

Filtering en restarting Orthogonal Projection Methods

Gorik De Samblanx Adhemar Bultheel

Report TW 248, December 1996

Department of Computer Science, K.U.Leuven

Abstract

We consider the class of the Orthogonal Projection Methods (OPM) to solve iteratively large and generalised 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. We show that many iterative eigenvalue solvers, such as the Arnoldi and Davidson method, can be seen as an OPM. We show in the second part of the text how an OPM can be restarted – implicitly and explicitly. This restart can be combined with an implicit filtering step, even if inaccurate arithmetic is assumed.

AMS(MOS) Classification : 65F15

Keywords : Davidson, implicitly restarted Arnoldi, generalised eigenvalue problem, shift-invert

1 Introduction

Consider an implementation of a solution to finding a limited set of solutions to the generalised eigenvalue problem

$$Ax = \lambda Bx, \quad A, B \in \mathbf{C}^{n \times n}.$$

If the dimension of this problem is very large and the matrices are sparse, then it can not be tackled using a direct solver, QR or QZ, due to time and memory limitations. Fortunately, in many cases such as in stability analysis, people do not want to know all those solutions of their problem, but they need only a specific part of the spectrum. Therefore, a family of iterative methods has been derived that has the purpose to solve efficiently this more specific problem.

The solution that is probably best known, is the *inverse iteration* method. Inverse iteration is popular because it is well known, simple and robust, it has good convergence properties and it uses only a limited, predefined amount of memory. If the other iterative methods want to attain a comparable level of fame, then they must adopt some of these properties.

For Arnoldi's method, this goal was achieved by the Implicitly Restarted Arnoldi method [20]. The IRA method restarts the Arnoldi algorithm 'automatically' when the memory reaches its bounds before it converges and, by implementing a polynomial filter, IRA corresponds more to inverse iteration, which makes it more robust. For the other methods, a similar effort has been done by different authors, e.g. [6, 7, 9, 18].

In this text, we review some properties of iterative eigenvalue solvers that are called *orthogonal projection methods (OPM)*. These methods project the large eigenvalue problem on a small subspace, where direct methods are workable. In the first part of the text, we look at these methods, we show how they are connected and how they differ. We distinguish two general types of orthogonal projection methods. The distinction can be illustrated as follows :

Suppose that $y = x + \delta x$ and $\theta = \lambda + \delta \lambda$ are good approximations for the eigenvalue and eigenvector ('good' means that δx and $\delta \lambda$ are small enough to neglect the product $\delta \lambda \cdot \delta x$ in the following equations) that are computed from a k -dimensional subspace V_k . We call the residual vector

$$r(y) = Ay - \theta y = (A - \lambda I)\delta x - \delta \lambda x - \delta \lambda \delta x.$$

Looking for the optimal update for y , the first type of methods assumes that $\delta x = 0$. Thus, $(A - \lambda I)\delta x - \delta \lambda \delta x \simeq 0$ and $r(y) \simeq \delta \lambda x$. The vector that is added to the subspace V_k will be equal to $r(y)$. This is the Arnoldi approach. The second type of methods assumes that $\delta \lambda = 0$ and it chooses the update for y to be an approximation of δx , computed from

$$(A - \lambda I)\delta x \simeq r(y),$$

(since λ is not known, another parameter is used). Some methods can perform these operations assuming exact arithmetic, other ones allow computational errors – we will refer to this by *inexact* or inaccurate arithmetic.

In the second part of the text, we show how these methods can be restarted. We also show how this restart can be combined with an implicit filter. Actually, we show why it is so hard to combine both the ideas of filtering and restarting for a solver that uses inexact arithmetic.

The paper is structured as follows. In §1, we review the properties of projection matrices and their effect on the numerical rank of a matrix. §2 presents a general algorithm of an OPM. We show which eigenvalue solvers (for the standard problem) are rank-conservative. §3 extends some results to the generalised eigenvalue problem. §4 shows how different OPM can be restarted explicitly and implicitly and how this restart can be combined with an implicit filtering step. The restart algorithm is illustrated with a numerical example. §5 closes the text with some conclusions.

1.1 Finding eigenvalues with orthogonal projection methods

The goal of iterative eigenvalue solvers is to produce an invariant subspace for a given matrix A or a given matrix pair (A, B) . For the iterative method to be competitive with the 'direct' eigenvalue solvers, such as QR, the dimension of this subspace must be (much) smaller than the dimension of the eigenproblem itself. An invariant subspace is represented by its orthonormal basis $V_k = [v_1, \dots, v_k]$, from which a number of eigenvalues are easily computed. We call $\mathcal{R}(V_k)$ a k -dimensional invariant subspace of a matrix A if

$$AV_k = V_k H_k, \quad \text{with } H_k = V_k^* AV_k \in \mathbf{C}^{k \times k}. \quad (1)$$

If (θ, z) is an eigenpair of H_k , then $(\theta, V_k z)$ is an eigenpair of A . If H_k is reduced to upper triangular form, then (1) represents a partial Schur form of A [8].

In practice however, the iterative methods compute only an approximate partial Schur form

$$AV_k = V_k H_k + F_k, \text{ with } V_k^* F_k = 0. \quad (2)$$

We call F_k the residual term or *residual* for short. We expect that if $\|F_k\|$ is small, (2) will be a good approximation of (1) and the eigenvalues of H_k will be good approximations of eigenvalues of A . A method that generates such an equation is called an *orthogonal projection method (OPM)*, since it projects the matrix A on the column space of V_k and computes the eigenvalues in this reduced space. More generally, we call each algorithm that computes an equation

$$AV_k L_k = V_k K_k + G_k, \quad (3)$$

with $V_k^* G_k = 0$ an OPM. We assume that H_k, L_k and K_k always have full rank, so (2) and (3) are equivalent: $H_k = K_k L_k^{-1}$. Nevertheless, we consider both forms, since L_k and K_k can have a special meaning, e.g. they can collect the orthogonalisation coefficients (see further). We also assume that V_k does *not contain* an exact eigenvector for the problem. If V_k would contain an eigenvector different from the wanted solution, then it must be removed from the subspace first. The eigenvalues θ of H_k are called *Ritz values* of A and the corresponding approximate eigenvectors y are called Ritz vectors, since their residuals are orthogonal to the subspace V_k : $r(y) = Ay - \theta y \perp V_k$. As defined in [13, 14, 19], we call $\theta' = \mu + \theta^{-1}$ a *Harmonic Ritz value*, if θ is a Ritz value of $(A - \mu I)^{-1}$, for some $\mu \in \mathbb{C}$. If $G_k = (A - \mu I)F_k$, with $V_k^* F_k = 0$, then

$$(A - \mu I)^{-1} V_k (K_k - \mu L_k) = V_k L_k + F_k.$$

The eigenvalues of the small, generalised eigenvalue problem (K_k, L_k) can then be seen as Harmonic Ritz values of A . The difference between Ritz and Harmonic Ritz values is not investigated here. We will not make this distinction between them, we assume that the approximate eigenvalues are always computed according to the equation that they emerge from, i.e. as the eigenvalues of H_k or (K_k, L_k) respectively.

For the generalised eigenvalue problem, the iterative methods build

$$AV_k = BV_k H_k + F_k, \text{ with } H_k = (V_k^* BV_k)^{-1} V_k^* AV_k \in \mathbb{C}^{k \times k} \quad (4)$$

or

$$AV_k L_k = BV_k K_k + G_k, \text{ with } V_k^* G_k = 0 \text{ or } G_k = (A - \mu B)F_k. \quad (5)$$

1.2 Projection matrices

The analysis of orthogonal projection matrices will, not surprisingly, use projections. In this section, we recall some facts about the projection matrices that are applied in the following paragraphs. The properties of projection matrices will be used to show that the difference between the solutions of an OPM in exact arithmetic and the solutions that are found when rounding errors are allowed, is 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 projector. Otherwise, \mathcal{P} is called an oblique projection matrix.*

If we apply \mathcal{P} on 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 of rank m can, using its singular value decomposition, always be written as $\mathcal{P} = Q_m Q_m^*$, with $Q_m^* Q_m = I$. Q_m then forms an orthogonal basis for the column space of \mathcal{P} . Inversely, we will denote the projection matrices that are generated by Q_m by

$$\mathcal{P}_{Q_m} = Q_m Q_m^*, \quad \mathcal{P}_{Q_m}^\perp = I - \mathcal{P}_{Q_m} = I - Q_m Q_m^*.$$

Analogously, an oblique projector matrix of rank m can be written as $BQ_m(Q_m^* BQ_m)^{-1} Q_m^*$, for a certain $B \in \mathbb{C}^{n \times n}$ and

$$\mathcal{P}_{Q_m}^B = I - BQ_m(Q_m^* BQ_m)^{-1} Q_m^*.$$

Finally, we will denote more generally a matrix that projects a vector on the subspace orthogonal to the columns of a given matrix F by \mathcal{P}_F^\perp , the corresponding oblique projector by \mathcal{P}_F^B .

Orthogonal projection methods are based on operations with projection matrices.

Definition 1.2 Given a matrix $F \in \mathbf{C}^{n \times m}$ with singular values $\sigma_1, \sigma_2, \dots$ then define

$$\text{rank}(F) = \#\{\sigma_i \mid \sigma_i \neq 0\} \text{ and define } \text{rank}(F, \varepsilon) = \#\{\sigma_i \mid \sigma_i \geq \varepsilon\}.$$

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 $F \in \mathbf{C}^{n \times k}$ of rank k , an orthogonal projector \mathcal{P} and a vector $f \in \mathbf{C}^n$. Let $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k$ be the singular values of F .

1. If $\sigma'_1 \geq \sigma'_2 \geq \dots \geq \sigma'_k$ are the singular values of $\mathcal{P}F$, then $\sigma'_1 \leq \sigma_1$, $\sigma'_2 \leq \sigma_2, \dots$, $\sigma'_k \leq \sigma_k$
2. Given a vector $v = Ft + e$ for some $t \in \mathbf{C}^k$, with $\|t\| = 1$ and $\mathcal{P}v = 0$. If $\sigma_p < \|e\|$, then $\sigma'_{p-1} < \|e\|$. Moreover, $\text{rank}(\mathcal{P}F, \|e\|) \leq p$.
3. If $\sigma''_1 \geq \dots \geq \sigma''_{k+1}$ denote the singular values of $[F, f]$, then

$$\begin{aligned} \sigma''_{k+1} &\leq \frac{1}{\sqrt{1 + h^*h}} \|\mathcal{P}_F^\perp f\| \\ \sigma''_i &\leq \kappa \sigma_i + \|\mathcal{P}_F^\perp f\| \\ \sigma''_i &\leq \sqrt{\sigma_i^2 + \|f\|^2}, \quad i = 1, \dots, k \end{aligned}$$

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

Proof See appendix A. □

Note that the conclusions of this lemma only hold for orthogonal projection matrices. When oblique projectors are used, it is possible that $\|\mathcal{P}_F^\perp f\| \gg \|f\|$. 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 $f \in \mathcal{R}(F)$ then $\|\mathcal{P}_F^\perp f\| = 0$ and the matrix $[F, f]$ will have a new singular value that is equal to zero. If $\|\mathcal{P}_F^\perp f\| = \varepsilon$ 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 $f \perp \mathcal{R}(F)$ 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 $\|\mathcal{P}_F^\perp f\| \simeq \|f\|$. The other singular values will not change much.
3. If $f \notin \mathcal{R}(F)$, but $\|f\|$ is small, then the singular values will not grow much either. The new singular value will be approximately equal to $\|f\|$.

Notation 1.1 In the continuation of this text, we will use the notation $r_k := Ay_k - \theta_k y_k = F_k z_k = \mathcal{P}_{y_k}^\perp Ay_k$ for the residual of the Ritz vector, and we define the residual of an arbitrary vector $u \in \mathbf{C}^n$ of unit length as $r(u) = \mathcal{P}_u^\perp Au = Au - (u^* Au)u$. Formally, we can thus write that $F_k = r(V_k)$. The subscript of a matrix will always refer to the iteration step in which it is computed (see further). Computations are organised such that this number also corresponds to the altering dimensions of the matrices: $F_k \in \mathbf{C}^{n \times k}$, $H_k \in \mathbf{C}^{k \times k}$, etc. The (i, j) -th element of a matrix is denoted by $(\cdot)_{i,j}$. The norm $\|\cdot\|$ that is used in this text, is the spectral norm.

2 Orthogonal Projection Methods

In this section, we present the framework of the *orthogonal projection methods* [12] for solving eigenvalue problems. We consider only the standard eigenvalue problem. We will give an extension to the generalised problem in the next section. However, it can already be seen that most conclusions are the same for both problems. Indeed, most eigenvalue solvers transform a generalised problem (A, B) into a standard one with $\bar{A} \leftarrow (A - \mu B)^{-1} B$ (the shift μ may change in each iteration step), which makes this generalisation reasonable.

This section is structured as follows. First, we show how almost any iterative eigenvalue solver can be seen within the framework of orthogonal projection method. Then we prove a recurrence relation for the residual F_k of these methods and we derive the class of *rank conservative* solvers, i.e. solvers for which the (numerical) rank of F_k does not grow. We show that the distinction between rank conservative solvers and other solvers, approximately corresponds to the distinction between iterative solvers that assume exact arithmetic and other solvers.

2.1 A template for the general OPM

Let us specify more precisely what is meant with an OPM. A distinctive way is to give an algorithm that covers any OPM. We underline that this template is far from an algorithm in pseudo code that is ready-to-implement. It only shows which entries must be computed to find a solution and at what stage of the algorithm that they can be computed. For many problems, these matrices come for ‘free’ and they must not be computed explicitly as shown.

Algorithm 1 *Orthogonal Projection Method (template)*

1. Given A , B , $V_1 = [v_1]$, $\|v_1\| = 1$
2. For $k = 1, 2, 3, \dots$
 - 2.1. $H_k = (V_k^* B V_k)^{-1} V_k^* A V_k$
 - 2.2. $F_k = A V_k - B V_k H_k$
 - 2.3. Compute (θ_k, z_k) from $H_k z_k = \theta_k z_k$
 - 2.4. $y_k \leftarrow V_k z_k$
 - 2.5. If convergence then exit
 - 2.6. Compute w_k
 - 2.7. $v_{k+1} \leftarrow \mathcal{P}_{V_k}^\perp w_k / \|\mathcal{P}_{V_k}^\perp w_k\|$
 - 2.8. $V_{k+1} \leftarrow [V_k, v_{k+1}]$

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 [3]. There are different possibilities to check the convergence at step 2.5., depending on the individual method. Most commonly, a measure for the residual norm $\|r(y_k)\| = \|A y_k - B y_k\|$ is used.

All OPMs are equivalent, except for the computation of the vector w_k . If the computation of w_k is done exactly by an algorithm that is fixed before the start of the OPM, then the subspace V_k and the approximate eigenvalue θ , only depend on the starting vector v_1 . Such an OPM is called an *exact OPM*. If w_k is computed, allowing large (random) errors, then the method is called *inexact*. For example, if w_k is found accurately as the solution of a linear system using Gaussian elimination, then the method can be called exact. If the same system is approximately solved iteratively, then the method will be inexact.

Example 2.1 Many well known iterative methods for solving eigenvalue problems can be fitted into this scheme.

- **Arnoldi’s method** [1] : If we choose $w_k = A v_k$, then we get Arnoldi’s method. The matrix H_k is upper Hessenberg and its k -th column h_k contains the orthogonalisation coefficients of the k -th iteration step : $w_k = V_{k+1} h_k$. Moreover, $F_k = [0 \ \dots \ 0 \ f_k]$ and $f_k = \eta v_{k+1}$. The next vector in the Arnoldi iteration is equal to the residual of equation (2) – which is called the Arnoldi equation. If A is symmetric, then H_k is a tridiagonal matrix and the method is the symmetric Lanczos procedure [10].
- **RKS** [15, 17, 16] : If we choose $w_k = (A - \mu_k I)^{-1} V_k t_k$, with $t_k \in \mathbf{C}^k$ some *continuation vector*, we get the Rational Krylov sequence method (RKS). RKS corresponds to an OPM that builds equation (3), where L_k contains the orthogonalisation coefficients and $K_k = L_k \text{diag}(\mu_i) + T_k$ is an upper Hessenberg matrix. T_k is the upper triangular matrix that collects the continuation vectors. As for the Arnoldi algorithm, G_k only contains one non-zero column, but $G_k = (A - \mu_k I) F_k$, with $V_k^* F_k = 0$. For RKS, the subdiagonal elements of L_k will always be non-zero – unless the method has converged.
- **Davidson** [2, 4] : If we choose $w_k = (A - \mu_k I)^{-1} (A - \theta_k I) y_k = (A - \mu_k I)^{-1} r_k$, then Algorithm 1 corresponds to Davidsons method. The Davidson method is well suited for use with an inexact linear system solver. If the system $(A - \mu_k I) w_k = r_k$ is solved exactly, then this method corresponds to an extended RKS method and $\text{rank}(F_k) = 1$. If the linear system is solved inexactly, e.g. with an iterative method, then $\text{rank}(F_k) > 1$.

- **Jacobi-Davidson** [19] : If w_k is computed as $\mathcal{P}_{y_k}^\perp (A - \theta_k I) \mathcal{P}_{y_k}^\perp w_k = -(A - \theta_k I) y_k$, by use of an iterative system solver, then the OPM is the Jacobi-Davidson algorithm. This method is a variation on the Davidson method. A short comparison between both methods can be found in [12]. Notice that for both Davidson variants, H_k is no longer an upper Hessenberg matrix.
- All these methods can also find solutions for the generalised eigenvalue problem, computing $w_k = (A - \mu B)^{-1} B u$ or $w_k = (A - \mu B)^{-1} (A - \sigma B) u$.

2.2 Rank-conservative eigenvalue solvers

The subspace V_k depends directly on the computation of the w_k . As for the Arnoldi algorithm, these computations can often be written as a recurrence relation. The same can be done for the residual. In this section, we prove the relation between subsequent residuals of an OPM. Using this relation, we show that many solvers have the property that the rank of their residual is one. These solvers are called *rank conservative*. The condition for a method to be rank conservative yields to a framework in which most OPMs fit (at least all methods of Example 2.1).

Theorem 2.1 *Following Algorithm 1, we have for the standard eigenvalue problem*

$$F_{k+1} = \mathcal{P}_{V_{k+1}}^\perp [F_k \quad r(v_{k+1})]. \quad (6)$$

Proof At step k of Algorithm 1, (2) can be decomposed into

$$A [V_k \quad v_{k+1}] = [V_k \quad v_{k+1}] \begin{bmatrix} H_k & h_c \\ h_r^* & \alpha \end{bmatrix} + [\bar{F}_k \quad f_{k+1}]$$

with $V_{k+1}^* \bar{F}_k = 0$ and $V_{k+1}^* f_{k+1} = 0$. Splitting this equation, results in

$$\begin{aligned} f_{k+1} &= Av_{k+1} - \alpha v_{k+1} - V_k h_c = (Av_{k+1} - \alpha v_{k+1}) - V_k V_k^* Av_{k+1} = (I - V_k V_k^*)(Av_{k+1} - \alpha v_{k+1}) \\ \bar{F}_k &= (AV_k - V_k H_k) - v_{k+1} h_r^* = (AV_k - V_k H_k) - v_{k+1} v_{k+1}^* AV_k = (I - v_{k+1} v_{k+1}^*)(AV_k - V_k H_k). \end{aligned}$$

Hence, we get

$$F_{k+1} = [\mathcal{P}_{v_{k+1}}^\perp F_k \quad \mathcal{P}_{V_k}^\perp r(v_{k+1})] = \mathcal{P}_{V_{k+1}}^\perp [F_k \quad r(v_{k+1})]$$

□

Definition 2.1 *If for an OPM, based on Algorithm 1, it holds that for any eigenvalue problem $\text{rank}(F_{k+1}) \leq \text{rank}(F_k)$, then the method is called rank-conservative. If, for any standard eigenvalue problem, $\text{rank}(F_{k+1}, \varepsilon) \leq \text{rank}(F_k, \varepsilon')$, with $\varepsilon' = O(\varepsilon)$, then the method is called ε -rank conservative.*

From Theorem 2.1, we derive a condition (for the standard eigenvalue problem) that a rank conservative solver must fulfill.

Proposition 2.2 *In exact arithmetic, it holds that $\text{rank}(F_{k+1}) \leq \text{rank}(F_k)$ iff*

$$r(v_{k+1}) \in \mathcal{R}(F_k) \cup \mathcal{R}(V_k) \quad \underline{\text{OR}} \quad v_{k+1} \in \mathcal{R}(F_k).$$

Numerically, we can say that if

$$\| \mathcal{P}_{F_k}^\perp \mathcal{P}_{V_k}^\perp r(v_{k+1}) \| < \varepsilon \quad \underline{\text{OR}} \quad \| \mathcal{P}_{F_k}^\perp v_{k+1} \| < \varepsilon,$$

then there exists a $\varepsilon' = O(\varepsilon)$ such that $\text{rank}(F_{k+1}, \varepsilon') < \text{rank}(F_k, \varepsilon)$.

Proof Since Theorem 2.1 shows that $F_{k+1} = [\mathcal{P}_{v_{k+1}}^\perp F_k \quad \mathcal{P}_{V_k}^\perp r(v_{k+1})]$, the rank of this matrix will not grow if $\mathcal{P}_{V_k}^\perp r(v_{k+1}) \in \mathcal{R}(F_k)$ or if $\text{rank}(\mathcal{P}_{v_{k+1}}^\perp F_k) = \text{rank}(F_k) - 1$. The first condition is true if $r(v_{k+1}) \in \mathcal{R}(F_k) \cup \mathcal{R}(V_k)$, the second one holds if $v_{k+1} \in \mathcal{R}(F_k)$.

Using Lemma 1.1, we can prove the second part. Say $\sigma_1 \geq \sigma_2 \geq \dots$ are the singular values of F_k and $\sigma'_1 \geq \sigma'_2 \geq \dots$ are the singular values of $[F_k \quad \mathcal{P}_{V_k}^\perp r(v_{k+1})]$. Combining the first condition with Lemma 1.1, gives

$$\begin{aligned} \sigma'_{k+1} &\leq \| \mathcal{P}_{F_k}^\perp \mathcal{P}_{V_k}^\perp r(v_{k+1}) \| < \varepsilon \\ \sigma_i < \varepsilon, \text{ then } \sigma_i &\leq \kappa \sigma_i + \| \mathcal{P}_{F_k}^\perp \mathcal{P}_{V_k}^\perp r(v_{k+1}) \| \leq (\kappa + 1) \varepsilon = \varepsilon'. \end{aligned}$$

Thus, $\text{rank}(F_{k+1}, \varepsilon') \leq \text{rank}([F_k \quad \mathcal{P}_{V_k}^\perp r(v_{k+1})], \varepsilon) \leq \text{rank}(F_k, \varepsilon)$. The proof is completed by noticing that if $v_{k+1} = F_k h + \varepsilon u$, with $\|u\| = 1$ and $\|h\| = O(1)$, then Lemma 1.1 can be applied : if we set $\varepsilon' = \|h\| \varepsilon$, then $\text{rank}(\mathcal{P}_{v_{k+1}}^\perp F_k, \varepsilon') \leq \text{rank}(F_k, \varepsilon) - 1$. □

Proposition 2.2 divides the set of rank conservative solvers into two different types. First, there is the Arnoldi type algorithm for which $v_{k+1} \in \mathcal{R}(F_k)$. This condition corresponds to setting $w_k \leftarrow AV_k t_k$, for some vector t_k .

The second type is a RKS or Davidson type algorithm. These methods compute v_{k+1} such that $r(v_{k+1}) \in \mathcal{R}(AV_k) \cup \mathcal{R}(V_k)$. In order to compute v_{k+1} (even implicitly) from $r(v_{k+1})$, these methods will need to solve a linear system.

2.3 The set of rank-conservative eigenvalue solvers

There are two major possibilities to compute v_{k+1} in each step of Algorithm 1. Arnoldi's method uses matrix vector products to find extremal eigenvalues for the problem. The matrix can be preconditioned in several ways, e.g. by shifting and inverting it. The second possibility is to generalise the idea of shift and invert. All these methods are rank conservative, as is shown in the Proposition 2.3.

Proposition 2.3 *At step 2.6. of Algorithm 1, w_k fulfills*

$$(\alpha_k A - \beta_k I)w_k = V_k p_k + F_k q_k = b_{k+1}$$

iff $\text{rank}(F_{k+1}) \leq \text{rank}(F_k)$.

Proof ' \Rightarrow ': Suppose $\alpha_k \neq 0$. If $w_k = \eta v_{k+1} + V_k t_k$, ($\eta \neq 0$), then define L such that

$$AV_{k+1}L = AV_{k+1} \begin{bmatrix} I & t_k - \frac{1}{\alpha_k} t_k \\ 0 & \eta \end{bmatrix} = V_{k+1} \begin{bmatrix} H_k & \frac{1}{\alpha_k} (p_k - H_k q_k + \beta_k t_k) \\ 0 & \frac{\eta \beta_k}{\alpha_k} \end{bmatrix} + \begin{bmatrix} F_k & 0 \end{bmatrix}.$$

Hence, $F_{k+1} = [F_k \ 0]L^{-1}$ and thus $\text{rank}(F_{k+1}) = \text{rank}(F_k)$. If $\alpha_k = 0$, then $v_{k+1} = -\frac{1}{\beta_k \eta} F_k q_k$. Following Proposition 2.2, the residual rank remains the same.

' \Leftarrow ': This follows directly from Proposition 2.2. \square

Proposition 2.3 shows that if a rank conservative eigenvalue solver is not an Arnoldi type method, then it is equivalent to a RKS type process. The following proposition formulates the same condition for inexact algorithms.

Proposition 2.4 *Given Algorithm 1, let at step k the system*

$$(A - \alpha_k I)\hat{w}_k = V_k p_k + F_k q_k = b_{k+1}$$

be solved inexactly such that $s_{k+1} = b_{k+1} - (A - \alpha_k I)\hat{w}_k \neq 0$. If $\|\mathcal{P}_{F_k}^\perp \mathcal{P}_{V_{k+1}}^\perp s_k\| \leq \varepsilon \|\mathcal{P}_{V_k}^\perp \hat{w}_k\|$, then $\text{rank}(F_{k+1}, \varepsilon') \leq \text{rank}(F_k, \varepsilon)$, with $\varepsilon' = O(\varepsilon)$, i.e. the method is ε' -rank conservative.

Proof Say $\alpha = v_{k+1}^* A v_{k+1}$ and $\hat{w}_k = V_k t + \eta v_{k+1}$, then

$$r(v_{k+1}) = Av_{k+1} - \alpha v_{k+1} = \frac{1}{\eta} (b_{k+1} - s_k - V_k (H_k - \alpha I)t - F_k t) + (\alpha_k - \alpha)v_{k+1}$$

$$\|\mathcal{P}_{F_k}^\perp \mathcal{P}_{V_k}^\perp r(v_{k+1})\| = \|\mathcal{P}_{F_k}^\perp \mathcal{P}_{V_{k+1}}^\perp r(v_{k+1})\| = \frac{1}{\eta} \|\mathcal{P}_{F_k}^\perp \mathcal{P}_{V_{k+1}}^\perp s_k\| \leq \varepsilon.$$

Using Proposition 2.2, this proves the theorem. \square

Example 2.2 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. The starting vector is $v_1 = [0.1, 0.1, \dots, 0.1]^*$ and v_{k+1} is computed from $w_k = (A - \mu_k I)v_k$, where $\mu_1 = 1$ and $\mu_i = \theta_i$, for $i > 1$. The approximation θ_i is computed as the rightmost eigenvalue of H_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 to 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 when the method converges.

The second part of the table shows what happens when we fix the shift $\mu_k = \theta_5$ for $k > 5$. 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 $\varepsilon = 1e-11$, then the ε -rank of F_k is one.

k	$ \theta_k - \lambda $	$\ Ay_k - \theta_k y\ $	$\ (A - \mu_k I)w_k - v_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	$2.3e-7$	$1.0e-5$	$4.3e-15$	$2.5e+1$	$3.0e-12$	$1.0e-14$
7	$9.4e-12$	$1.2e-9$	$1.3e-15$	$2.5e+1$	$5.6e-12$	$5.7e-14$
8	$2.1e-15$	$1.1e-13$	$4.6e-15$	$2.5e+1$	$8.0e-12$	$1.2e-13$

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

3 The generalised eigenvalue problem

The properties that we showed in Section 2 hold for the standard eigenvalue problem. Most of these properties also hold for the generalised eigenvalue problem, however there are some differences. The main difference lies in the fact that the generalised eigenvalue problem uses oblique projectors. If the matrix B is symmetric and positive (semi-)definite, then this correspondence could be totally carried out. One could then use the B -inner product, defined as $v^B w = v^* B w$ [11]. The basis V_k is then B -orthogonal and the residual has the form $B F_k$.

If we do not use B -orthogonality, then the recurrence relation (6) becomes somewhat complicated

$$F_{k+1} = [\mathcal{P}_{v_{k+1}}^B (F_k + B V_k \Delta H_k) \quad \mathcal{P}_{V_k}^B (r(v_{k+1}) + \Delta \alpha B v_{k+1})], \quad (7)$$

with $\alpha = (v_{k+1}^* B v_{k+1})^{-1} (v_{k+1}^* A v_{k+1})$ and

$$\Delta H_k = H_k - \bar{H}_{k+1} \quad \text{and} \quad \Delta \alpha = \alpha - (H_{k+1})_{k+1, k+1}.$$

where \bar{H}_{k+1} is equal to the $k \times k$ leading submatrix of H_{k+1} .

The generalised condition of Proposition 2.2 is not providing much useful information. One could ask whether an Arnoldi-like algorithm is possible for the generalised eigenvalue problem that only uses matrix vector products. From (7), it can be seen that such an algorithm computes v_{k+1} such that

$$B v_{k+1} \in \mathcal{R} (F_k + B V_k \Delta H_k).$$

In order to compute v_{k+1} , from F_k and $B V_k$, a linear system with the matrix B must be solved.

There is a simple extension of Proposition 2.3. A straightforward adaption of the proof says that $\text{rank}(F_{k+1}) \leq \text{rank}(F_k)$ if

$$(A - \beta_k B) w_k = B V_k p_k + F_k q_k,$$

for some β_k , p_k , q_k .

4 Filtering and restarting of an OPM

In practice, we cannot assume that Algorithm 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 (2) must be considered. A different reason for restarting the Arnoldi relation (for the generalised eigenvalue problem) was formulated in [11]: 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 [20] algorithm can be used to filter away these spurious eigenvalues. In [6], 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 H_k for the Jacobi-Davidson algorithm is proposed in [7]. 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 equation (4) to an equation

$$A V_{k-p}^+ = V_{k-p}^+ H_{k-p}^+ + F_{k-p}^+ \quad \text{or} \quad A V_{k-p}^+ L_{k-p}^+ = B V_{k-p}^+ K_{k-p}^+ + G_{k-p}^+ \quad (8)$$

with V_{k-p}^+ , F_{k-p}^+ , $G_{k-p}^+ \in \mathbf{C}^{n \times k-p}$ and H_{k-p}^+ , K_{k-p}^+ , $L_{k-p}^+ \in \mathbf{C}^{k-p \times k-p}$. 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*. An implicitly restart procedure, often can be seen as a *filtering* procedure. Suppose that B is nonsingular, then after the filtering steps, the new basis V_{k-p}^+ contains a filtered version of the old basis V_k or V_{k-p} ,

$$V_{k-p}^+ = \psi(B^{-1}A)V_{k-p},$$

where $\psi(B^{-1}A)$ is a polynomial or a rational function that depends on the restart algorithm.

4.1 Reducing the OPM relation

Considering the reduction of equation (4) or (5) to an equation of lower dimension (8). There are two possible ways to follow. Both methods perform first an orthogonal transformation on the subspace basis V_k . The first method then multiplies both sides of the equations with a matrix Z that is constructed such that the emerging rest term becomes zero. The second method hides this term in the residual F_k .

Proposition 4.1 *Given a set of matrices V_{k+1} , K_{k+1} , L_{k+1} , G_{k+1} that fulfill (5). Say $\text{rank}(G_{k+1}) = l$ and $G_{k+1} = gr^*$, $g \in \mathbf{C}^{n \times l}$, $r \in \mathbf{C}^{k+1 \times l}$. If $Q \in \mathbf{C}^{k+1 \times k}$ is an orthogonal matrix, $q \in \mathbf{C}^{k+1}$ a vector of unit length that is orthogonal to Q : $q^*Q = 0$, and $Z \in \mathbf{C}^{k+1 \times k}$ is a full rank matrix such that*

- (a) $q^*K_{k+1}Z = 0 = q^*L_{k+1}Z$ or
- (b) $q^*K_{k+1} = g_K r^*$ and $q^*L_{k+1} = g_L r^*$, $g_K, g_L \in \mathbf{C}^{1 \times l}$

then with $V_k^+ = V_{k+1}Q$, $K_k^+ = Q^*K_{k+1}Z$, $L_k^+ = Q^*L_{k+1}Z$ and (a) $G_k^+ = G_{k+1}Z$ or (b) $G_k^+ = G_{k+1}Z + wr^*Z$,

$$AV_k^+L_k^+ = BV_k^+K_k^+ + G_k^+.$$

Moreover, $\text{rank}(G_k^+) \leq l$

Proof Since $QQ^* + qq^* = I$, multiplication of (5) with Z gives

$$AV_{k+1}QQ^*L_{k+1}Z + \underbrace{AV_{k+1}qq^*L_{k+1}Z}_{0 \mid AV_{k+1}g_L r^* Z} = BV_{k+1}QQ^*K_{k+1}Z + \underbrace{BV_{k+1}qq^*K_{k+1}Z}_{0 \mid BV_{k+1}g_K r^* Z} + \underbrace{G_{k+1}Z}_{gr^* Z}.$$

Thus, $w = BV_{k+1}g_K - AV_{k+1}g_L$ and clearly $\text{rank}(G_k^+) \leq l$. \square

Because the reduction of the OPM relation is an operation that happens independently of the matrix pair (A, B) , the same operation must hold for all possible matrix pairs that fit into the relation. Therefore, the conditions of Proposition 4.1 are in practice *necessary and sufficient* conditions for restarting an OPM. The two ways to restart an OPM are clear from the proof : the first solution is constructed in such a way that the ‘residual’ wr^*Z of the restart 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 with the same rank as the non-restarted one.

(a) 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 decomposition computes a set of orthogonal matrices $Q_k, Z_k \in \mathbf{C}^{k \times k}$, such that $Q_k^*K_kZ_k = T_K$ and $Q_k^*L_kZ_k = T_L$, with triangular matrices $T_K, T_L \in \mathbf{C}^{k \times k}$. If we multiply (5) on the right by Z_k , then

$$\begin{aligned} AV_kQ_kQ_k^T L_kZ_k &= BV_kQ_kQ_k^T K_kZ_k + F_kZ_k \\ AV_kQ_kT_L &= BV_kQ_kT_K + F_kZ_k. \end{aligned}$$

If $F_kZ_k = 0$, then this would be a partial Schur decomposition of (A, B) . We set $V_kQ_k = V_k^+$ 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 easily be reduced to (8) by deleting the last p columns. If Q_k 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 $\text{rank}(F_k) = \text{rank}(F_{k-p}^+)$.

Obviously, the matrices involved in this restarting procedure can also be found without using a Schur decomposition.

Lemma 4.2 Given matrices V_{k+1} , K_{k+1} , L_{k+1} , G_{k+1} as in Proposition 4.1. Let $y_i = V_{k+1}L_{k+1}z_i$, $i = 1, \dots, k+1$ be the Ritz pairs of the problem. Define the matrices Q and Z such that Q is orthogonal and

$$\mathcal{R}(Z) = \mathcal{R}(z_1, \dots, z_{m-1}, z_{m+1}, \dots, z_{k+1}) \text{ and } \mathcal{R}(Q) = \mathcal{R}([L_{k+1}Z, u]), u^*L_{k+1} = 0,$$

then $V_k^+ = V_{k+1}Q$, $K_k^+ = Q^*K_{k+1}Z$, $L_k^+ = Q^*L_{k+1}Z$ and $G_k^+ = G_{k+1}Z$ define an OPM relation that has the same Ritz vectors, except for y_m .

Proof We only have to prove that $q^*K_{k+1}Z = 0 = q^*L_{k+1}Z$. Since there exist a matrix P for which $QP = L_{k+1}Z$, $q^*L_{k+1}Z$ must be zero. Also, for all $i \neq m$, $q^*K_{k+1}z_i = \theta_i q^*L_{k+1}z_i = 0$, which proves $q^*K_{k+1}Z = 0$. \square

(b) Implicit restarting based on a shifted QR decomposition

IRA gives us another way to look at a restarting algorithm. Given a shift μ , IRA combines the basis vectors V_{k+1} of a Krylov subspace $\mathcal{K}_{k+1}(A, v_1)$ to a new orthogonal basis of $\mathcal{K}_k(A, (A - \mu I)v_1)$. Thus, it computes the results V_k^+ , H_k^+ of an Arnoldi iteration with a new starting vector $(A - \mu I)v_1$ in an implicit way. This implicit multiplication does not come for free : the resulting basis will have a dimension k instead of $k+1$, so we sacrifice one Arnoldi step. This corresponds to the one matrix-vector multiplication that we 'gain' by the implicit restart.

Suppose that we ran the Arnoldi algorithm to obtain V_{k+1} and H_{k+1} (which is a Hessenberg matrix) with a residual $F_{k+1} = f_{k+1}e_{k+1}^*$. Given a shift μ , we can rewrite (4) as

$$\begin{aligned} (A - \mu I)V_{k+1} &= V_{k+1}(H_{k+1} - \mu I) + F_{k+1} \\ &= V_{k+1}QR + F_{k+1} = V_{k+1} \begin{bmatrix} Q_1 & q_2 \end{bmatrix} \begin{bmatrix} R_1 & r_2 \\ 0 & r_3 \end{bmatrix} + F_{k+1}, \end{aligned}$$

where Q is a square orthogonal upper Hessenberg matrix of size $(k+1) \times (k+1)$ and R is upper triangular with the same dimensions.

1. If we delete the last column of both sides of the equation then the F_{k+1} vanishes :

$$\begin{aligned} (A - \mu I)V_k &= V_{k+1}Q_1R_1 + 0 \\ \Rightarrow V_{k+1}Q_1 &= (A - \mu I)V_kR_1^{-1}. \end{aligned}$$

We will define the new basis $V_k^+ = V_{k+1}Q_1$.

2. On the other hand, if we multiply the equation from the right by Q_1 , we get an equation similar to (4) :

$$\begin{aligned} (A - \mu I)V_{k+1} &= V_{k+1}Q_1 \begin{bmatrix} R_1 & r_2 \end{bmatrix} Q_1 + V_{k+1}q_2 \begin{bmatrix} 0 & r_3 \end{bmatrix} Q_1 + F_{k+1}Q_1 \\ &= V_k^+(H_k^+ - \mu I) + F_k^+, \\ \text{with } H_k^+ &= \begin{bmatrix} R_1 & r_2 \end{bmatrix} Q_1 + \mu I = Q_1^*H_{k+1}Q_1 \\ F_k^+ &= V_{k+1} \begin{bmatrix} 0 & q_2r_3 \end{bmatrix} Q_1 + F_{k+1}Q_1. \end{aligned}$$

So, because $Q_1 \in \mathbb{C}^{(k+1) \times k}$ is upper Hessenberg, F_k^+ inherits the form of F_{k+1} :

$$F_k^+ = \begin{bmatrix} 0 & \dots & 0 & f_k^+ \end{bmatrix} \Rightarrow \text{rank}(F_k) = 1.$$

It is easy to see that H_k^+ is upper Hessenberg. Thus, the matrices V_k^+ , H_k^+ , F_k^+ can be seen as if they were generated by an Arnoldi process with a different starting vector. The restarting process can be repeated until the dimension of V_k has decreased down to a satisfying level.

One of the advantages of this method is that we can chose the subsequent shifts μ to be equal to the 'unwanted' eigenvalues of H_k , i.e. to the eigenvalues of H_k that approximate eigenvalues of A that we do not want to compute. The implicit multiplication with $(A - \mu I)$ then filters out of V_k the corresponding, unwanted Ritz-vectors. Indeed, if μ is chosen as an eigenvalue of H_k , then $r_3 = 0$ and the restarting scheme fulfills condition (a) of Proposition 4.1. In that case, it implements Lemma 4.2. The μ can also be chosen such that the filter accelerates the convergence of the algorithm in the region of the complex plane that is focussed.

Proposition 4.3 *A restarting Algorithm that reduces the size of the subspace V_{k+1} with one, can only be a polynomial (of degree one) filter if the rank of the residual equals one.*

Proof Due to the correspondence with an Arnoldi or RKS process (Proposition 2.3), a relation with $\text{rank}(F_k) = 1$ can be filtered while restarting. This is shown in [6, 20]. Inversely, for a rational filter, given a full rank matrix $P \in \mathbf{C}^{k \times k}$ (set $\bar{P} = [P^* \ 0]^*$), and given any vector $t_{k+1} \in \mathbf{C}^{k+1}$

$$\begin{aligned} V_{k+1}Q &= V_k^+ &= (\alpha A - \beta I)^{-1}(A - \mu I)V_k P \\ AV_{k+1}(\alpha Q + \bar{P}) &= V_{k+1}(\beta Q - \mu \bar{P}) \\ AV_{k+1}[\alpha Q + \bar{P} \ t_{k+1}] &= V_{k+1}[(\beta Q - \mu \bar{P}) (V_{k+1}^* AV_{k+1} t_{k+1})] + [0 \cdots 0 \ \mathcal{P}_{V_{k+1}}^\perp AV_{k+1} t_{k+1}], \end{aligned}$$

Thus, $\text{rank}(F_k) = \text{rank}(\mathcal{P}_{V_{k+1}}^\perp AV_{k+1}) = 1$. Setting $\alpha = 0$ and $\beta = 1$, gives the result for a polynomial filter. \square

4.2 Inexact filtering

Proposition 4.3 showed that we cannot filter the subspace using a restarting procedure that decrements the size of the subspace V_{k+1} with 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 4.4 *Given V_{k+p} , H_{k+p} , F_{k+p} , such that $AV_{k+p} = BV_{k+p}H_{k+p} + F_{k+p}$. Suppose that $F_{k+p} = [0 \ F_p]$, $F_p \in \mathbf{C}^{n \times p}$ and $\text{rank}(F_{k+p}) = p$. If, given a parameter μ , the matrix Q_1 contains the first k columns of the QR decomposition of $H_{k+p} - \mu I = [Q_1 \ Q_2]R$, then $H_k^+ = Q_1^* H_{k+p} Q_1$ and $V_k^+ = V_{k+p} Q_1$ form a restarted relation*

$$AV_k^+ = BV_k^+ H_k^+ + G_k^+, \text{ with } \text{rank}(G_k^+) \leq p.$$

Moreover, if B is nonsingular, then

$$\mathcal{R}(V_k^+) = \mathcal{R}(B^{-1}(A - \mu B)V_k).$$

In order to show exactly what happens in an inexact filtering step, we give a constructive proof. This will make Algorithm 2 trivial to understand.

Proof If we compute a ‘skinny’ QR-factorisation for F_{k+p}

$$F_p = U_p T_p, U_p^* U_p = I, \text{ with } U_p \in \mathbf{C}^{n \times p}, T_p \in \mathbf{C}^{p \times p}$$

where T_p is upper triangular, then we can rewrite (4) as

$$\begin{aligned} (A - \mu B)V_{k+p} &= [BV_{k+p} \ U_p] \begin{bmatrix} H_{k+p} - \mu I \\ 0 \ T_p \end{bmatrix} = [BV_{k+p} \ U_p] QR \\ &= [V_{k+p} \ U_p] \begin{bmatrix} Q_1 & Q_2 \\ 0 & Q_4 \end{bmatrix} \begin{bmatrix} R_1 & R_2 \\ 0 & R_4 \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} \text{with } R_1, Q_1 &\in \mathbf{C}^{k+p \times k}, Q_1^* Q_1 = I \\ R_2, Q_2 &\in \mathbf{C}^{k+p \times p}, Q_2^* Q_2 = I, Q_2^* Q_1 = 0 \\ R_4, R_4 &\in \mathbf{C}^{p \times p}. \end{aligned}$$

1. If we delete the last p columns of both sides of the equation, we get the definition of V_k^+

$$\begin{aligned} (A - \mu B)V_k &= BV_{k+p} Q_1 R_1 \\ V_k^+ = V_{k+p} Q_1 &= B^{-1}(A - \mu B)V_k R_1^{-1}. \end{aligned}$$

2. On the other hand, multiplying both sides from the right by Q_1

$$\begin{aligned} (A - \mu B)V_{k+p} Q_1 &= BV_{k+p} Q_1 [R_1 \ R_2] Q_1 + (BV_{k+p} Q_2 + U_p Q_4) [0 \ R_4] Q_1 \\ (A - \mu B)V_k^+ &= BV_k^+ (H_k^+ - \mu I) + F_k^+, \text{ with } H_k^+ = [R_1 \ R_2] Q_1 + \mu I \\ F_k^+ &= (BV_{k+p} Q_2 + U_p Q_4) [0 \ R_4] Q_1 \\ \text{rank}(F_k^+) &\leq p. \end{aligned}$$

□

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

$$AV_k^+G = V_k^+G(G^*H_k^+G) + F_k^+G, \text{ with } G^*G = I = GG^* \text{ and } F_k^+G = [0 \ \cdots \ 0 \ \star].$$

Notice that G can be computed from Q_1 .

The results for the inexact filtering procedure are summarised in the Algorithm 2. We do not need to know the matrix T_p to compute the restarted equation.

Algorithm 2 *Inexact filtering*

1. Given $F_k = [0 \ \cdots \ 0, \star]$, H_k
2. Set $[\tilde{H}_k \ \tilde{H}_p] = H_{k+p} - \mu I$
3. Compute Q_1, R_1 from $\tilde{H}_k = Q_1R_1$
4. Set $R_2 \leftarrow Q_1^*\tilde{H}_p$
5. Compute G from Q_1
6. Set $V_k^+ \leftarrow V_{k+p}Q_1G$
7. Set $H_k^+ \leftarrow G^*[R_1 \ R_2]Q_1G + \mu I$

4.3 Example

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 . 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 2.

We iterated the OPM algorithm $k = 10$ times with a starting vector $v_1 = [1 \ \cdots \ 1]^*/\sqrt{n}$ and $w_k = (A - \sigma B)^{-1}By_k$, with a shift $\sigma = -1$. Setting the matrices $K_k = (V_k^*BV_k)^{-1}$ 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 H_k or F_k in relation

$$(A - \sigma B)^{-1}BV_k = V_kH_k + F_k,$$

but we can compute G_k from

$$AV_kL_k = BV_kK_k + G_k.$$

Setting $H_k = L_k(K_k - \sigma L_k)^{-1}$ and $F_k = (A - \sigma B)^{-1}G_k(K_k - \sigma L_k)^{-1}$, will give the input matrices for Algorithm 2. Indeed, if $G_k(K_k - \sigma L_k)^{-1}$ has the right form, then F_k will have the same form (we do not need F_k explicitly). The linear system was solved using GMRES to a tolerance of $10e-6$, for the first 8 steps and to a tolerance of $10e-3$ for the last 2 steps (since inexact iterative methods converge asymptotically well, it is better to use a higher tolerance at the beginning of the process [5]). The results are shown in Table 2.

Implicitly restarting the relation seems not to conflict with the convergence. 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 $p = 3$, all spurious eigenvalues with a positive real part are filtered away. The filter is then applied with very good accuracy but some convergence is lost. It should be noted that the accuracy of the filter is proportional to σ_{p+1} .

	$k = 10$	$p = 1$	$p = 2$	$p = 3$
$\ r(y_k)\ $	$1.1e-7$	$1.1e-7$	$8.4e-7$	$1.3e-5$
# positive	1	1	2	0
$\ \mathcal{P}_{V_{k-p}^+}^\perp W_{k-p}\ $		$3.7e-6$	$1.1e-7$	$7.6e-9$
σ_1	$1.2e-3$	$1.3e-2$	$5.0e-4$	$1.6e-2$
σ_2	$4.4e-6$	$3.5e-5$	$1.0e-4$	$6.2e-5$
σ_3	$3.0e-7$	$3.0e-7$	$3.1e-8$	$9.9e-7$

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 $p = 1, 2, 3$.

5 Conclusions

The set of orthogonal projection methods can be structured following two different criterions. First, there is the distinction between rank conservative solvers and non rank conservative solvers. Most algorithms known, are rank conservative if they use direct solvers and assume exact arithmetic. Therefore, exact methods can be considered rank conservative; inexact methods always have a residual with a higher rank. A different possibility is to distinct the methods by the way they weave their residual into the next iteration vector : if the new vector is chosen in the range of the residual matrix, then we have the Arnoldi type; if a linear system with A and B is solved, then it is Davidson type.

An orthogonal projection method can always be restarted, implicitly or (more) explicitly. The implicit restart can often be combined with an implicit filtering step. The rank of the residual turns out to be decisive here. Rank conservative methods can be implicitly filtered, other methods can not. For the Arnoldi type methods, this filter is a polynomial filter. For Davidson type methods it will be a rational filter. The application of an implicit filter depends on an implicit relation that is assumed to be exact. Therefore, the accuracy of this filter also depends on this relation. ϵ -rank conservative methods will be restarted less accurately than ‘true’ rank conservative methods.

There is a lot of work that must be done here. In practice, Algorithm 2 is too general to restart a well structured algorithm, such as RKS. Different methods have different properties that can be exploited while restarting. The implicit filter can be used to reduce the size of the subspace, and not only to filter away spurious eigenvalues. It is not clear whether in that case an explicit filter could not perform better. It is well known that IRA can ‘fail’, due to the forward instability of the QR method that it depends on. Obviously, this possibility must also be taken into account while restarting other methods.

Acknowledgements

The authors are grateful to Karl Meerbergen, who gave the initial impetus to this text.

This research was supported by the National Fund for Scientific Research (NFWO), project Lanczos, grant #2.0042.93 and by the Human Capital and Mobility project ROLLS of the European Community under contract ERBCHRXCT930416.

References

- [1] W.E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quart. Appl. Math.*, 9:17–29, 1951.
- [2] M. Crouzeix, B. Philippe, and M. Sadkane. The Davidson method. *SIAM J. Sci. Comput.*, 15:62–76, 1994.
- [3] J.W. Daniel, W.B. Gragg, L. Kaufman, and G.W. Stewart. Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization. *Math. Comp.*, 30:772–795, 1976.
- [4] E.R. Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *J. Comput. Phys.*, 17:87–94, 1975.

- [5] G. De Samblanx and A. Bultheel. On the convergence and the restarting of inexact eigenvalue solvers. TW-REPORT, Dept. Computing Science, K.U.Leuven, 1996.
- [6] G. De Samblanx, K. Meerbergen, and A. Bultheel. The implicit application of a rational filter in the RKS method. Report TW239, Department of Computing Science, K.U.Leuven, 1996.
- [7] D.R. Fokkema, G.L.G. Sleijpen, and H.A. Van der Vorst. Jacobi-Davidson style QR and QZ algorithms for the partial reduction of matrix pencils. Technical Report 941, Dept. of Mathematics, Utrecht University, jan 1996.
- [8] G. Golub and C. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 2nd edition, 1989.
- [9] E.J. Grimme, D.C. Sorensen, and P. Van Dooren. Model reduction of state space systems via an implicitly restarted Lanczos method. Report CRPC-TR94458, Center for Research on Parallel Computing, Rice University, Houston, May 1994.
- [10] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Stand.*, 45:255–281, 1950.
- [11] K. Meerbergen and A. Spence. Implicitly restarted Arnoldi and purification for the shift-invert transformation. *Math. Comp.*, 1996. Accepted for publication.
- [12] Karl Meerbergen. *Robust methods for the calculation of rightmost eigenvalues of nonsymmetric eigenvalue problems*. PhD thesis, Dept. of Computing Science, K.U.Leuven, March 1996.
- [13] R.B. Morgan. Computing Interior Eigenvalues of Large Matrices. *Linear Alg. Appl.*, 154–156:289–309, 1991.
- [14] C. Paige, B.N. Parlett, and H.A. Van der Vorst. Approximate solutions and eigenvalue bounds from Krylov subspaces. *Num. Lin. Alg. Appl.*, 2:115–133, 1995.
- [15] A. Ruhe. Rational Krylov sequence methods for eigenvalue computation. *Linear Alg. Appl.*, 58:391–405, 1984.
- [16] A. Ruhe. The Rational Krylov algorithm for nonsymmetric eigenvalue problems, III: Complex shifts for real matrices. *BIT*, 34:165–176, 1994.
- [17] A. Ruhe. Rational Krylov algorithms for nonsymmetric Eigenvalue problems, II: Matrix pairs. *Linear Alg. Appl.*, 197/198:283–296, 1994.
- [18] A. Ruhe. Rational Krylov, a practical algorithm for large sparse nonsymmetric matrix pencils. Technical Report UCB/CSD-95-871, Computer Science Division, University of California, Berkeley, 1995.
- [19] G.L.G. Sleijpen and H.A. Van der Vorst. A generalized Jacobi-Davidson iteration method for linear eigenvalue problems. Preprint 856, University of Utrecht, Department of Mathematics, 1994.
- [20] D.C. Sorensen. Implicit application of polynomial filters in a k -step Arnoldi method. *SIAM J. Matrix Anal. Applic.*, 13:357–385, 1992.
- [21] J.H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, 1965.

A Proof of Lemma 1.1

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

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

where \mathcal{X}_i is the set of subspaces of dimension i [8, 21]. Therefore,

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

We also rely on the fact that

$$\max_{X \in \mathcal{X}_i} \min_{x \in X} \{\cdot\} \leq \max_{X \in \mathcal{X}_i} \min_{x \in X, x \neq t} \{\cdot\} = \max_{X \in \mathcal{X}_i, t \notin X} \min_{x \in X} \{\cdot\} \leq \max_{X \in \mathcal{X}_{i-1}} \min_{x \in X} \{\cdot\},$$

i.e. the maximum over a ‘smaller’ set \mathcal{X}_{i-1} is larger than the maximum over the original set \mathcal{X}_i .

Proof

$$1. \sigma'_i = \max_{X \in \mathcal{X}_i} \min_{x \in X} \frac{\|\mathcal{P}F\mathbf{x}\|}{\|\mathbf{x}\|} \leq \max_{X \in \mathcal{X}_i} \min_{x \in X} \frac{\|F\mathbf{x}\|}{\|\mathbf{x}\|} = \sigma_i.$$

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

$$\begin{aligned} \sigma'_{p-1} &= \min_{X \in \mathcal{X}_{k-p+2}} \left\{ \max_{x \in X, \|\mathbf{x}\|=1} \|\mathcal{P}F\mathbf{x}\| \right\} \leq \min_{X \in \mathcal{X}_{k-p+2}} \left\{ \max_{x \in X \setminus \{t\}, \|\mathbf{x}\|=1} \{\|\mathcal{P}F\mathbf{x}\|, \|\mathcal{P}Ft\|\} \right\} \\ &\leq \min_{X \in \mathcal{X}_{k-p+1}} \left\{ \max_{x \in X, \|\mathbf{x}\|=1} \|\mathcal{P}F\mathbf{x}\|, \|e\| \right\} \leq \min\{\sigma_p, \|e\|\} \end{aligned}$$

3. Say $f = Fh + g$ and let $y = [x^* \ \alpha^*]^*$, where $x \in \mathbf{C}^k$. 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, $\alpha_0^2 = 1 - x_0^*x_0 \leq 1$ and

$$\begin{aligned} \sigma''_{k+1} &= \min_{x^*x + \alpha^2 = 1} \|F\mathbf{x} + f\alpha\| = \min_{x^*x + \alpha^2 = 1} \|F(\mathbf{x} + \alpha h) + \alpha \mathcal{P}_F^\perp f\| \\ &\leq \min \left\{ \alpha_0 \|\mathcal{P}_F^\perp f\|, \min_{x + \alpha h \neq 0} \|F(\mathbf{x} + \alpha h)\| + \alpha \|\mathcal{P}_F^\perp f\| \right\} \leq \alpha_0 \|\mathcal{P}_F^\perp f\| \end{aligned}$$

For the other singular values, we can write

$$\begin{aligned} \sigma''_i &= \max_{X \in \mathcal{X}_i} \min_{x \in X, \alpha} \|F\mathbf{x} + \alpha f\| \leq \max_{x, \alpha} \min_{\alpha} \|F(\mathbf{x} + \alpha h)\| + \alpha \|\mathcal{P}_F^\perp f\| \\ &\leq \max \left\{ \min \left\{ \alpha_0 \|\mathcal{P}_F^\perp f\|, \min_{x \in X \setminus \{h\}} \frac{\|F(\mathbf{x} + \alpha h)\|}{\|\mathbf{x} + \alpha h\|} \|\mathbf{x} + \alpha h\| + \alpha \|\mathcal{P}_F^\perp f\| \right\} \right\} \\ &\leq \max \left\{ \min \left\{ \alpha_0 \|\mathcal{P}_F^\perp f\|, \min_{x \in X \setminus \{h\}} \frac{\|F(\mathbf{x} + \alpha h)\|}{\|\mathbf{x} + \alpha h\|} \kappa + \|\mathcal{P}_F^\perp f\| \right\} \right\} \leq \sigma_i \kappa + \|\mathcal{P}_F^\perp f\|. \end{aligned}$$

Finally,

$$\begin{aligned} \sigma''_i &\leq \min_X \max_{x, \alpha} \|F\mathbf{x}\| + \alpha \|f\| \leq \min_X \max_{\alpha} \frac{\|F\mathbf{x}\|}{\|\mathbf{x}\|} \sqrt{1 - \alpha^2} + \alpha \|f\| \\ &\leq \max_{\alpha} \sigma_i \sqrt{1 - \alpha^2} + \alpha \|f\|. \end{aligned}$$

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 $\sqrt{\sigma_i^2 + \|f\|^2}$.

□