Chapter 7

Arnoldi and Jacobi-Davidson methods for generalized eigenvalue problems $A\mathbf{x} = \lambda B\mathbf{x}$ with singular B

Abstract. In many physical situations, a few specific eigenvalues of a large sparse generalized eigenvalue problem $A\mathbf{x} = \lambda B\mathbf{x}$ are needed. If exact linear solves with $A - \sigma B$ are available, implicitly restarted Arnoldi with purification is a common approach for problems where B is positive semidefinite. In this chapter, a new approach based on implicitly restarted Arnoldi will be presented that avoids most of the problems due to the singularity of B. Secondly, if exact solves are not available, Jacobi-Davidson QZ will be presented as a robust method to compute a few specific eigenvalues. Results are illustrated by numerical experiments.

Key words. sparse generalized eigenvalue problems, purification, semi-inner product, implicitly restarted Arnoldi, Jacobi-Davidson, preconditioning

7.1 Introduction

Large sparse generalized eigenvalue problems of the form

$$A\mathbf{x} = \lambda B\mathbf{x}, \qquad \mathbf{x} \neq 0, \tag{7.1.1}$$

with $A, B \in \mathbb{R}^{n\times n}$, $\mathbf{x} \in \mathbb{C}^n$ and $\lambda \in \mathbb{C}$, arise in physical situations like stability analysis of the discretized Navier-Stokes equations. Typically, the matrix A is nonsymmetric and of full rank, and B is singular. The pencil (A, B) is regular, that is, $A-\gamma B$ is singular only for a finite number of $\gamma \in \mathbb{C}$. Because B is singular, (7.1.1) can have eigenvalues at infinity, which are of no physical relevance, but may lead to numerical difficulties. In practice, one is often interested in the few left- or

rightmost *finite* eigenvalues that determine the stability, and hence one wants to avoid approximations to eigenvalues at infinity. This chapter is concerned with the computation of a few left- or rightmost eigenvalues of large generalized eigenvalue problems.

One way to compute a few eigenvalues of (7.1.1) close to $\sigma \in \mathbb{C}$ is to apply Arnoldi's method to the shift-and-invert transformation $S = (A - \sigma B)^{-1}B$:

$$S\mathbf{x} = \tilde{\lambda}\mathbf{x}, \qquad \mathbf{x} \neq 0. \tag{7.1.2}$$

An eigenpair $(\lambda, \mathbf{x})$ of (7.1.1) corresponds to an eigenpair $(\tilde{\lambda} = (\lambda - \sigma)^{-1}, \mathbf{x})$ of (7.1.2). Hence, the infinite eigenvalues of (7.1.1) correspond to eigenvalues $\tilde{\lambda} = 0$ of (7.1.2). Arnoldi's method may compute approximations $\tilde{\theta}$ to $\tilde{\lambda} = 0$. These approximations are known as *spurious* eigenvalues and after back transformation via $\theta = \tilde{\theta}^{-1} + \sigma$, they may be hard to distinguish from wanted eigenvalues, which typically reside in the exterior of the spectrum. This problem has been addressed for the symmetric nondefective problem [46, 104] and for the defective problem [46]. The ideas presented there are extended to the nonsymmetric defective case in [97], where the implicitly restarted Arnoldi method [85, 146] is implemented with a B semi-inner product and purification. Purification is a technique to remove unwanted components from Arnoldi vectors and approximate eigenvectors, and will be explained in more detail in Section 7.3. A new strategy will be presented that, by exploiting the structure of (7.1.1), reduces the corruption by unwanted components significantly.

The scheme based on the Arnoldi method fails to be applicable if the linear system solves with $A - \sigma B$, e.g. via the LU-factorization of $A - \sigma B$, are inaccurate or not computable within reasonable time. The Jacobi-Davidson QZ method [51] has the advantage that it computes with the matrices A and B directly and that in principle no inverses or exact solves are needed; furthermore, it poses no restrictions on the matrices A and B, so it is also applicable if B is not symmetric positive semidefinite or if both A and B are singular. In Section 7.4, it is shown that the Jacobi-Davidson method with harmonic Petrov values has some favorable properties with respect to purification. If, additionally, a preconditioner is available in the form of an LU-factorization, the correction equation can be solved efficiently and purification is obtained automatically.

Throughout this chapter, it will be assumed that the leftmost finite eigenvalues are wanted. This is a natural assumption in practical situations where the stability of steady states for a number of different parameter values is to be determined, see for instance [33, 69]: not only the leftmost eigenvalue is of interest, but also the eigenvalue(s) close to the leftmost that may become the leftmost for different parameter values. The theory extends readily to problems where the rightmost finite eigenvalues are wanted.

The outline of the chapter is as follows. Some properties of generalized eigenvalue problems are described in Section 7.2. In Section 7.3, the Arnoldi method with purification is explained and the new scheme is presented, illustrated by numerical examples. The approach based on the JDQZ method is described in Section 7.4. Section 7.5 concludes.

7.2 Some properties of generalized eigenvalue problems

Central point of the discussion is the generalized eigenproblem

$$A\mathbf{x} = \lambda B\mathbf{x}, \qquad \mathbf{x} \neq 0,$$

with $A, B \in \mathbb{R}^{n\times n}$, $\mathbf{x} \in \mathbb{C}^n$ and $\lambda \in \mathbb{C}$. Only regular matrix pencils will be considered, i.e. pencils (A, B) for which $A - \gamma B$ is singular only for a finite number of $\gamma \in \mathbb{C}$. Note that B is allowed to be singular. The corresponding ordinary eigenproblem is

$$S\mathbf{x} = \tilde{\lambda}\mathbf{x}, \qquad \mathbf{x} \neq 0,$$

with $S = (A - \sigma B)^{-1}B$ for a σ such that $A - \sigma B$ is non singular. A generalized eigenpair $(\lambda, \mathbf{x})$ corresponds to an ordinary eigenpair $(\tilde{\lambda} = (\lambda - \sigma)^{-1}, \mathbf{x})$ of (7.1.2). The generalized eigenvalues can be computed via the relation $\lambda = \tilde{\lambda}^{-1} + \sigma$.

The eigenspace corresponding to the infinite eigenvalues is the null space $\mathcal{N}(S)$ of S:

$$V_\infty = \mathcal{N}(S) = \mathcal{N}(B) = \{\mathbf{x} \in \mathbb{R}^n \mid B\mathbf{x} = 0\}.$$

The eigenvectors corresponding to the finite eigenvalues span a real invariant subspace of S and form a subspace of the range of S^{j_s}, $\mathcal{R}(S^{j_s})$:

$$V_{finite} \subseteq \mathcal{R}(S^{j_s}) = \{\mathbf{x} \in \mathbb{R}^n \mid ((A - \sigma B)^{-1}B)^{j_s}\mathbf{y} = \mathbf{x}, \mathbf{y} \in \mathbb{R}^n\}, \tag{7.2.1}$$

where j_s is the size of the largest Jordan block corresponding to the zero eigenvalue of S. The generalized null space $\mathcal{G}(S)$ of S is defined as the complement in $\mathcal{N}(S^{j_s})$ of $\mathcal{N}(S)$

$$\mathcal{G}(S) = \mathcal{N}(S^{j_s})\backslash\mathcal{N}(S),$$

It follows that $\mathbb{R}^n = \mathcal{R}(S^{j_s}) + \mathcal{G}(S) + \mathcal{N}(S)$. Note that (7.2.1) becomes an equality if for all finite eigenvalues the algebraic multiplicity is equal to the geometric multiplicity.

It is important to keep in mind that eigenvectors $\mathbf{x}$ corresponding to finite eigenvalues do not necessarily satisfy $\mathbf{x} \perp V_\infty$. In other words, in general it does not hold that $\mathcal{R}(S) \perp \mathcal{N}(S)$. So restricting the search space to $\mathcal{R}(B)$, to avoid approximations to infinite eigenvalues, is not effective. Only if A is block upper triangular and B is block diagonal it is effective, but then the problem can also easily be reduced to a smaller problem by considering the non-zero diagonal blocks.

This chapter is concerned with block structured generalized eigenvalue problems of the form

$$\begin{bmatrix} K & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \lambda \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix}, \tag{7.2.2}$$

with $n = m + k$, $C \in \mathbb{R}^{m\times k}$ of full rank, stiffness matrix $K \in \mathbb{R}^{m\times m}$, mass matrix $M = M^T \in \mathbb{R}^{m\times m}$, velocity $\mathbf{u} \in \mathbb{C}^m$ and pressure $\mathbf{p} \in \mathbb{C}^k$, that arise in the

linearized stability analysis of steady state solutions of the Navier-Stokes equations [33]. The corresponding ordinary eigenproblem is

$$\begin{bmatrix} S_1 & 0 \\ S_2 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \tilde{\lambda} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix}, \qquad S_1 \in \mathbb{R}^{m\times m}, S_2 \in \mathbb{R}^{k\times m},$$

and, as is also noted in [97], this leads to the reduced problem

$$S_1\mathbf{u} = \tilde{\lambda}\mathbf{u}, \qquad S_1 \in \mathbb{R}^{m\times m}. \tag{7.2.3}$$

If $(\tilde{\lambda}, \mathbf{u})$ is an exact eigenpair of S_1, then for nonzero $\tilde{\lambda}$, $(\tilde{\lambda}, [\mathbf{u}^*, \mathbf{p}^*]^*)$ with $\mathbf{p} = \tilde{\lambda}^{-1} S_2\mathbf{u}$ is an exact eigenpair of S. It can be shown [97, Section 2.3] that $\dim(\mathcal{N}(S_1)) = k$, $\dim(\mathcal{R}(S_1)) = m - k$ and

$$\mathbf{u} \in \mathcal{N}(S_1) \quad \Leftrightarrow \quad \begin{bmatrix} \mathbf{u} \\ 0 \end{bmatrix} \in \mathcal{G}.$$

Hence, by reducing the problem to (7.2.3), the geometric multiplicity of the k eigenvalues $\tilde{\lambda} = 0$ is reduced from 2 to 1.

7.3 Arnoldi methods with purification

In Section 7.3.1, the implicitly restarted B-orthogonal Arnoldi method will be described. In Section 7.3.2, it will be shown how this method can be improved by exploiting the specific structure of the generalized eigenproblem. A new strategy, based on this known but previously not used fact, for the computation of a few leftmost eigenvalues will be presented in Section 7.3.3, followed by numerical examples in Section 7.3.4.

7.3.1 Implicitly restarted B-orthogonal Arnoldi methods

B-orthogonal Arnoldi

The B-orthogonal Arnoldi method is the standard Arnoldi method with the usual inner product $(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$ replaced by the semi-inner product $(\mathbf{x}, \mathbf{y})_B = \mathbf{x}^T B\mathbf{y}$. The B-orthogonal Arnoldi method constructs a B-orthonormal basis $\mathbf{v}_1, \ldots, \mathbf{v}_{k+1}$ for the Krylov subspace

$$\mathcal{K}^{k+1}(S, \mathbf{v}_1) = \operatorname{span}(\mathbf{v}_1, S\mathbf{v}_1, \ldots, S^k\mathbf{v}_1),$$

where $S = (A - \sigma B)^{-1}B$. The basis vectors are related by

$$SV_k = V_k H_k + h_{k+1,k}\mathbf{v}_{k+1}\mathbf{e}_k^T = V_{k+1}\underline{H}_k, \quad V_{k+1}^T B V_{k+1} = I, \tag{7.3.1}$$

where $V_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k] \in \mathbb{R}^{n\times k}$ and $H_k \in \mathbb{R}^{k\times k}$ and $\underline{H}_k = [H_k^T, h_{k+1,k}\mathbf{e}_k]^T \in \mathbb{R}^{(k+1)\times k}$ are upper Hessenberg[1]. Relation (7.3.1) characterizes a k-step Arnoldi

[1] Barred identifiers $\underline{H}_k$ are elements of $\mathbb{R}^{(k+1)\times k}$, whereas $H_k \in \mathbb{R}^{k\times k}$.

factorization. Like in the standard Arnoldi method, approximate eigenpairs $(\theta_i, V_k\mathbf{y}_i)$, called Ritz pairs, can be computed from eigenpairs $(\theta_i, \mathbf{y}_i)$ of H_k.

The usual criterion for convergence of a Ritz pair $(\theta, V_k\mathbf{y})$ with $H_k\mathbf{y} = \theta\mathbf{y}$ is derived from the relation

$$SV_k\mathbf{y} = V_kH_k\mathbf{y} + h_{k+1,k}\mathbf{v}_{k+1}\mathbf{e}_k^T\mathbf{y} = \theta V_k\mathbf{y} + h_{k+1,k}\mathbf{v}_{k+1}\mathbf{e}_k^T\mathbf{y}.$$

If $\|h_{k+1,k}\mathbf{v}_{k+1}\mathbf{e}_k^T\mathbf{y}\|$ is smaller than a given tolerance τ, the Ritz pair $(\theta, V_k\mathbf{y})$ is said to be converged. It follows that if the $\mathbf{v}_i$ are orthonormalized in the 2-norm, it suffices to inspect $|h_{k+1,k}\mathbf{e}_k^T\mathbf{y}|$. Since for B-orthogonal Arnoldi the B-inner product is used, the convergence criterion becomes

$$|h_{k+1,k}\mathbf{e}_k^T\mathbf{y}| \cdot \|\mathbf{v}_{k+1}\|_2 < \tau. \tag{7.3.2}$$

In [97] and [104], the use of the semi-inner product is motivated by the fact that the B inner product is not affected by components of $\mathbf{v}_i$ in the null space of B, $\mathcal{N}(B)$, and hence H_k is independent of components in $\mathcal{N}(S)$. Note, however, that H_k can be corrupted by components in the generalized null space $\mathcal{G}(S)$. Moreover, *because* the B-inner product is not affected by components in the null space of B, there is no reason to assume that components of v_i in the null space of B will not grow; for $\mathbf{z} \in \mathcal{N}(B)$, $\mathbf{x} \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, one has $\|\mathbf{x}\|_B = \|\mathbf{x} + \alpha\mathbf{z}\|_B$. As a consequence, the Ritz vector $V_k\mathbf{y}$ will be spoiled with error components in $\mathcal{N} + \mathcal{G}$ (see also [97, Sect. 4.1] and [104, Sect. 2.3]).

The presence of components in $\mathcal{N} + \mathcal{G}$ in the Arnoldi basis may not only cause spurious eigenvalues and inaccurate Ritz vectors, it may also hamper convergence to the wanted eigenvalues. Purification techniques aim at eliminating the components in $\mathcal{N} + \mathcal{G}$ from the Arnoldi vectors, with the following three goals:

- removal of spurious eigenpair approximations;
- improvement of wanted eigenpair approximations by removing $\mathcal{N} + \mathcal{G}$ components from the Ritz vectors;
- increase of the speed of convergence.

Following [97], the notion of purification can be used in several ways, but the idea boils down to eliminating components in $\mathcal{N} + \mathcal{G}$ of a vector $\mathbf{x}$ by applying S^{j_s} to it, either explicitly or implicitly. In exact arithmetic, the effect is that $S^{j_s}\mathbf{x} \in \mathcal{R}(S^{j_s})$, i.e. $S^{j_s}\mathbf{x}$ is in the wanted eigenspace. See [47] and [104] for the first occurences of the term purification.

Implicitly restarted B-orthogonal Arnoldi with purification

In [97, Sect. 3.2], an implicitly restarted Arnoldi method with B-inner product is proposed. The method, see Alg. 7.1, reduces the corruption of H_k by components in $\mathcal{N}$ and $\mathcal{G}$ significantly (after the implicit restart), and only requires one additional purification step of the Ritz vectors. The result of the implicit restart in step 4,

Algorithm 7.1 Implicitly restarted B-orthogonal Arnoldi with purification

1: Choose an initial vector $\mathbf{v}_1 \leftarrow S^2\mathbf{v}_1$
2: Do $k+1$ steps of B-orthogonal Arnoldi to compute V_{k+2} and $\underline{H}_{k+1}$
3: Compute the QR-factorization $\underline{H}_{k+1} = \underline{Q}_{k+1}R_{k+1}$
4: Implicitly restart: $W_{k+1} = V_{k+2}\underline{Q}_{k+1}, \underline{G}_k = R_{k+1}\underline{Q}_k$
5: Compute eigenpairs $(\theta_i, \mathbf{y}_i)$ of the upper $k \times k$ part of $\underline{G}_k$
6: Purify the Ritz vectors: $\mathbf{x}_i = S(W_k\mathbf{y}_i) = W_{k+1}\underline{G}_k\mathbf{y}_i$
7: The eigen approximations for the generalized problem are $(1/\theta_i + \sigma, \mathbf{x}_i)$

is that the Arnoldi vectors in W_{k+1} and the upper Hessenberg $\underline{G}_k$ are the same as the ones that would have been computed with starting vector $S\mathbf{v}_1/\|S\mathbf{v}_1\|_B$. In other words, the implicit restart removes the $\mathcal{N}$ part from V_{k+2} and the $\mathcal{G}$ part from $\underline{H}_{k+1}$, and it maps the $\mathcal{G}$ part from V_{k+2} to the $\mathcal{N}$ part of W_{k+1}. Note that because of the B-inner product, $\underline{H}_{k+1}$ and $\underline{G}_k$ are free of contributions of components in $\mathcal{N}$. The second purification, in step 6, removes the $\mathcal{N}$ part from the Ritz vector (the $\mathcal{G}$ part was already removed by the implicit restart). The method can still fail due to corruption of $\underline{H}_{k+1}$ by rounding errors, but this can be detected by inspecting $\|R_k^{-1}\|_2$ [97, Thm. 4]: if $\|R_k^{-1}\|_2$ is large and growing for successive values of k, spurious Ritz values may be computed. Secondly, purification of the Ritz vector $W_k\mathbf{y}_i$ may fail if the corresponding Ritz value θ_i is small, i.e. $\theta_i \sim \epsilon\|S\|$.

7.3.2 Exploiting the structure of $A\mathbf{x} = \lambda B\mathbf{x}$

In [97, p. 670], [84, p. 8] and [34, p. 1313] it is concluded that the reduced problem

$$S_1\mathbf{u} = \tilde{\lambda}\mathbf{u}, \qquad S_1 \in \mathbb{R}^{m\times m},$$

see also (7.2.3), is only of theoretical interest, because S_1 and S_2 depend on blocks in A^{-1} which are unlikely to be known. However, matrix vector multiplications with S_1, the only operation with S_1 that is required by the Arnoldi algorithm, and with S_2, can easily be performed by making use of the available multiplication with S. Note that in practical situations also S is not available explicitly and that matrix vector multiplications with S are for instance implemented using the LU-factorization of A.

Theorem 7.3.1. *Let $S \in \mathbb{R}^{n\times n}$ have the block structure*

$$\begin{bmatrix} S_1 & 0 \\ S_2 & 0 \end{bmatrix},$$

with $S_1 \in \mathbb{R}^{m\times m}, S_2 \in \mathbb{R}^{k\times m}$, and let $P = [I_m, 0]^T \in \mathbb{R}^{n\times m}$, $Q = [0, I_k]^T \in \mathbb{R}^{n\times k}$ with $I_m \in \mathbb{R}^{m\times m}$ an identity matrix of dimension m. Then for $\mathbf{x} \in \mathbb{C}^m$,

$$\begin{aligned} S_1\mathbf{x} &= P^T SP\mathbf{x}, \\ S_2\mathbf{x} &= Q^T SP\mathbf{x}. \end{aligned}$$

Proof. The results follow immediately from the identities $S_1 = P^T SP$ and $S_2 = Q^T SP$. □

The operations with P and Q in Theorem 7.3.1 can be performed very efficiently and hence with virtually no additional costs the Arnoldi method can be applied to S_1. This leads to Alg. 7.2, a modification of Alg. 7.1. The Arnoldi basis vectors have length $m < n$, which reduces the costs of orthogonalization (although usually the costs of operations with S are dominant). In step 1, only a single explicit purification of the initial vector is needed. Furthermore, the B-inner product, that was used for its purifying property, and the purification in step 6, are no longer needed, because the implicit restart removes all corruption by components in $\mathcal{N}$ from V_{k+2} and H_{k+1}. On the other hand, to recover the eigenvectors of the original problem, an additional multiplication with S_2 is needed.

Algorithm 7.2 Implicitly restarted Arnoldi for S_1

1: Choose an initial vector $\mathbf{v}_1 \leftarrow S_1\mathbf{v}_1 \in \mathbb{R}^m$
2: Do $k+1$ steps of Arnoldi with S_1 to compute V_{k+2} and $\underline{H}_{k+1}$
3: Compute the QR-factorization $\underline{H}_{k+1} = \underline{Q}_{k+1}R_{k+1}$
4: Implicitly restart: $W_{k+1} = V_{k+2}\underline{Q}_{k+1}, \underline{G}_k = R_{k+1}\underline{Q}_k$
5: Compute eigenpairs $(\theta_i, \mathbf{y}_i)$ of the upper $k \times k$ part of $\underline{G}_k$
6: Compute $\mathbf{p}_i = \theta_i^{-1} S_2\mathbf{x}_i$ with $\mathbf{x}_i = W_k\mathbf{y}_i$
7: The eigen approximations for the generalized problem are $(1/\theta_i + \sigma, [\mathbf{x}_i^*, \mathbf{p}_i^*]^*)$

Improved rounding error analysis

The most important consequence of Theorem 7.3.1, however, is that the results of the error analysis in [97, Section 5] improve considerably. Following the notation and assumptions there, let $P_{\mathcal{R}_1}$ and $P_{\mathcal{N}_1}$ be normalized projectors that map a vector into $\mathcal{R}_1 = \mathcal{R}(S_1)$ and $\mathcal{N}_1 = \mathcal{N}(S_1)$ respectively, so $\mathbf{x} \in \mathbb{C}^m$ can be decomposed uniquely as $\mathbf{x} = P_{\mathcal{R}_1}\mathbf{x} + P_{\mathcal{N}_1}\mathbf{x}$. Note that $P_{\mathcal{N}_1}S_1 = 0$. The computed Arnoldi vectors satisfy

$$\begin{aligned} h_{j+1,j}\mathbf{v}_{j+1} &= S\mathbf{v}_j - \sum_{i=1}^{j} h_{ij}\mathbf{v}_i + \psi_j, \\ h_{ij} &= \mathbf{v}_i^T S\mathbf{v}_j + \delta_{ij}, \\ \mathbf{v}_i^T\mathbf{v}_j &= \begin{cases} 1+\gamma_{ij}, & j = i \\ \gamma_{ij} & j = 1, \ldots, i+1, j \neq i. \end{cases} \end{aligned}$$

In block form, the round-off errors $\|\Psi_{k+1}\|_2$, $\|\Gamma_{k+1}\|_2$ and $\|\Delta_k\|_2$ for the k-step Arnoldi factorization are given by the following relations:

$$V_{k+1}\underline{H}_k = SV_k + \Psi_{k+1}, \tag{7.3.3}$$

$$V_{k+1}^T V_{k+1} = I + \Gamma_{k+1}, \tag{7.3.4}$$

$$\underline{H}_k = V_{k+1}^T SV_k + \Delta_k. \tag{7.3.5}$$

Result 7.3.2. *The $\mathcal{N}_1$ component in $\mathbf{v}_j$ may increase as j increases.*

Proof. Repeating the proof in [97, Section 4.1] leads to

$$\begin{aligned} h_{j,j+1}P_{\mathcal{N}_1}\mathbf{v}_{j+1} &= P_{\mathcal{N}_1}S_1\mathbf{v}_j - \sum_{i=1}^{j} h_{ij}P_{\mathcal{N}_1}\mathbf{v}_i + P_{\mathcal{N}_1}\psi_j \\ &= -\sum_{i=1}^{j} h_{ij}P_{\mathcal{N}_1}\mathbf{v}_i + P_{\mathcal{N}_1}\psi_j. \end{aligned}$$

There is no reason to assume that $\|P_{\mathcal{N}_1}\mathbf{v}_{j+1}\|$ does not increase. □

The improvement over the result in [97, Section 4.1] is that for Arnoldi applied to S, there may be an increase of both components in $\mathcal{N}$ and $\mathcal{G}$, while here there only may be a smaller increase of components in $\mathcal{N}_1$.

The following result shows the improved effect of the implicit purification via $\mathbf{x}_j = S(V_k\mathbf{z}_j) = V_{k+1}\underline{H}_k\mathbf{z}_j$, where $\mathbf{z}_j$ is an eigenvector of H_k. Although this purification is not needed in Alg. 7.2, as will become clear in result 7.3.4 and Theorem 7.3.5, it is included here, however, to show that the relative contributions of the $\mathcal{N}_1$ components are smaller than in the results in [97, Section 4.2].

Result 7.3.3. *The purification operation $\mathbf{x}_j = V_{k+1}\underline{H}_k\mathbf{z}_j$ produces an approximate eigenvector with no $\mathcal{N}_1$ component. This step may fail if $|\theta_j^{-1}| \gg \epsilon_M^{-1}$, where ϵ_M is the machine precision number.*

Proof. From the proof in [97, Section 4.2], it follows that the purified $\mathbf{x}_j$ computed by $\mathbf{x}_j = V_{k+1}\underline{H}_k\mathbf{z}_j$ with $\|\mathbf{z}_j\|_2 = 1$ satisfies

$$\begin{aligned} P_{\mathcal{N}_1}\mathbf{x}_j &= P_{\mathcal{N}_1}S_1V_k\mathbf{z}_j + P_{\mathcal{N}_1}\xi_j \\ &= P_{\mathcal{N}_1}\xi_j, \end{aligned} \tag{7.3.6}$$

with

$$\|\xi_j\|_2 \leq 3k^{3/2}\|V_{k+1}\|_F\|A^{-1}\|_2\epsilon_M + \|\Psi_{k+1}\|_2 + O(\epsilon_M^2).$$

If $\|\mathbf{z}_j\|_2 = 1$ (note that $H_k\mathbf{z}_j = \theta_j\mathbf{z}_j$), then $\|V_k\mathbf{z}_j\|_2 \simeq 1$ and $\|\mathbf{x}_j\|_2 \simeq \theta_j$ and hence relative contributions of the $\mathcal{N}_1$ components in $\mathbf{x}_j$ are obtained by dividing (7.3.6) by θ_j. If θ_j is small, these relative contributions become large and purification may fail. If $|\theta_j^{-1}|\|\xi_j\|_2 \ll 1$, then the $\mathcal{N}_1$ component in $\mathbf{x}_j$ is removed. □

Result 7.3.4. *One implicit restart of Arnoldi produces a $\underline{G}_k$ that is not corrupted by $\mathcal{N}_1$ components, and a W_{k+1} that has no $\mathcal{N}_1$ component. This step may fail if $\|R_{k+1}^{-1}\|_2 \gg \epsilon_M^{-1}$.*

Proof. Repeating the proof in [97, Section 4.4] leads to

$$\|P_{\mathcal{N}_1}W_{k+1}\|_2 \leq \|P_{\mathcal{N}_1}\Xi_{k+1}\|_2 + O(\epsilon_M^2),$$

with

$$\begin{aligned}\|\Xi_{k+1}\|_2 &\leq (k+2)^{3/2}\|V_{k+2}\|_F\epsilon_M \\ &\quad +(\omega\|V_{k+1}\|_2\|A^{-1}\|_2\epsilon_M + \|\Psi_{k+2}\|_2)\|R_{k+1}^{-1}\|_2 + O(\epsilon_M^2),\end{aligned}$$

and $\omega = O(1)$. If $\|R_{k+1}^{-1}\|_2$ is small, W_{k+1} has no significant components in $\mathcal{N}_1$. If $\|R_{k+1}^{-1}\|_2$ is large, then $\|\Xi_k\|_2 \gg \epsilon_M$ and W_{k+1} can have components in $\mathcal{N}_1$. Consequently, G_k is corrupted by the components in $\mathcal{N}_1$ and may cause spurious Ritz values. □

Compared to the results in [97, Section 4.4], the corruption in W_{k+1} and G_k is decreased.

The following theorem shows that, as a Ritz pair $(\tilde{\lambda}, \mathbf{x})$ with not too small $\tilde{\lambda}$ converges to an eigenpair of S_1, it is purified automatically. This is consistent with results 7.3.3 and 7.3.4, and also explains why implicit purification of converged Ritz pairs (step 6 in Alg. 7.1) is not needed.

Theorem 7.3.5. *Let $(\tilde{\lambda}, \mathbf{x})$ be a converged Ritz pair of S_1, with $\mathbf{r} = S_1\mathbf{x} - \tilde{\lambda}\mathbf{x}$ and $\|\mathbf{r}\|_2 < \epsilon$. Then $\|P_{\mathcal{N}_1}\mathbf{x}\|_2 \leq \epsilon/\tilde{\lambda}$.*

Proof. Write $\tilde{\lambda}P_{\mathcal{N}_1}\mathbf{x} = P_{\mathcal{N}_1}S_1\mathbf{x} - P_{\mathcal{N}_1}\mathbf{r}$ and note that $P_{\mathcal{N}_1}S_1\mathbf{x} = 0$. □

Although failure of IRA if $\|R_k^{-1}\|_2$ is large is still possible, the results above show that the (growth of the) corruption by $\mathcal{N}_1$ components is reduced. The rounding errors made during the orthogonalization phase are also reduced, because the standard inner product is used instead of the B-inner product, and hence no additional multiplications with B are needed.

Numerical example

To illustrate the new results, the growth of $\|\Psi_{k+1}\|_2$ (see (7.3.3)) for S and S_1 is compared. Figure 7.1 shows $\|\Psi_{k+1}\|_2$ at every Arnoldi iteration for the example matrix pencil taken from [97, Sect. 3.3]. For S, the B-orthogonal Arnoldi method is used, while for S_1 Arnoldi with the usual inner product is used. For both cases, the initial vector $\mathbf{v}_1$, with all entries equal to one, was purified using $\mathbf{v}_1 \leftarrow S^2\mathbf{v}_1$ and $\mathbf{v}_1 \leftarrow S_1\mathbf{v}_1$ respectively. It is clear that the growth of $\|\Psi_{k+1}\|_2$ is much smaller for S_1. The growth of $\|\Psi_{k+1}\|_2$ for S can be explained as follows: let $\mathbf{w}_k$ be the new Arnoldi vector in iteration k, just after orthogonalization against V_k, but before normalization. The B-inner product neglects any components in $\mathcal{N}(B)$, but these components are normalized with the same factor $h_{k+1,k} = \|\mathbf{w}_k\|_B$. If $h_{k+1,k} < 1$, then these components increase in 2-norm. This may lead to an increase of $\|\Psi_{k+1}\|_2$. Typical values of $h_{k+1,k}$ in this example were of order $O(10^{-4})$. An explanation for the apparent stagnation of the growth of $\|\Psi_{k+1}\|_2$ at some iterations may be: the new Arnoldi vector is computed as $S\mathbf{v}_k$ and $S_1\mathbf{v}_k$ respectively, which is in fact an explicit purification of $\mathbf{v}_k$. Combined with a not too small $h_{k+1,k}$, this will cause only a limited increase.

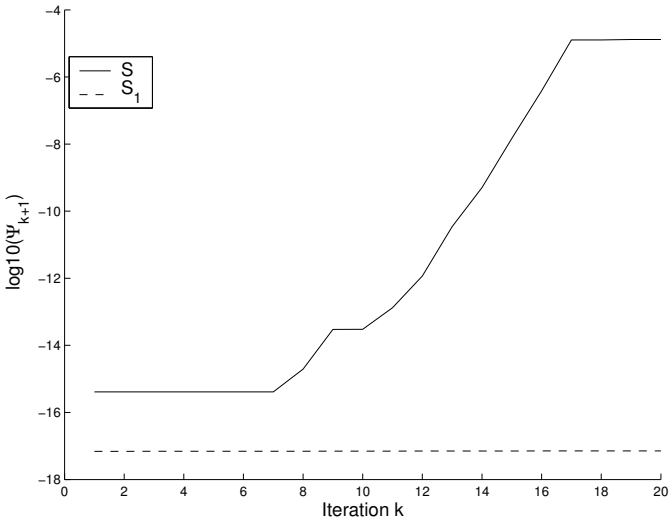

Figure 7.1: The size of $\|\Psi_{k+1}\|_2$ for B-orthogonal Arnoldi applied to $S = (A - 60B)^{-1}B$, and Arnoldi applied to S_1.

A large $\|\Psi_{k+1}\|_2$ may not only prevent the implicit restart with zero shift from purifying the factorization, it also reduces the effect of the implicit purification via $\mathbf{x}_j = V_{k+1}\underline{H}_k\mathbf{z}_j$, as can be deduced from results 7.3.3 and 7.3.4 and their equivalents in [97]. With this in mind, the choice for Arnoldi with S_1 is obvious.

7.3.3 A new strategy

Implicitly restarted Arnoldi with deflation

It is not clear from [97] how the idea of the implicit restart with shift $\sigma_0 = 0$ (Alg. 7.1) is incorporated with the implicitly restarted Arnoldi method with deflation [85, 146]. The IRA method starts with a k-step Arnoldi factorization $SV_k = V_kH_k + h_{k+1,k}\mathbf{v}_{k+1}\mathbf{e}_k^T$. Then, until convergence, the following steps are iterated:

1. Compute the Ritz values θ_i, i.e. the eigenvalues of H_k and split them in a set of wanted Ritz values $\{\theta_1 \dots \theta_j\}$ and unwanted Ritz values $\{\sigma_1 \dots \sigma_p\}$, with $k = j + p$.
2. Apply p QR-steps to H_k with shifts σ_i to remove the unwanted Ritz values.
3. Extend the j-step Arnoldi factorization to a k-step Arnoldi factorization.

Like in Alg. 7.2, the idea now is to implicitly restart with $\sigma_0 = 0$ just before the computation of the Ritz values in step (1), i.e. just after the extension of the Arnoldi factorization. Any detected spurious Ritz values can be removed by

including these as shifts for the implicit restarts. The algorithm is summarized in Alg. 7.3. For details about the implementation of implicit shifts, deflation and the locking procedure, the reader is referred to [85, 146, 147].

Algorithm 7.3 Implicitly restarted Arnoldi for S_1 with purification and deflation

1: Choose an initial vector $\mathbf{v}_1 \leftarrow S_1\mathbf{v}_1$
2: Do $k+1$ steps of Arnoldi to compute V_{k+2} and $\underline{H}_{k+1}$
3: **while** not all converged **do**
4: Purify by applying one restart with $\sigma = 0$: $[V_{k+1}, \underline{H}_k] = \text{purify}(V_{k+2}, \underline{H}_{k+1})$
5: Compute $\lambda(H_k)$ and lock converged wanted Ritz values
6: Select p shifts $\sigma_1, \ldots, \sigma_p$
7: Apply p implicit shifts to compute the $(k-p)$ step Arnoldi factorization $S_1V_{k-p} = V_{k-p+1}\underline{H}_{k-p}$
8: Extend $S_1V_{k-p} = V_{k-p+1}\underline{H}_{k-p}$ to $S_1V_{k+1} = V_{k+2}\underline{H}_{k+1}$
9: **end while**

Exploiting transformations to improve selection and convergence

Besides the shift-and-invert transformation $T_{SI}(A, B, \sigma) = (A - \sigma B)^{-1}$, the generalized Cayley transformation

$$T_C(A, B, \alpha_1, \alpha_2) = (A - \alpha_1 B)^{-1}(A - \alpha_2 B) = B + (\alpha_1 - \alpha_2)T_{SI}, \quad \alpha_1, \alpha_2 \in \mathbb{R} \tag{7.3.7}$$

with $\alpha_1 < \alpha_2$ and $\alpha_1 \neq \lambda_i, i = 1, \ldots, n$, can be used for problems of the form (7.2.2), see [33, 34, 84]. The eigenvalues μ_i of T_C are related to the eigenvalues of (A, B) by the relation $\mu_i = (\lambda_i - \alpha_1)^{-1}(\lambda_i - \alpha_2)$ and the infinite eigenvalues are transformed to 1. Eigenvalues close to α_1 are mapped to eigenvalues far from the unit circle, while eigenvalues close to α_2 are mapped to eigenvalues with small magnitude. The property that is of most use is that eigenvalues with $\text{Re}(\lambda_i) < (\alpha_1 + \alpha_2)/2$ are mapped outside the unit circle, while eigenvalues with $\text{Re}(\lambda_i) > (\alpha_1 + \alpha_2)/2$ are mapped inside the unit circle. The modified Cayley transformation is defined by

$$T_M(A, B, \alpha_1, \alpha_2, \alpha_3) = \begin{bmatrix} K - \alpha_1 M & C \\ C^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} K - \alpha_2 M & \alpha_3 C \\ \alpha_3 C^T & 0 \end{bmatrix}, \tag{7.3.8}$$

and has the same properties as the generalized Cayley transform, except that the infinite eigenvalues are transformed to α_3 [33, 34, 84].

In [84], an algorithm is described where Cayley-transformations are combined with shift-and-invert Arnoldi. The algorithm is based on the hybrid algorithm presented in [33, Section 2.3] and consists of two phases. In the first phase, an r-step Arnoldi factorization is computed using B-orthogonal Arnoldi with purification. The corresponding r Ritz values are used to determine the parameters $\alpha_1, \alpha_2 \in \mathbb{R}$ of the (modified) Cayley transform T_C. In the second phase, implicitly restarted B-orthogonal Arnoldi with purification is applied to T_C to compute the wanted eigenvalues. The parameters α_1, α_2 are updated during the restarts.

The Ritz values that are computed in phase 1 may not have converged (moreover, in [84] there is *no* convergence testing for the Ritz pairs of phase 1, to avoid accepting wrong eigenvalues), and there may be spurious Ritz values as well. Also in the second phase spurious Ritz values may be computed, that make the determination of α_1, α_2 more difficult. In [84] a selection strategy is used to deal with spurious Ritz values. The approach presented here makes such a strategy unnecessary and also reduces the number of different Cayley-transformations needed.

Assume that the $k = 2$ leftmost eigenvalues of (7.2.2) are wanted (complex conjugate pairs counted as one eigenvalue), including any eigenvalues with negative real part. The algorithm is readily adjustable for any number of wanted eigenvalues.

Simply computing the leftmost eigenvalues of $S = T_{SI}(A, 0) = A^{-1}B$ is not advisable for several reasons. Firstly, even if there are eigenvalues with negative real part, the process will most likely be disturbed by spurious Ritz values, as has been explained in the previous sections. Secondly, the leftmost eigenvalues of S do not necessarily correspond to the leftmost eigenvalues of (A, B). The extremal eigenvalues $\tilde{\lambda}_i$ of S, that correspond to the eigenvalues $\lambda_i = 1/\tilde{\lambda}_i$ of (A, B), however, can be computed safely, efficiently and accurately with IRA. These eigenvalues, sorted in increasing real part order, that also not necessarily are the leftmost eigenvalues of (A, B), can be used to compute α_1, α_2 for the modified Cayley transform:

- If $\mathrm{Im}(\lambda_1) = 0$, then $\alpha_1 = \lambda_1 + \frac{\mathrm{Re}(\lambda_2) - \lambda_1}{2}$.
- If $\mathrm{Im}(\lambda_1) \neq 0$, then $\alpha_1 = \lambda_1$.
- In both cases, $\alpha_2 = 2 \times \mathrm{Re}(\lambda_2) - \alpha_1$.

With these choices for α_1, α_2, eigenvalues with $\mathrm{Re}(\lambda_i) < \mathrm{Re}(\lambda_2)$ correspond to eigenvalues $\tilde{\lambda}_i$ of $S_M = T_M(A, B, \alpha_1, \alpha_2, 0)$ with $|\tilde{\lambda}_i| > 1$, while eigenvalues with $\mathrm{Re}(\lambda_i) > \mathrm{Re}(\lambda_2)$ are transformed inside the unit circle. Hence also any missed eigenvalues between λ_1 and λ_2 correspond to eigenvalues μ_i of S_M with $|\mu_i| > 1$. The eigenvalues of S_M with largest magnitude can again be computed by IRA and because the infinite eigenvalues are transformed to $\alpha_3 = 0$, there is virtually no danger that spurious Ritz values will be selected as wanted eigenvalue. As soon as eigenvalues inside the unit circle are computed, it can be safely concluded that the leftmost eigenvalues (including the eigenvalues with negative real part) are found. The strategy is shown in Alg. 7.4.

The strategy consists of two phases: in phase 1 (step 1-2), the largest eigenvalues (in magnitude) of S are computed. Phase 2 (step 3-6) checks for any missed eigenvalues using the Cayley transformation. In step 1, a larger number r will increase the chance of computing the leftmost eigenvalue already in this phase. In step 4, one could also take $\alpha_1 = 0$, but because any missed eigenvalues are expected to be close to λ_1, this is not preferred. Additional verification of any missed eigenvalues can be done by choosing new α_1, α_2 based on the eigenvalues found in step 7 to compute the largest eigenvalues of the new S_M, or by using techniques described in [98].

Algorithm 7.4 Strategy for computing the 2 left most eigenvalues of (A, B)

PHASE 1

1: Compute the $r \geq 2$ largest eigenvalues $\tilde{\lambda}_i$ of S_1 with Alg. 7.3
2: Order $\lambda_i = 1/\tilde{\lambda}_i$, $i = 1, \ldots, r$ by increasing real part

PHASE 2

3: Determine α_1 and α_2, and $\alpha_3 = 0$
4: $S_M = T_M(A, B, \alpha_1, \alpha_2, \alpha_3)$
5: Compute the largest 2 eigenvalues μ_i of S_{M1} with Alg. 7.3
6: The eigenvalues of (A, B) are $\lambda_i = \frac{\alpha_1\mu_i - \alpha_2}{\mu_i - 1}$

The difference with existing approaches is that in the determination of α_1, α_2 rather accurate eigenvalue approximations are used, with as advantages that fewer updates of S_M are needed and that the risk of missing eigenvalues is reduced. Furthermore, the possible disturbance by spurious Ritz values is reduced by first computing only the largest eigenvalues of S_1. Note that with the choice $\alpha_3 = 0$, S_M can be reduced to S_{M1} in the same way as S to S_1, as described in Section 7.3.2. If $(\mu, \mathbf{u})$ is an exact eigenpair of S_{M1}, then for nonzero μ, $(\mu, [\mathbf{u}^*, \mathbf{p}^*]^*)$ with $\mathbf{p} = \mu^{-1} S_{M2}\mathbf{u}$ is an exact eigenpair of S_M. If $(\mu, [\mathbf{u}^*, \mathbf{p}^*]^*)$ is an eigenpair of the modified eigenvalue problem (7.3.8), then $(\frac{\alpha_1\mu_i - \alpha_2}{\mu_i - 1}, [\mathbf{u}^*, \mathbf{q}^*]^*)$, with $\mathbf{q} = (\mu - \alpha_3)/(\mu - 1)\mathbf{p}$, is an eigenpair of the original generalized eigenvalue problem (7.2.2), provided $\mu \notin \{1, \alpha_3\}$ (see [61, 84]).

It may seem that there is no advantage in using $S_M = T_M(\alpha_1, \alpha_2, 0)$ instead of S, since in exact arithmetic, due to shift-invariance of Krylov subspaces, Arnoldi for S and S_M produces the same eigenvalue estimates of (A, B) [98, Lemma 2.5]. However, when using Arnoldi for S, the spurious Ritz values may be hard to distinguish from wanted leftmost Ritz values, as both may be close to zero, while when using Arnoldi for S_M, the spurious Ritz values (near zero) are clearly separated from the wanted Ritz values (magnitude larger than 1).

7.3.4 Realistic examples

The strategy in Alg. 7.4 is applied to two large-scale examples. The first example is the stability analysis of the flow over a backward facing step, a well known benchmark problem from fluid dynamics [69]. The second examples is the flow in a driven cavity [43, Section 7.1.3]. When referring to (finite) eigenvalues λ_i, it is assumed that the λ_i are sorted in increasing real part order, i.e. λ_1 is the leftmost eigenvalue. For more information about the bifurcation analysis of such nonlinear systems, see [35].

The method `eigs` of Matlab 6.5, which is a wrapper around ARPACK [86], is used in all experiments. The stopping criterion is $\tau = 10^{-6}$ and the size of the Arnoldi factorization is $k = 20$.

Table 7.1: Statistics for Alg. 7.4 for the flow over a backward facing step with Reynolds number $Re = 800$ (Section 7.3.4): number of restarts, time, found eigenvalues and residuals after each phase.

	reduced		unreduced	
	phase 1	phase 2	phase 1	phase 2
#restarts	3	2	3	3
time (s)	118	95	120	97
eigenvalues	$\lambda_1, \lambda_{2,3}$	$\lambda_1, \lambda_{2,3}$	$\lambda_1, \lambda_{2,3}$	$\lambda_1, \lambda_{2,3}$
$\max_i \|A\mathbf{x}_i - \lambda_i B\mathbf{x}_i\|$	$1 \cdot 10^{-12}$	$1 \cdot 10^{-12}$	$9 \cdot 10^{-11}$	$9 \cdot 10^{-11}$

Table 7.2: Statistics for Alg. 7.4 for the driven cavity with Reynolds number $Re = 500$ (Section 7.3.4): number of restarts, time, found eigenvalues and residuals after each phase.

	$r = 2$		$r = 5$	
	phase 1	phase 2	phase 1	phase 2
#restarts	1	4	6	4
time (s)	33	112	106	112
eigenvalues	λ_1, λ_4	$\lambda_1, \lambda_{2,3}$	λ_1, λ_{4-7}	$\lambda_1, \lambda_{2,3}$
$\max_i \|A\mathbf{x}_i - \lambda_i B\mathbf{x}_i\|$	$1 \cdot 10^{-16}$	$1 \cdot 10^{-14}$	$1 \cdot 10^{-10}$	$1 \cdot 10^{-14}$

Flow over a backward facing step

The matrices A and B with $n = m + p = 21,730 + 7,872 = 29,602$, were obtained using the package IFISS [138]. The Reynolds number was $Re = 800$ (see [43, p. 315]). Table 7.1 shows statistics for Alg. 7.4 with $r = 2$, both for S and the reduced problem S_1. The leftmost eigenvalues $\lambda_1 = 6.04 \cdot 10^{-2}$ and $\lambda_{2,3} = 7.97 \cdot 10^{-2} \pm i1.92 \cdot 10^{-2}$ were already found in the first phase of the algorithm: the validation in phase 2 did not result in new eigenvalues. Although the running times for both the reduced and the unreduced problem are equal, as expected, the residuals are better for the reduced problem. The claim in [69], that the steady state flow at a Reynolds number $Re = 800$ is stable, is confirmed by the results.

Flow in a driven cavity

The matrices A and B with $n = m + p = 8,450 + 1,089 = 9,539$, for Reynolds number $Re = 500$, were obtained using the package IFISS [138]. Table 7.2 shows statistics for Alg. 7.4 with $r = 2$ and $r = 5$, for the reduced problem S_1. The eigenvalues $\lambda_1 = 3.21 \cdot 10^{-2}$ and $\lambda_4 = 1.01 \cdot 10^{-1}$ were found in the first phase of the algorithm. The validation in phase 2 identified the missed eigenvalue pair $\lambda_{2,3} = 6.20 \cdot 10^{-2} \pm i4.61 \cdot 10^{-1}$. Increasing r does not help finding the missed eigenvalue in phase 1, while it increases the running time.

7.4 Jacobi-Davidson methods, preconditioning and purification

If the linear system solves with $A - \sigma B$ are inaccurate or not computable within reasonable time, the strategy based on the implicitly restarted Arnoldi method is no longer applicable, although an inexact variant could be considered [96]. Here a scheme based on the Jacobi-Davidson QZ method [51] is proposed, that does not require exact solves with $(A - \sigma B)$.

The Jacobi-Davidson method [140] combines two principles to compute eigenpairs of eigenvalue problems $A\mathbf{x} = \lambda\mathbf{x}$. The first principle is to apply a Ritz-Galerkin approach with respect to a subspace spanned by $\mathbf{v}_1, \ldots, \mathbf{v}_k$, the search space. The second principle is the computation of a correction orthogonal the current eigenvector approximation. The Jacobi-Davidson method for generalized eigenvalue problems will be briefly explained in Sections 7.4.1 and 7.4.2. For a more detailed description, the reader is referred to [51, 139, 140].

In Section 7.4.3, it will be shown that when an exact preconditioner is used to solve the correction equation, purification is obtained automatically. In Section 7.4.4, this fact will be combined with other properties of Jacobi-Davidson to obtain an efficient method for the computation of a few selected eigenvalues.

7.4.1 The Jacobi-Davidson method for generalized eigenproblems

Given the generalized eigenvalue problem

$$A\mathbf{x} = \lambda B\mathbf{x}, \qquad \mathbf{x} \neq 0,$$

with $A, B \in \mathbb{R}^{n\times n}$, the Jacobi-Davidson method applies a Petrov-Galerkin condition to compute approximate eigenpairs. If the search space is spanned by $\mathbf{v}_1, \ldots, \mathbf{v}_k$, with $V_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$ orthogonal, and the test space is spanned by $\mathbf{w}_1, \ldots, \mathbf{w}_k$, with $W_k = [\mathbf{w}_1, \ldots, \mathbf{w}_k]$ orthogonal, the Petrov-Galerkin condition becomes

$$AV_k\mathbf{s} - \theta BV_k\mathbf{s} \perp \{\mathbf{w}_1, \ldots, \mathbf{w}_k\}.$$

This leads to the reduced $k \times k$ system

$$W_k^* AV_k\mathbf{s} = \theta W_k^* BV_k\mathbf{s},$$

which can be solved using full space methods like QZ to compute eigenpair approximations $(\theta_i, \mathbf{q}_i = V_k\mathbf{s}_i)$ of (7.4.1).

Given such an eigenpair approximation $(\theta_i, \mathbf{q}_i)$, the question is how to expand the search and test space to improve the approximation. With the corresponding residual vector given by

$$\mathbf{r}_i = (A\mathbf{q}_i - \theta_i B\mathbf{q}_i),$$

the Jacobi-Davidson method computes a correction $\mathbf{t} \perp \mathbf{q}_i$ from the Jacobi-Davidson correction equation

$$(I - \mathbf{z}_i\mathbf{z}_i^*)(A - \theta_i B)(I - \mathbf{q}_i\mathbf{q}_i^*)\mathbf{t} = -\mathbf{r}_i, \tag{7.4.1}$$

where the test vector $\mathbf{z}_i = \mu A\mathbf{q}_i + \nu B\mathbf{q}_i$ for a suitable pair $\mu, \nu \in \mathbb{C}$. The search space is expanded with $\mathbf{t}$ and the test space is expanded with $\mu A\mathbf{t} + \nu B\mathbf{t}$. A Ritz pair is accepted if $\|\mathbf{r}_i\|_2 = \|(A\mathbf{q}_i - \theta_i B\mathbf{q}_i)\|_2$ is smaller than a given tolerance.

7.4.2 Jacobi-Davidson QZ

In [51, 139], the Jacobi-Davidson method is extended with deflation. The Jacobi-Davidson QZ (JDQZ) method computes a partial generalized Schur form of the pencil (A, B). Let the current approximate partial generalized Schur form be given by

$$AQ_k = Z_k S_k, \qquad BQ_k = Z_k T_k,$$

with Q_k, Z_k $n \times k$ matrices and S_k, T_k upper triangular $k \times k$ matrices. The problem of finding the next Schur triple $(\mathbf{q}_{k+1}, \mathbf{z}_{k+1}, (\alpha_{k+1}, \beta_{k+1}))$ with $\theta_{k+1} = \alpha_{k+1}/\beta_{k+1}$ can be rewritten as a deflated generalized eigenvalue problem

$$Q_k^*\mathbf{q}_{k+1} = 0, \qquad (I - Z_kZ_k^*)(\beta_{k+1}A - \alpha_{k+1}B)(I - Q_kQ_k^*)\mathbf{q}_{k+1} = 0, \tag{7.4.2}$$

which can be solved by the Jacobi-Davidson method. With the search space represented by the orthogonal matrix V and the test space by the orthogonal matrix W, so that $V^*Q_k = W^*Z_k = 0$, the reduced system matrices become

$$\begin{aligned} M_A &\equiv W^*(I - Z_kZ_k^*)A(I - Q_kQ_k^*) = W^*AV, \\ M_B &\equiv W^*(I - Z_kZ_k^*)B(I - Q_kQ_k^*) = W^*BV. \end{aligned}$$

The generalized Schur decomposition of (M_A, M_B) is computed using QZ:

$$Z_M^*M_AQ_M = S_A, \qquad Z_M^*BQ_M = S_B.$$

The generalized Schur form is ordered with respect to the target τ, and an approximate Petrov triple for (7.4.2) is obtained as

$$(\tilde{\mathbf{q}}, \tilde{\mathbf{z}}, (\tilde{\alpha}, \tilde{\beta})) = (VQ_M\mathbf{e}_1, WZ_M\mathbf{e}_1, (s_{A,11}, s_{B,11})).$$

Given a Petrov triple $(\tilde{q}, \tilde{z}, (\tilde{\alpha}, \tilde{\beta}))$ for the deflated problem, the corresponding generalized deflated correction equation becomes

$$(I - \tilde{\mathbf{z}}\tilde{\mathbf{z}}^*)(I - Z_kZ_k^*)(\tilde{\beta}A - \tilde{\alpha}B)(I - Q_kQ_k^*)(I - \tilde{\mathbf{q}}\tilde{\mathbf{q}}^*)\mathbf{t} = -\tilde{\mathbf{r}}_i, \tag{7.4.3}$$

where the residual $\tilde{\mathbf{r}}$ is

$$\tilde{\mathbf{r}} = (I - Z_kZ_k^*)(\tilde{\beta}A - \tilde{\alpha}B)(I - Q_kQ_k^*)\tilde{\mathbf{q}},$$

and $Q_k^*\mathbf{t} = Z_k^*\tilde{\mathbf{z}} = Q_k^*\tilde{\mathbf{q}} = 0$, $\tilde{\mathbf{q}}^*\mathbf{t} = 0$, $\|t\|_2 = 1$. The search space is expanded with the orthogonal complement of t, and the test space is orthogonally expanded with $(I - Z_kZ_k^*)(\mu A + \nu B)(I - Q_kQ_k^*)\mathbf{t}$.

If the correction equation is solved exactly, the Jacobi-Davidson method converges asymptotically quadratically. In fact, the method can be shown to be a Newton scheme. Solving the correction equation exactly may be too expensive in practice and therefore Krylov subspace methods with preconditioning are used to solve the correction equation approximately. With a preconditioner $K \approx A - \tau B$, the correction equation (7.4.3) can be preconditioned by

$$(I - \tilde{\mathbf{z}}\tilde{\mathbf{z}}^*)(I - Z_kZ_k^*)K(I - Q_kQ_k^*)(I - \tilde{\mathbf{q}}\tilde{\mathbf{q}}^*).$$

With $Q_k := [Q_k, \tilde{\mathbf{q}}]$, $Z_k := [Z_k, \tilde{\mathbf{z}}]$, $Y_k = K^{-1}Z_k$ and $H_k = Q_k^*Z_k$, the left preconditioned correction equation becomes

$$(I - Y_kH_k^{-1}Q_k^*)K^{-1}(\beta A - \alpha B)(I - Y_kH_k^{-1}Q_k^*)\mathbf{t} = -\mathbf{r}, \tag{7.4.4}$$

where $\mathbf{r} = (I - Y_kH_k^{-1}Q_k^*)K^{-1}\tilde{\mathbf{r}}$. In the appendix (Section 7.6) it is described how for certain types of ILU preconditioners the process of solving the correction equation (7.4.3) can be implemented efficiently.

7.4.3 Purification

Jacobi-Davidson style methods select a new Petrov pair according to some criterion, for instance the leftmost Petrov pair, at every iteration. In the absence of infinite eigenvalues, selecting the leftmost Petrov pair will usually result in convergence to the leftmost eigenvalue, assuming that the initial search space contains components in that direction. In the presence of infinite eigenvalues however, this will no longer be a smart strategy: Petrov values will go to infinity, without a proper mechanism to identify them as infinite eigenvalue approximations.

If the search space is restricted to $\mathcal{R}(S^j)$, approximations to infinite eigenvalues can be avoided. Projection of the search space vectors onto $\mathcal{R}(S^j)$ is not attractive because an orthogonal basis for $\mathcal{R}(S^j)$ is not cheaply available. The following lemmas are needed for proving Theorem 7.4.6, which states that if an exact preconditioner[2] is used for the correction equation and if the initial search space $V_0 \subset \mathcal{R}(S^j)$, then, in exact arithmetic, no spurious eigenvalues are computed during the Jacobi-Davidson process.

Lemma 7.4.1. *Let* $\mathbf{q} = ((A - \sigma B)^{-1}B)^j\mathbf{x} \in \mathcal{R}(S^j)$ *and* $K = A - \tau_0 B$. *Then* $\mathbf{r} = (\beta A - \alpha B)\mathbf{q} \in \mathcal{R}(BS^{j-1})$.

Proof. The result follows from some linear algebra:

$$\begin{aligned}(\beta A - \alpha B)\mathbf{q} &= \beta((A - \sigma B) + (\sigma - \alpha/\beta)B)\mathbf{q} \\ &= \beta((\sigma - \alpha/\beta)B\mathbf{q} + (A - \sigma B)((A - \sigma B)^{-1}B)^j)\mathbf{x} \\ &= \beta B((\sigma - \alpha/\beta)\mathbf{q} + ((A - \sigma B)^{-1}B)^{j-1})\mathbf{x} \in \mathcal{R}(BS^{j-1}),\end{aligned}$$

[2] An exact preconditioner is a preconditioner $K = A - \tau_0 B$ for which linear systems of the form $Kx = y$ can be solved exactly, for instance by using an exact LU-factorization $LU = K$.

where in the last step $\mathcal{R}(BS^j) \subseteq \mathcal{R}(BS^{j-1})$ is used. □

Lemma 7.4.2. *Let* $\mathbf{y} = BS^{j-1}\mathbf{x} \in \mathcal{R}(BS^{j-1})$ *and* $K = A - \tau_0 B$. *Then* $K^{-1}\mathbf{y} \in \mathcal{R}(S^j)$.

Proof. With basic linear algebra, one finds

$$\begin{aligned} K^{-1}\mathbf{y} &= (A-\tau_0 B)^{-1}\mathbf{y} \\ &= (A-\tau_0 B)^{-1}BS^{j-1}\mathbf{x} \\ &= (A-\sigma B)^{-1}(I+(\tau_0-\sigma)B(A-\tau_0 B)^{-1})BS^{j-1}\mathbf{x} \in \mathcal{R}(S^j). \end{aligned}$$

□

Lemma 7.4.3. *Let* $\mathbf{r} \in \mathcal{R}(S^j)$, $K = A - \tau_0 B$, $AQ_k = Z_k S_A$, $BQ_k = Z_k S_B$, $\mathbf{q} \in \mathcal{R}(S^j)$, $\mathbf{z} = \nu A\mathbf{q} + \mu B\mathbf{q}$, $Y_k = K^{-1}[Z_k, \mathbf{z}]$ *and* $H_k = [Q_k, \mathbf{q}]^* Z_k$. *Then* $(I - Y_k H_k^{-1}[Q_k, \mathbf{q}]^*)\mathbf{r} \in \mathcal{R}(S)$.

Proof. First note that $\mathcal{R}(Z_k) = \mathcal{R}(AQ_k) = \mathcal{R}(BQ_k)$. It follows from Lemma 7.4.1 that $\mathbf{z} \in \mathcal{R}(BS^{j-1})$ and hence $\mathcal{R}(K^{-1}[Z_k, \mathbf{z}]) \subseteq \mathcal{R}(S^j)$. Consequently, $(I - Y_k H_k^{-1}[Q_k, \mathbf{q}]^*)\mathbf{r} \in \mathcal{R}(S^j)$. □

Lemma 7.4.4. *Let* $\mathbf{r} \in \mathcal{R}(S^j)$, $K = A - \tau_0 B$, $AQ_k = Z_k S_A$, $BQ_k = Z_k S_B$, $\mathbf{q} \in \mathcal{R}(S^j)$, $\mathbf{z} = \nu A\mathbf{q} + \mu B\mathbf{q}$, $Y_k = K^{-1}[Z_k, \mathbf{z}]$ *and* $H_k = [Q_k, \mathbf{q}]^* Z_k$. *Then*

$$\mathcal{K}^j((I - Y_k H_k^{-1} Q_k^*)K^{-1}(\beta A - \alpha B)(I - Y_k H_k^{-1} Q_k^*), \mathbf{r}) \subseteq \mathcal{R}(S^j).$$

Proof. The result follows from applying subsequently Lemma 7.4.3, 7.4.1, 7.4.2 and again 7.4.3. □

This lemma not only enables one to use a Krylov solver for the correction equation, it also has consequences for purification in Jacobi-Davidson.

Lemma 7.4.5. *If the initial search space* $V_0 \subset \mathcal{R}(S^j)$ *and the Jacobi-Davidson correction equation is solved using an exact preconditioner, then all subsequent search spaces* $V_k \subset \mathcal{R}(S^j)$.

Proof. The result follows from Lemma 7.4.4. □

Theorem 7.4.6. *If the initial search space* $V_0 \subset \mathcal{R}(S^j)$ *and the Jacobi-Davidson correction equation is solved using an exact preconditioner, then in exact arithmetic no spurious eigenpairs are computed during the Jacobi-Davidson process.*

Proof. The reduced system is $(M_A, M_B) = (W^*AV, W^*BV)$ with test space $W = \nu AV + \mu BV$. Applying Lemma 7.4.1 to W gives $W \subset \mathcal{R}(BS^{j-1})$ and no spurious eigenvalues are computed. From Lemma 7.4.5 it follows that the Petrov vectors $\mathbf{q}_i = V_k \mathbf{s}_i$ satisfy $\mathbf{q}_i \in \mathcal{R}(S^j)$. □

The last theorem says that, in exact arithmetic, if the Jacobi-Davidson method with exact preconditioning starts with $V_0 \subset \mathcal{R}(S^j)$, then $V_k \subset \mathcal{R}(S^j)$ and $W \subset \mathcal{R}(BS^{j-1})$ and no spurious eigenpairs are computed. In other words, with exact preconditioning the search space is purified automatically. The effect is even enforced because usually more than one iteration of the Krylov solver is needed.

However, in finite arithmetic components in $\mathcal{N}+\mathcal{G}$ may still arise due to rounding errors, and if an exact preconditioner is not available, Theorem 7.4.6 is also not applicable. Fortunately, there is a result similar to Theorem 7.3.5. Let $P_{\mathcal{R}}$, $P_{\mathcal{N}}$ and $P_{\mathcal{G}}$ be normalized projectors that map a vector into $\mathcal{R} = \mathcal{R}(A^{-1}B)$, $\mathcal{N} = \mathcal{N}(A^{-1}B)$ and $\mathcal{G} = \mathcal{G}(A^{-1}B)$ respectively, so $\mathbf{x} \in \mathbb{C}^m$ can be decomposed uniquely as $\mathbf{x} = P_{\mathcal{R}}\mathbf{x} + P_{\mathcal{N}}\mathbf{x} + P_{\mathcal{G}}\mathbf{x}$. The following theorem shows that a converged Petrov pair $(\lambda, \mathbf{x})$ is purified automatically, provided $|\lambda|$, $\|A^{-1}\|_2$ and $\|B\|_2$ are not too large.

Theorem 7.4.7. *Let $(\lambda, \mathbf{x})$ be a converged Petrov pair of (A, B), with $\mathbf{r} = A\mathbf{x} - \lambda B\mathbf{x}$ and $\|\mathbf{r}\|_2 < \epsilon$. Then*

$$\begin{aligned} \|P_{\mathcal{N}}\mathbf{x}\|_2 &\leq \epsilon\|A^{-1}\|_2(1 + |\lambda|\|A^{-1}\|_2\|B\|_2), \\ \|P_{\mathcal{G}}\mathbf{x}\|_2 &\leq \epsilon\|A^{-1}\|_2. \end{aligned}$$

Proof. Use $\mathbf{x} = A^{-1}(\mathbf{r} + \lambda B\mathbf{x})$, $P_{\mathcal{N}}(A^{-1}B) = (A^{-1}B)P_{\mathcal{G}}$ and $P_{\mathcal{G}}(A^{-1})B = 0$. □

7.4.4 Harmonic Ritz-values, exact targets and purification

Numerical experiments show that the JDQZ process with harmonic Petrov values and a target equal to an eigenvalue does not converge to this eigenvalue, but to eigenvalues closest to the target. This observation can be understood from a theoretical point of view, as will be explained next, and may be of use in avoiding convergence to eigenvalues at infinity.

First consider the Jacobi-Davidson process for the ordinary eigenproblem

$$A\mathbf{x} = \lambda\mathbf{x}$$

In [140] it is shown that the harmonic Ritz values of A are equal to the eigenvalues of the $k \times k$ matrix

$$\tilde{H}_k = (W_k^*V_k)^{-1}W_k^*AV_k = (W_k^*V_k)^{-1}$$

with $W_k = AV_k$ and $W_k^*W_k = I$. Note that $\tilde{H}_k^{-1} = W_k^*V_k = W_k^*A^{-1}W_k$, the projection of A^{-1} with respect to an orthonormal basis W_k. In practice it is not necessary to invert $\tilde{H}_k^{-1}$, because the harmonic Ritz values of A are the reciprocals of the eigenvalues of $\tilde{H}_k = W_k^*V_k$. Hence, no problems are encountered if $W_k^*V_k$ is singular, which may happen if A has an eigenvalue at zero.

Theorem 7.4.8. *Let $A \in \mathbb{R}^{n\times n}$ be a normal matrix and $\tau \in \mathbb{R}$. If τ exactly equals an eigenvalue of A, then, in exact arithmetic, the Jacobi-Davidson process with harmonic Ritz values and target τ will not converge to the eigenvalue $\lambda = \tau$.*

Proof. Without loss of generality, let $\tau = \lambda = 0$: if $\tau = \lambda \neq 0$, the proof follows for $A - \tau I$. Denote the null space of A by $\mathcal{N}$ and the range of A by $\mathcal{R}$. The eigenspace corresponding to the eigenvalue $\lambda = 0$ is (a subset of) the null space $\mathcal{N}$. However, because of the normality of A, the space spanned by the columns of $W_k = AV_k$ does not contain any elements of the eigenspace of $\lambda = 0$. Because the eigenspaces of A and A^{-1} are the same, the proof would be complete if A^{-1} would exist. However, in this case there is an eigenvalue $\lambda = 0$ and hence A^{-1} does not exist.

If $\delta \in \mathbb{C} \backslash \Lambda(A)$, then $(A + \delta I)^{-1}$ exists and the eigenspaces of A, $A + \delta I$, and $(A + \delta I)^{-1}$ are the same, but the eigenvalues are λ, $\lambda + \delta$, and $(\lambda + \delta)^{-1}$ respectively. With the iteration vectors w_k still generated by Av_k, and hence W_k containing no components of the null space of A and the eigenspace V_δ of $A + \delta I$, it follows that the eigenvalue δ^{-1} will not be contained in the set of eigenvalues of $W_k^*(A + \delta I)^{-1} W_k$. □

In finite arithmetic, W_k can still contain components of the (generalized) null space of A, which may hamper convergence to the desired eigenvalues or even cause convergence to the undesired, perturbed eigenvalue. If the starting vector is in the null space of A, Jacobi-Davidson with harmonic Ritz values will break down.

The proof for the Jacobi-Davidson QZ process for generalized eigenproblems is similar. In [141] these observations are used to derive a selection strategy for Ritz pairs.

7.4.5 A strategy with JDQZ

The strategy in Alg. 7.5 is conceptually the same as Alg. 7.4, with JDQZ instead of IRA. By considering the pencil (B, A) instead of (A, B), the infinite eigenvalues are transformed to zero. The extremal eigenvalues of (B, A) can be computed safely, efficiently and accurately by JDQZ with harmonic Petrov values and target $\tau = 0$ (see Theorem 7.4.8). These eigenvalues can be used to determine α_1, α_2 and $\alpha_3 = 0$ for the modified Cayley transform (see also Section 7.3.3), here formulated as the generalized eigenvalue problem $\mathcal{A}(\alpha_2, \alpha_3)\mathbf{x} = \mu \mathcal{B}(\alpha_1)\mathbf{x}$ with

$$\mathcal{A}(\alpha_2, \alpha_3) = \begin{bmatrix} K - \alpha_2 M & \alpha_3 C \\ \alpha_3 C^T & 0 \end{bmatrix}, \quad \mathcal{B}(\alpha_1) = \begin{bmatrix} K - \alpha_1 M & C \\ C^T & 0 \end{bmatrix}.$$

Eigenpairs that are found in phase 1 (step 1 - 3) can be deflated from the problem in phase 2 (step 4 - 6).

7.4.6 Realistic examples

The strategy in Alg. 7.5 is applied to the test problems of Section 7.3.4. To make a fair comparison with the IRA strategy in Alg. 7.4, two situations were considered:

- The correction equation is not solved exactly, but with 20 steps of (unrestarted) GMRES [133] with preconditioner A.

Algorithm 7.5 Strategy for computing the 2 left most eigenvalues of (A, B)

PHASE 1

1: Choose a suitable preconditioner for the correction equation
2: Compute the $r \geq 2$ largest eigenvalues $\tilde{\lambda}_i$ of (B, A) with JDQZ ($\tau = 0$)
3: Order $\lambda_i = 1/\tilde{\lambda}_i$, $i = 1, \ldots, r$ by increasing real part

PHASE 2

4: Determine α_1 and α_2, and $\alpha_3 = 0$
5: Compute the largest 2 eigenvalues μ_i of $(\mathcal{A}(\alpha_2, \alpha_3), \mathcal{B}(\alpha_1))$ with JDQZ
6: The eigenvalues of (A, B) are $\lambda_i = \frac{\alpha_1 \mu_i - \alpha_2}{\mu_i - 1}$

- The correction equation is solved exactly and an initial search space of size $j_{\min}$ is computed with Arnoldi.

In both situations, the initial vector $\mathbf{v}_1$ had all entries one and was not purified. The search and test space dimensions are limited by $j_{\min} = 15$ and $j_{\max} = 20$, and the residual tolerance was 10^{-6} (see [51] for more details about the several parameters and sophisticated stopping criteria). In situation 1, solves with preconditioner A are needed and hence one could argue that in that case also the IRA strategy in Alg. 7.4 could be used. The goal here however is to show that even if the correction equation is not solved exactly and the initial search space is not constructed with Arnoldi, JDQZ is able to compute the leftmost eigenvalues. In this way, situation 1 resembles the situation where indeed solves with A are not possible and the IRA strategy is not applicable. The quality of the preconditioner influences the speed of convergence of the GMRES process, but this chapter is not concerned with designing a good preconditioner (see [43, Chapter 8] and references therein for preconditioners of related systems). For the experiments, a variant of the JDQZ algorithm, that keeps the search and test spaces real, is used (RJDQZ [158]).

Flow over a backward facing step

Table 7.3 shows statistics for Alg. 7.5 with $r = 2$, for both exact and inexact solution of the correction equation. The leftmost eigenvalues $\lambda_1 = 6.04 \cdot 10^{-2}$ and $\lambda_{2,3} = 7.97 \cdot 10^{-2} \pm i1.92 \cdot 10^{-2}$ were already found in the first phase of the algorithm: the validation in phase 2 did not result in new eigenvalues. The differences in residual norms can be explained by the asymptotically quadratical convergence of the exact variant. Concerning the higher computing times for the inexact variant, one should keep in mind that for stability analysis the quality of the solution (no missed eigenvalues) is the most important. Furthermore, it may be expected that the times can be decreased by using a more effective preconditioning, but this goes beyond the scope of this chapter. The exact variant is faster than implicitly restarted Arnoldi (cf. Table 7.2).

Table 7.3: Statistics for Alg. 7.5 for the backward facing step with Reynolds number $Re = 800$ (Section 7.4.6): number of iterations, restarts, time, found eigenvalues and residuals after each phase. No restarts were needed.

	Inexact		Exact	
	phase 1	phase 2	phase 1	phase 2
#iterations	18	9	4	2
time (s)	1400	1000	85	80
eigenvalues	$\lambda_1, \lambda_{2,3}$	λ_1	$\lambda_1, \lambda_{2,3}$	$\lambda_1, \lambda_{2,3}$
$\max_i \|A\mathbf{x}_i - \lambda_i B\mathbf{x}_i\|$	$1 \cdot 10^{-11}$	$1 \cdot 10^{-11}$	$1 \cdot 10^{-15}$	$1 \cdot 10^{-15}$

Table 7.4: Statistics for Alg. 7.4 for the driven cavity with Reynolds number $Re = 500$ (Section 7.4.6): number of restarts, time, found eigenvalues and residuals after each phase. No restarts were needed.

	Inexact		Exact	
	phase 1	phase 2	phase 1	phase 2
#iterations	13	12	3	3
time (s)	330	340	21	58
eigenvalues	λ_1, λ_4	λ_1,$\lambda_{2,3}$	λ_1, λ_4	λ_1,$\lambda_{2,3}$
$\max_i \|A\mathbf{x}_i - \lambda_i B\mathbf{x}_i\|$	$1 \cdot 10^{-10}$	$1 \cdot 10^{-10}$	$1 \cdot 10^{-15}$	$1 \cdot 10^{-15}$

Flow in a driven cavity

Table 7.4 shows statistics for Alg. 7.5 with $r = 2$, for both exact and inexact solution of the correction equation. The eigenvalues $\lambda_1 = 3.21 \cdot 10^{-2}$ and $\lambda_4 = 1.01 \cdot 10^{-1}$ were found in the first phase of the algorithm. The validation in phase 2 found the missed eigenvalue $\lambda_{2,3} = 6.20 \cdot 10^{-2} \pm i4.61 \cdot 10^{-1}$. The exact variant is faster than implicitly restarted Arnoldi (cf. Table 7.2).

7.5 Conclusions

The strategy based on implicitly restarted Arnoldi is a reliable and fast method to compute the leftmost eigenvalues of large-scale eigenproblems, if the solves needed for the shift-and-invert and Cayley transformations can be done efficiently and exactly. By exploiting the structure of the generalized eigenvalue problem and by choosing suitable parameters for the modified Cayley transformation, the troubles caused by infinite eigenvalues are circumvented and no purification is needed.

If the solves that are needed for the transformations cannot be done exactly, the Jacobi-Davidson QZ method is a good alternative. Following the same strategy, JDQZ is able the compute the leftmost eigenvalues, without corruption due to infinite eigenvalues. If solves can be done exactly, it is faster than implicitly restarted Arnoldi. Jacobi-Davidson puts no requirements on the matrix pencil: it

can handle both regular and singular pencils.

7.6 Appendix: Efficient preconditioning

For efficiency reasons, the preconditioner for the correction equation must be available in such a way that solves with K are relatively cheap. In practice, K is usually available as (incomplete) LU factorization of $A-\tau_0 B$ for some target τ_0. For some types of preconditioners, the process of solving correction equation (7.4.4) can be implemented rather efficiently.

7.6.1 The case $K = LU = A - \tau_0 B$

If the factorization $LU = A - \tau_0 B$ is available, then it is a good candidate for a preconditioner $K = LU$ of (7.4.3). Straightforward application of a Krylov solver to (7.4.4) requires one matrix-vector operation with A, one matrix-vector operation with B, two scalar-vector operations, one vector-vector addition, one solve with K, and one projection with $(I - Y_k H_k^{-1} Q_k^*)$ per iteration. The following lemma shows that the operation with A can be saved.

Lemma 7.6.1. *Let $A, B \in \mathbb{C}^{n\times n}$, $\alpha, \beta, \tau_0 \in \mathbb{C}$, $K = LU$ be the LU-factorization of $A - \tau_0 B$ and $\mathbf{t} \in \mathbb{C}^n$. Then $\mathbf{x} \in \mathbb{C}^n$ can be computed from*

$$\mathbf{x} = K^{-1}(\beta A - \alpha B)\mathbf{t}$$

at the expense of one matrix-vector operation with B, one solve with K, one vector-vector addition and two scalar-vector operations.

Proof. If $\beta = 0$, the result follows immediately. If $\beta \neq 0$, straightforward linear algebra gives

$$\begin{aligned} \mathbf{x} &= K^{-1}(\beta A - \alpha B)\mathbf{t} \\ &= \beta K^{-1}(A - \tau_0 B + (\tau_0 - \alpha/\beta)B)\mathbf{t} \\ &= \beta(I + (\tau_0 - \alpha/\beta)K^{-1}B)\mathbf{t}, \end{aligned}$$

where in the last step $K = LU = A - \tau_0 B$ is used. The result follows from counting the operations in the last equality. □

Lemma 7.6.1 says that one can save one matrix-vector operation per iteration of the Krylov solver. Because there are several iterations of the Krylov solver per Jacobi-Davidson iteration, this may lead to a substantial reduction in computational costs. The reduction also depends on the sparsity of A. Note that for JDQR for the ordinary eigenproblem ($B = I$), no matrix-vector operations are needed during iterations of the linear solver.

7.6.2 The case $K = \mathbf{ILU}(A - \tau_0 B)$

If an exact LU-factorization is not available, then in some cases still a computational gain can be obtained. Suppose that the incomplete factorization can be written as

$$K = (L_A - \tau_0 L_B + D)D^{-1}(U_A - \tau_0 U_B + D), \tag{7.6.1}$$

where $A - \tau_0 B = L_A - \tau_0 L_B + \mathrm{diag}(A - \tau_0 B) + U_A - \tau_0 U_B$, and L_A, L_B and U_A, U_B are the strictly lower and strictly upper triangular parts of A and B, respectively. Incomplete LU factorizations and modifications can be written in this form. Then Eisenstat's trick [42, 157] can be applied to save one multiplication with A. In practice, $ILU(0)$-factorizations are a popular choice as preconditioner for the Jacobi-Davidson correction equation, and hence Eisenstat's trick can be applied if two-sided preconditioning is used. With K as in (7.6.1), the system is first (diagonally) preconditioned with D:

$$\tilde{A} - \tau_0 \tilde{B} = D^{-1/2}(A - \tau_0 B)D^{-1/2}$$

It then follows that

$$\tilde{K} = (L_{\tilde{A}} - \tau_0 L_{\tilde{B}} + I)(U_{\tilde{A}} - \tau_0 U_{\tilde{B}} + I) \equiv \tilde{K}_l \tilde{K}_r.$$

With $D_{\tilde{A}\tilde{B}} = \mathrm{diag}(\tilde{A} - \tau_0 \tilde{B})$, an application of $\tilde{K}_l^{-1}(\tilde{A} - \theta\tilde{B})\tilde{K}_r^{-1}$ to a vector $\mathbf{z}$ can be rewritten as

$$\begin{aligned} & (L_{\tilde{A}} - \tau_0 L_{\tilde{B}} + I)^{-1}(\tilde{A} - \theta\tilde{B})(U_{\tilde{A}} - \tau_0 U_{\tilde{B}} + I)^{-1}\mathbf{z} \\ = \;& \tilde{K}_l^{-1}(L_{\tilde{A}} - \tau_0 L_{\tilde{B}} + I + D_{\tilde{A}\tilde{B}} + (\tau_0 - \theta)B - 2I + U_{\tilde{A}} - \tau_0 U_{\tilde{B}} + I)\tilde{K}_r^{-1}\mathbf{z} \\ = \;& \mathbf{t} + (L_{\tilde{A}} - \tau_0 L_{\tilde{B}} + I)^{-1}(\mathbf{z} + ((\tau_0 - \theta)B + D_{\tilde{A}\tilde{B}} - 2I)\mathbf{t}) \end{aligned}$$

where $\mathbf{t} = (U_{\tilde{A}} - \tau_0 U_{\tilde{B}} + I)^{-1}\mathbf{z}$. In this way a multiplication with $\tilde{A}$ is saved.

In order to apply Eisenstat's trick to the deflated Jacobi-Davidson correction, twosided preconditioning is needed. Denote the deflated preconditioner by

$$K = ((I - ZZ^*)K_l)(K_r(I - QQ^*)) \equiv \tilde{K}_l \tilde{K}_r,$$

where $K_l K_r$ is still of the form (7.6.1). The twosided preconditioned deflated operator is

$$\tilde{K}_l^{\dagger}(I - ZZ^*)(\beta A - \alpha B)(I - QQ^*)\tilde{K}_r^{\dagger}, \tag{7.6.2}$$

where $P^{\dagger}\mathbf{s}$ must be read as "solve $\mathbf{t}$ from $P\mathbf{t} = \mathbf{s}$". Straightforward application of the operator (7.6.2) consists of the following operations:

1. Solve $\tilde{\mathbf{t}}$ with $Q^*\tilde{\mathbf{t}} = 0$ from $(I - QQ^*)K_r(I - QQ^*)\tilde{\mathbf{t}} = \mathbf{s}$ for $\mathbf{s}$ with $Q^*K_r^{-1}\mathbf{s} = 0$,
2. Compute $\mathbf{p} = (I - ZZ^*)(\beta A - \alpha B)\tilde{\mathbf{t}}$,
3. Solve $\mathbf{t}$ with $Q^*K_r^{-1}\mathbf{t} = 0$ from $(I - ZZ^*)K_l\mathbf{t} = \mathbf{p}$, for $\mathbf{p}$ with $Z^*\mathbf{p} = 0$.

The requirement $Q^*K_r^{-1}\mathbf{t} = 0$ in step 3 is needed to guarantee that the domain and image space of (7.6.2) are equal[3]. Following the approach in [141], these operations can be simplified. Using the relation $Q^*\tilde{\mathbf{t}} = Q^*K_r^{-1}\mathbf{s} = 0$ in step 1, $\tilde{\mathbf{t}}$ can be computed as $\tilde{\mathbf{t}} = K_r^{-1}\mathbf{s}$ and the projection with $(I - QQ^*)$ on the right in step 2 can be saved. Step 3 can be worked out similarly: $\mathbf{t} = \tilde{\mathbf{p}} - K_l^{-1}ZH_l^{-1}Q^*K_r^{-1}\tilde{\mathbf{p}}$ with $\tilde{\mathbf{p}} = K_l^{-1}\mathbf{p}$ and $H_l = Q^*K_r^{-1}K_l^{-1}Z$. Note that the solution $\mathbf{t}$ also satisfies the equalities

$$Q^*K_r^{-1}\mathbf{t} = 0, \quad (I - ZZ^*)K_l\mathbf{t} = (I - ZZ^*)\mathbf{p},$$

for arbitrary $\mathbf{p}$. This saves the projection with $(I - ZZ^*)$ on the left in step 2. These observations lead to the following efficient scheme for the application of the twosided preconditioned deflated operator (7.6.2):

1. Compute $\tilde{\mathbf{p}} = K_l^{-1}(\beta A - \alpha B)K_r^{-1}\tilde{\mathbf{t}}$,
2. Compute $\mathbf{t} = \tilde{\mathbf{p}} - K_l^{-1}ZH_l^{-1}Q^*K_r^{-1}\tilde{\mathbf{p}}$.

Step 1 now can be computed using Eisenstat's trick (if the preconditioner is of the form (7.6.1)). Because the resulting vector $\mathbf{t}$ is orthogonal to $K_r^{-*}Q$, no intermediate projections are needed for the Krylov subspace method, provided the initial vector is orthogonal to $K_r^{-*}Q$. The right-hand side must be preconditioned as well:

$$\tilde{\mathbf{r}}_i = (I - K_l^{-1}ZH_l^{-1}Q^*K_r^{-1})K_l^{-1}\mathbf{r}_i.$$

The approximate solution $\mathbf{t}$ of the correction equation (the new expansion vector) can be obtained from the solution $\mathbf{y}$ of the twosided preconditioned deflated correction equation

$$P_l(K_l^{-1}(\beta A - \alpha B)K_r^{-1})P_l\mathbf{y} = -\tilde{\mathbf{r}}_i,$$

with $P_l = I - K_l^{-1}ZH_l^{-1}Q^*K_r^{-1}$ as follows:

$$\mathbf{t} = K_r^{-1}\mathbf{y},$$

where explicit projection with $(I - QQ^*)$ is not needed because $Q^*K_r^{-1}\mathbf{y} = 0$.

Addendum

Except for some changes in notation and the appendix, this chapter is also available as [121]

> Joost Rommes, *Arnoldi and Jacobi-Davidson methods for generalized eigenvalue problems $A\mathbf{x} = \lambda B\mathbf{x}$ with singular B*, Preprint 1339, Utrecht University, 2005 (revised 2007),

and has been accepted for publication in Mathematics of Computation.

[3]Thanks to Gerard Sleijpen for observing this choice, which saves one projection per iteration.