$$(V^+_{k-p})^*F^+_{k-p} = 0 \text{ and } \mathscr{R}\left(V^+_{k-p}\right) = \mathscr{R}\left((\alpha A - \beta I)V_{k-p}\right).$$

Proof. This theorem is a special case of Theorem 3.4 with $L_k = I$, $G_k = K_k$, $\sigma_k = -1$ and $\tau_k = 0$.

Notice that with

$$\alpha G_k - \beta I = [Q_1\ Q_2]\begin{bmatrix} R_1 & R_2 \\ 0 & R_4 \end{bmatrix},$$

we derive that $\alpha G^+_{k-p} = Q_1^*(\alpha G_k - \beta I + \beta I)Q_1 = [R_1\ R_2]Q_1 + \beta I$. □

The implicit restart cannot be repeated immediately, because the first $k - 2p$ columns of F^+_{k-p} are not equal to zero. However, the restart can be repeated on the rotated system

$$AV^+_{k-p}G = V^+_{k-p}G(G^*G^+_{k-p}G) + F^+_{k-p}G$$

or

$$AV^+_{k-p}G(G^*L^+_{k-p}G) = V^+_{k-p}G(G^*K^+_{k-p}G) + (A - \sigma^+_{k-p}I)F^+_{k-p}G.$$

with $G^*G = I = GG^*$ and $F^+_{k-p}G = [0\ \cdots\ 0\ \bigstar]$. Notice that G can be computed from Z or Q_1 (G must be constructed such that the $p \times (k-p)$ lower left part of ZG or Q_1G is zero).

The results for the inexact filtering procedure are summarised in the Algorithms 3.1 and 3.2. We do not need to know the matrix T_p to compute the restarted equation.

Algorithm 3.1. *Inexact polynomial filtering*

In: $V_k, F_k = [0 \ldots 0, F_p]$ *with* $V_k^* = 0$, *and* G_k, $\alpha, \beta \in \mathbb{C}$
Out: G_{k-p}^+, V_{k-p}^+ *such that* $\mathscr{R}\left(V_{k-p}^+\right) = \mathscr{R}((\alpha A - \beta I)V_{k-p}^+)$
1. *Set* $[\bar{G}_{k-p}\ \bar{G}_p] = \alpha G_k - \beta I$
2. *Compute* Q_1, R_1 *from* $\bar{G}_{k-p} = Q_1 R_1$
3. *Set* $R_2 \leftarrow Q_1^* \bar{G}_p$
4. *Compute* G *from* Q_1
5. *Set* $V_{k-p}^+ \leftarrow V_k Q_1 G$
6. *Set* $G_{k-p}^+ \leftarrow \frac{1}{\alpha} G^* [R_1 R_2] Q_1 G + \frac{\beta}{\alpha} I$

Algorithm 3.2. *Inexact rational filtering*

In: $V_k, F_k = [0 \ldots 0, F_p]$ *with* $V_k^* F_k = 0$, *and* $K - k, L_k, \alpha, \beta \in \mathbb{C}$
Out: $K_{k-p}^+, L_{k-p}^+, V_{k-p}^+$ *such that* $\mathscr{R}\left(V_{k-p}^+\right) = \mathscr{R}((A - \sigma_k I)^{-1}(\alpha A - \beta I)V_{k-p}^+)$
1. *Set* $[\bar{G}_{k-p}\ \bar{G}_p] = \alpha K_k - \beta L_k$
2. *Compute* Q_1, R_1 *from* $\bar{G}_{k-p} = Q_1 R_1$
3. *Set* $Z \leftarrow (K_k - \sigma_k L_k)^{-1} Q_1$
4. *Compute* G *from* Z
5. *Set* $V_{k-p}^+ \leftarrow V_k Q_1 G$
6. *Set* $K_{k-p}^+ \leftarrow G^* Q_1^* K_k Z G$
7. *Set* $L_{k-p}^+ \leftarrow G^* Q_1^* L_k Z G$

Example 3.1. We illustrate the use of an implicit filter for a ε-rank conservative solver with an example. The example involves a generalised eigenvalue problem with a singular B. We bring this problem back to a *standard* eigenvalue problem by studying the eigenvalues of $(A - \sigma B)^{-1}B$, which has the same eigenvectors as the original problem. The matrices come from the simulation of flow of a viscous fluid with free surface on a tilted plane, using a finite element approach. The size of the eigenvalue problem $Ax = \lambda Bx$ is $n = 536$ and the rightmost eigenvalue is computed. The rightmost eigenvalue is equal to $\lambda = -9.4883$.

Since the matrix B is singular, the eigenvalue problem has an infinite eigenvalue. Approximations of the infinite eigenvalue will occur in the solution as large, finite eigenvalues. If such a *spurious* eigenvalue becomes the rightmost one, it will mislead the algorithm and the exact solution will not be found. Therefore, we will filter them out with the inexact filtering procedure of Algorithm 3.1.

We iterated the OPM algorithm $k = 10$ times with a starting vector $v_1 = [1 \ \cdots \ 1]^*/\sqrt{n}$ and $w_k = (A - \sigma B)^{-1} B v_k$, with a shift $\sigma = -1$. Setting the matrices $K_k = (V_k^* B V_k)^{-1}$ and $L_k = (V_k^* A V_k)^{-1}$, we computed the approximate eigenvalues as the eigenvalues of the small system

$$L_k^{-1} z = \theta K_k^{-1} z.$$

We do not know the matrices G_k or $\tilde{F}_k$ in the relation

$$(A - \sigma B)^{-1} B V_k = V_k G_k + \tilde{F}_k.$$

but we can compute the residual from $A V_k L_k = B V_k K_k + F_k$, which corresponds to

$$(A - \sigma B)^{-1} B V_k = V_k L_k (K_k - \sigma_k L_k)^{-1} + (A - \sigma B)^{-1} F_k (K_k - \sigma_k L_k)^{-1}.$$

Setting $G_k = L_k(K_k - \sigma_k L_k)^{-1}$ and using $F_k(K_k - \sigma_k L_k)^{-1}$, will give the input matrices for Algorithm 3.1.

The linear system was solved using GMRES to a tolerance of 10×10^{-6}, for the first eight steps and to a tolerance of 10×10^{-9} for the last two steps (since inexact iterative methods converge asymptotically well, it is better to use a higher tolerance at the end of the process [18]). The results are shown in Table 2.

Implicitly restarting the relation seems not to conflict with the convergence, i.e. it does not reduce the residual norm. Only when the parameter p – the assumed rank of G_k – is taken too large, some convergence is lost. The quality of the implicit filter grows with p, because the neglected part of the residual then becomes smaller. Since there are no eigenvalues with a positive real part, Ritz values with a positive real part can be considered as spurious eigenvalues. With $p \geqslant 2$, all spurious eigenvalues with a positive real part are filtered away. The filter is then applied with very good accuracy. But if we set $p = 3$, then some convergence is lost ($\|r(A, y_k)\| = 1 \times 10^{-4}$). Comparing the third row to the second column, one can see that the accuracy of the filter is proportional to σ_{p+1}.

4. Conclusion

We showed in this text that an implicit filtering procedure can only be applied on rank conservative methods or methods that have numerically a low residual rank. The larger the rank of the residual matrix is, the more inaccurate the implicit filtering will be or the more basis vectors must be sacrificed. We

Table 2
Application of the implicit restarting algorithm on an inexact OPM. The first column shows the residual norm of the Ritz-vector, the number of eigenvalues with a positive real part, the error on the implicit filter ($W_{k-p} = (A - \sigma B)^{-1} B V_{k-p}$) and the largest singular values of $F_k(K_k - \sigma L_k)^{-1}$. The following columns show the same quantities after a restart with an assumed residual rank of $p = 1, 2, 3$

	$k = 10$	$p = 1$	$p = 2$	$p = 3$
$\|r(A, y_k)\|$	1.1×10^{-7}	1.1×10^{-7}	$5.9 \times 10^{7-}$	1.0×10^{-4}
# positive	1	1	0	0
$\|\mathscr{P}^{\perp}_{V_{k-p}} W_{k-p}\|$		5.9×10^{-7}	1.9×10^{-8}	3.4×10^{-8}
σ_1	1.1×10^{-3}	5.1×10^{-3}	6.7×10^{-4}	8.7×10^{-4}
σ_2	6.3×10^{-7}	3.5×10^{-6}	2.0×10^{-4}	2.6×10^{-4}
σ_3	$3.1 \times 10^{-c*}$	6.7×10^{-10}	2.5×10^{-9}	5.6×10^{-5}
σ_4	5.0×10^{-10}	2.1×10^{-10}	2.1×10^{-10}	2.3×10^{10}

showed that the set of rank conservative eigenvalue solvers is restricted to those methods that use an Arnoldi type of scheme (with matrix-vector products) or that perform Rational Krylov like computations (solving a linear system with the matrix A accurately). Other methods and inexact variants of rank conservative methods cannot be filtered implicitly in combination with an implicit restart.

Alternatively, unwanted Ritz or Schur vectors can also be filtered away explicitly by using a Schur decomposition of the projection of A on the column range of V_k. The disadvantage of this explicit approach is that one must identify the unwanted vectors and that only Ritz vectors or harmonic Ritz vectors may be removed. Using an implicitly filtering function, one can also remove those vectors that correspond to spurious eigenvalues, even if these are not Ritz vectors. So the message is relatively pessimistic: true implicit filtering can only be applied in a very limited set of cases. If a better filter must be applied, then the cost – in terms of 'lost' subspace vectors – will be larger.

Acknowledgements

This text presents research results of the Belgian Incentive Programme 'Information Technology' – Computer Science of the Future (IT/IF/5), and the Belgian programme on Interuniversity Poles of Attraction (IUAP 17), initiated by the Belgian State – Prime Minister's Service – Federal Office for Scientific, Technical and Cultural Affairs. The authors are very grateful to the referees, who provided us with many useful comments.

References

[1] D.C. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, SIAM J. Matrix Anal. Applic. 13 (1992) 357–385.
[2] K. Meerbergen, A. Spence, Implicitly restarted Arnoldi and purification for the shift-invert transformation, Math. Comp. 66 (1997) 667–689.
[3] C. Paige, B.N. Parlett, H.A. Van der Vorst, Approximate solutions and eigenvalue bounds from Krylov subspaces, Numer. Linear Algebra Appl. 2 (1995) 115–133.
[4] J.W. Daniel, W.B. Gragg, L. Kaufman, G.W. Stewart, Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization, Math. Comp. 30 (1976) 772–795.
[5] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9 (1951) 17–29.
[6] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bur. Stand. 45 (1950) 255–281.
[7] A. Ruhe, Rational Krylov sequence methods for eigenvalue computation, Linear Algebra Appl. 58 (1984) 391–405.
[8] A. Ruhe, Rational Krylov algorithms for nonsymmetric eigenvalue problems II: Matrix pairs, Linear Alg. Appl. 197/198 (1994) 283–296.

[9] A. Ruhe, The rational Krylov algorithm for nonsymmetric eigenvalue problems III: Complex shifts for real matrices, BIT 34 (1994) 165–176.
[10] M. Crouzeix, B. Philippe, M. Sadkane, The Davidson method , SIAM J. Sci. Comput. 15 (1994) 62–76.
[11] E.R. Davidson, The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices, J. Comput. Phys. 17 (1975) 87–94.
[12] G.L.G. Sleijpen, H.A. van der Vorst, A Jacobi Davidson iteration method for linear eigenvalue problems, SIAM J. Matrix Anal. Appl. 17 (1996) 401–425.
[13] K. Meerbergen, Robust methods for the calculation of rightmost eigenvalues of nonsymmetric eigenvalue problems, Ph.D. Thesis, Department of Computer Science, K.U. Leuven, March 1996.
[14] G. Golub, C. Van Loan, Matrix Computations, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989, 1996.
[15] J.H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.
[16] G. De Samblanx, K. Meerbergen, A. Bultheel, The implicit application of a rational filter in the RKS method, BIT 37 (4) (1997) 924–945.
[17] D.R. Fokkema, G L.G. Sleijpen, H.A. Van der Vorst, Jacobi Davidson style QR and QZ algorithms for the partial reduction of matrix pencils, SIAM J. Sci. Comput. (to appear).
[18] G. De Samblanx, A. Bultheel, On the convergence and the restarting of inexact eigenvalue solvers, TW-report, Department of Computer Science, K.U. Leuven, 1996.