Chapter 6

Numerical computation of eigenvalues

Introduction

Calculating the eigenvalues and eigenvectors of a matrix is a task often encountered in scientific and engineering applications. Eigenvalue problems naturally arise in quantum physics, solid mechanics, structural engineering and molecular dynamics, to name just a few applications. The aim of this chapter is to present an overview of the standard methods for calculating eigenvalues and eigenvectors numerically. We focus predominantly on the case of a Hermitian matrix $\mathsf{A} \in \mathbf{C}^{n\times n}$, which is technically simpler and arises in many applications. The reader is invited to go through the background material in Appendix A.5 before reading this chapter. The rest of this chapter is organized as follows

- In Section 6.1, we make general remarks concerning the calculation of eigenvalues.
- In Section 6.2, we present standard methods based on a simple vector iteration.
- In Section 6.3, we present a method for calculating several eigenvectors simultaneously, based on iterating a subspace.
- In Section 6.4, we present method for constructing an approximation of the eigenvectors in a given subspace of $\mathbf{C}^n$.

6.1 Numerical methods for eigenvalue problems: general remarks

As mentioned in Appendix A.5, a complex number $\lambda \in \mathbf{C}$ is an eigenvalue of $\mathsf{A} \in \mathbf{C}^{n\times n}$ if and only if λ is a root of the characteristic polynomial $p_A \colon \mathbf{C} \to \mathbf{C}$ of A, which is given by

$$p_A(\lambda) = \det(\mathsf{A} - \lambda \mathsf{I}).$$

One may, therefore, calculate the eigenvalues of A by calculating the roots of the polynomial p_A using, for example, one of the methods presented in Chapter 5. While feasible for small matrices, this approach is not viable for large matrices, because the number of floating point operations required for calculating calculating the coefficients of the characteristic polynomial scales as the factorial of n.

In view of the prohibitive computational cost required for calculating the characteristic polynomial, other methods are required for solving large eigenvalue problems numerically. All the methods that we study in this chapter are of iterative nature. While some of them are aimed at calculating all the eigenpairs of the matrix A, other methods enable to calculate only a small number of eigenpairs at a lower computational cost, which is often desirable. Indeed, calculating all the eigenvalues of a large matrix is computationally expensive; on a personal computer, the following Julia code takes well over a second to terminate:

```
import LinearAlgebra
A = rand(2000, 2000)
LinearAlgebra.eigen(A)
```

In many applications, the matrix A is sparse, and in this case it is important to use algorithms for eigenvalue problems that do not destroy the sparsity structure. Note that the eigenvectors of a sparse matrix are generally not sparse.

To conclude this section, we introduce some notation used throughout this chapter. For a diagonalizable matrix A, we denote the eigenvalues by $\lambda_1, \dots, \lambda_n$, with $|\lambda_1| \geqslant |\lambda_2| \geqslant \dots \geqslant |\lambda_n|$. The associated normalized eigenvectors are denoted by $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$. Therefore, it holds that

$$\mathsf{V}^{-1}\mathsf{A}\mathsf{V} = \mathsf{D} = \operatorname{diag}(\lambda_1, \dots, \lambda_n), \qquad \text{where } \mathsf{V} = \begin{pmatrix} \boldsymbol{v}_1 & \dots & \boldsymbol{v}_n \end{pmatrix}.$$

6.2 Simple vector iterations

In this section, we present simple iterative methods aimed at calculating just one eigenvector of the matrix A, which we assume to be diagonalizable for simplicity.

6.2.1 The power iteration

The power iteration is the simplest method for calculating the eigenpair associated with the eigenvalue of A with largest modulus. Since the eigenvectors of A span $\mathbf{C}^n$, any vector $\boldsymbol{x}_0$ may be decomposed as

$$\boldsymbol{x}_0 = \alpha_1 \boldsymbol{v}_1 + \dots + \alpha_n \boldsymbol{v}_n. \tag{6.1}$$

The idea of the power iteration is to repeatedly left-multiply this vector by the matrix A, in order to amplify the coefficient of $\boldsymbol{v}_0$ relative to the other ones. Indeed, notice that

$$\mathsf{A}^k \boldsymbol{x}_0 = \lambda_1^k \alpha_1 \boldsymbol{v}_1 + \dots + \lambda_n^k \alpha_n \boldsymbol{v}_n.$$

If λ_1 is strictly greater in modulus than the other eigenvalues, and if $\alpha_1 \neq 0$, then for large k the vector $\mathsf{A}^k \boldsymbol{x}_0$ is approximately aligned, in a sense made precise below, with the eigenvector $\boldsymbol{v}_1$. In order to avoid overflow errors at the numerical level, the iterates are normalized at each iteration. The power iteration is presented in Algorithm 8.

To precisely quantify the convergence of the power method, we introduce the notion of *acute*

Algorithm 8 Power iteration

$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
for $i \in \{1, 2, \dots\}$ **do**
  $\boldsymbol{x} \leftarrow \mathsf{A}\boldsymbol{x}$
  $\boldsymbol{x} \leftarrow \boldsymbol{x}/\|\boldsymbol{x}\|$
end for

angle between vectors of $\mathbf{C}^n$.

$$\begin{aligned}\angle(\boldsymbol{x}, \boldsymbol{y}) &= \arccos\left(\frac{|\boldsymbol{x}^*\boldsymbol{y}|}{\sqrt{\boldsymbol{x}^*\boldsymbol{x}}\sqrt{\boldsymbol{y}^*\boldsymbol{y}}}\right) \\ &= \arcsin\left(\frac{\|(\mathsf{I} - \mathsf{P}_{\boldsymbol{y}})\boldsymbol{x}\|}{\|\boldsymbol{x}\|}\right), \qquad \mathsf{P}_{\boldsymbol{y}} := \frac{\boldsymbol{y}\boldsymbol{y}^*}{\boldsymbol{y}^*\boldsymbol{y}}.\end{aligned}$$

This definition generalizes the familiar notion of angle for vectors in $\mathbf{R}^2$ or $\mathbf{R}^3$, and we note that the angle function satisfies $\angle(\mathrm{e}^{\mathrm{i}\theta_1}\boldsymbol{x}, \mathrm{e}^{\mathrm{i}\theta_2}\boldsymbol{y}) = \angle(\boldsymbol{x}, \boldsymbol{y})$ as well as $\angle(\boldsymbol{x}, \boldsymbol{y}) \in [0, \pi/2]$. We can then prove the following convergence result.

Proposition 6.1 (Convergence of the power iteration)**.** *Suppose that* A *is diagonalizable and that* $|\lambda_1| > |\lambda_2|$*. Then, for every initial guess with* $\alpha_1 \neq 0$*, the sequence* $(\boldsymbol{x}_k)_{k\geqslant 0}$ *generated by the power iteration satisfies*

$$\lim_{k\to\infty} \angle(\boldsymbol{x}_k, \boldsymbol{v}_1) = 0.$$

Proof. By construction, it holds that

$$\boldsymbol{x}_k = \frac{\lambda_1^k\alpha_1\boldsymbol{v}_1 + \cdots + \lambda_n^k\alpha_n\boldsymbol{v}_n}{\|\lambda_1^k\alpha_1\boldsymbol{v}_1 + \cdots + \lambda_n^k\alpha_n\boldsymbol{v}_n\|} = \mathrm{e}^{\mathrm{i}\theta_k}\frac{\boldsymbol{v}_1 + \frac{\lambda_2^k\alpha_2}{\lambda_1^k\alpha_1}\boldsymbol{v}_2 + \cdots + \frac{\lambda_n^k\alpha_2}{\lambda_1^k\alpha_1}\boldsymbol{v}_n}{\left\|\boldsymbol{v}_1 + \frac{\lambda_2^k\alpha_2}{\lambda_1^k\alpha_1}\boldsymbol{v}_2 + \cdots + \frac{\lambda_n^k\alpha_n}{\lambda_1^k\alpha_1}\boldsymbol{v}_n\right\|}, \tag{6.2}$$

where

$$\mathrm{e}^{\mathrm{i}\theta_k} := \frac{\lambda_1^k\alpha_1}{|\lambda_1^k\alpha_1|}.$$

It follows from (6.2) that $\mathrm{e}^{-\mathrm{i}\theta_k}\boldsymbol{x}_k \to \boldsymbol{v}_1/\|\boldsymbol{v}_1\| = \boldsymbol{v}_1$ in the limit as $k \to \infty$, where we employed the fact that $\|\boldsymbol{v}_1\| = 1$. Using the definition of the angle between two vectors in $\mathbf{C}^n$, and the continuity with respect to either argument of the $\mathbf{C}^n$ Euclidean inner product and of the arccos function, we obtain that

$$\begin{aligned}\angle(\boldsymbol{x}_k, \boldsymbol{v}_1) &= \arccos\left(\frac{|\boldsymbol{v}_1^*\boldsymbol{x}_k|}{\sqrt{\boldsymbol{v}_1^*\boldsymbol{v}_1}\sqrt{\boldsymbol{x}_k^*\boldsymbol{x}_k}}\right) = \arccos\left(|\boldsymbol{v}_1^*\boldsymbol{x}_k|\right) \\ &= \arccos\left(\left|\boldsymbol{v}_1^*\left(\mathrm{e}^{-\mathrm{i}\theta_k}\boldsymbol{x}_k\right)\right|\right) \xrightarrow[k\to\infty]{} \arccos(1) = 0,\end{aligned}$$

which concludes the proof. □

An inspection of the proof also reveals that the dominant term in the error, asymptotically

in the limit as $k \to \infty$, is the one with coefficient $\frac{\lambda_2^k \alpha_2}{\lambda_1^k \alpha_1}$. Therefore, we deduce that

$$\angle(\boldsymbol{x}_k, \boldsymbol{v}_1) = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right).$$

The convergence is slow if $|\lambda_2/\lambda_1|$ is close to one, and fast if $|\lambda_2| \ll |\lambda_1|$. Once an approximation of the eigenvector $\boldsymbol{v}_1$ has been calculated, the corresponding eigenvalue λ_1 can be estimated from the *Rayleigh quotient:*

$$\rho_{\mathsf{A}} \colon \mathbf{C}_*^n \to \mathbf{C} \colon \boldsymbol{x} \mapsto \frac{\boldsymbol{x}^* \mathsf{A} \boldsymbol{x}}{\boldsymbol{x}^* \boldsymbol{x}}. \tag{6.3}$$

For any eigenvector $\boldsymbol{v}$ of A, the corresponding eigenvalue is equal to $\rho_{\mathsf{A}}(\boldsymbol{v})$. In order to study the error on the eigenvalue λ_1 for the power iteration, we assume for simplicity that A is Hermitian and that the eigenvectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$ are orthonormal. Substituting (6.2) in the Rayleigh quotient (6.3), we obtain

$$\rho_{\mathsf{A}}(\boldsymbol{x}_k) = \frac{\lambda_1 + \left|\frac{\lambda_2^k \alpha_2}{\lambda_1^k \alpha_1}\right|^2 \lambda_2 + \cdots + \left|\frac{\lambda_n^k \alpha_n}{\lambda_1^k \alpha_1}\right|^2 \lambda_n}{1 + \left|\frac{\lambda_2^k \alpha_2}{\lambda_1^k \alpha_1}\right|^2 + \cdots + \left|\frac{\lambda_n^k \alpha_n}{\lambda_1^k \alpha_1}\right|^2}.$$

Therefore, by reducing to a common denominator we deduce

$$\begin{aligned} |\rho_{\mathsf{A}}(\boldsymbol{x}_k) - \lambda_1| &= \left| \frac{\lambda_1 + \left|\frac{\lambda_2^k \alpha_2}{\lambda_1^k \alpha_1}\right|^2 \lambda_2 + \cdots + \left|\frac{\lambda_n^k \alpha_n}{\lambda_1^k \alpha_1}\right|^2 \lambda_n}{1 + \left|\frac{\lambda_2^k \alpha_2}{\lambda_1^k \alpha_1}\right|^2 + \cdots + \left|\frac{\lambda_n^k \alpha_n}{\lambda_1^k \alpha_1}\right|^2} - \lambda_1 \right| \\ &\leqslant \left|\frac{\lambda_2^k \alpha_2}{\lambda_1^k \alpha_1}\right|^2 |\lambda_2 - \lambda_1| + \cdots + \left|\frac{\lambda_n^k \alpha_n}{\lambda_1^k \alpha_1}\right|^2 |\lambda_n - \lambda_1| = \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right). \end{aligned}$$

The convergence of the eigenvalue in the particular case of a Hermitian matrix is faster than for a general matrix in $\mathbf{C}^{n \times n}$. For general matrices, it is possible to show using a similar argument that the error is of order $\mathcal{O}\big(|\lambda_2/\lambda_1|^k\big)$ in the limit as $k \to \infty$.

Essential convergence. It is useful at this point to introduce the concept of *essential convergence*. A sequence $(\boldsymbol{x}_k)$ in $\mathbf{C}^n$ is said to *converge essentially* to a vector $\boldsymbol{x}_\infty$ if there exists a sequence of complex numbers $\big(\mathrm{e}^{\mathrm{i}\phi_k}\big)$ such that the sequence $\big(\mathrm{e}^{\mathrm{i}\phi_k}\boldsymbol{x}_k\big)$ converges to $\boldsymbol{x}_\infty$. Equivalently, the sequence $(\boldsymbol{x}_k)$ converges essentially to $\boldsymbol{x}_\infty$ if $\angle(\boldsymbol{x}_k, \boldsymbol{x}_\infty)$ converges to 0. Proving this equivalence is the goal of Exercise 6.11. Reformulated in this new terminology, Proposition 6.1 states that the sequence $(\boldsymbol{x}_k)$ obtained from the power iteration converges essentially to $\boldsymbol{v}_1$.

6.2.2 Inverse iteration

The power iteration is simple but enables to calculate only the dominant eigenvalue of the matrix A, i.e. the eigenvalue of largest modulus. In addition, the convergence of the method is slow when $|\lambda_2| \approx |\lambda_1|$.

The inverse iteration enables a more efficient calculation of not only the dominant eigenvalue but also the other eigenvalues of A. It is based on applying the power iteration to $(\mathsf{A} - \mu\mathsf{I})^{-1}$,

where $\mu \in \mathbf{C}$ is a shift. The eigenvalues of $(\mathsf{A}-\mu\mathsf{I})^{-1}$ are given by $(\lambda_1-\mu)^{-1}, \dots, (\lambda_n-\mu)^{-1}$, with associated eigenvectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$. If $0 < |\lambda_J - \mu| < |\lambda_j - \mu|$ for all $j \neq J$, then the dominant eigenvalue of the matrix $(\mathsf{A} - \mu\mathsf{I})^{-1}$ is $(\lambda_J - \mu)^{-1}$, and so the power iteration applied to this matrix yields an approximation of the eigenvector $\boldsymbol{v}_J$. In other words, the inverse iteration with shift μ enables to calculate an approximation of the eigenvector of A corresponding to the eigenvalue nearest μ. The inverse iteration is presented in Algorithm 9. In practice, the inverse matrix $(\mathsf{A} - \mu\mathsf{I})^{-1}$ need not be calculated, and it is often preferable to solve a linear system at each iteration.

Algorithm 9 Inverse iteration

$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
for $i \in \{1, 2, \dots\}$ **do**
  Solve $(\mathsf{A} - \mu\mathsf{I})\boldsymbol{y} = \boldsymbol{x}$
  $\boldsymbol{x} \leftarrow \boldsymbol{y}/\|\boldsymbol{y}\|$
end for
$\lambda \leftarrow \boldsymbol{x}^*\mathsf{A}\boldsymbol{x}/\boldsymbol{x}^*\boldsymbol{x}$
return $\boldsymbol{x}, \lambda$

An application of Proposition 6.1 immediately gives the following convergence result for the inverse iteration.

Proposition 6.2 (Convergence of the inverse iteration)**.** *Assume that $\mathsf{A} \in \mathbf{C}^n$ is diagonalizable and that there exist J and K such that*

$$0 < |\lambda_J - \mu| < |\lambda_K - \mu| \leqslant |\lambda_j - \mu| \qquad \forall j \neq J.$$

Assume also that $\alpha_J \neq 0$, where α_J is the coefficient of $\boldsymbol{v}_J$ in the expansion of $\boldsymbol{x}_0$ given in (6.1). Then the iterates of the inverse iteration satisfy

$$\lim_{k\to\infty} \angle(\boldsymbol{x}_k, \boldsymbol{v}_J) = 0.$$

More precisely,

$$\angle(\boldsymbol{x}_k, \boldsymbol{v}_J) = \mathcal{O}\left(\left|\frac{\lambda_J - \mu}{\lambda_K - \mu}\right|^k\right).$$

Proposition 6.2 states that $\boldsymbol{x}_k$ converges essentially to $\boldsymbol{v}_J$. Notice that the closer μ is to λ_J, the faster the inverse iteration converges. Note also that with $\mu = 0$, the inverse iteration enables to calculate the eigenvalue of A of smallest modulus.

6.2.3 Rayleigh quotient iteration

Since the inverse iteration is fast when μ is close to an eigenvalue λ_J, it is natural to wonder whether the method can be improved by progressively updating μ as the simulation progresses. Specifically, an approximation of the eigenvalue associated with the current vector may be employed in place of μ. This leads to the Rayleigh quotient iteration, presented in Algorithm 10.

Algorithm 10 Inverse iteration

$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
for $i \in \{1, 2, \dots\}$ **do**
  $\mu \leftarrow \boldsymbol{x}^* \mathsf{A} \boldsymbol{x} / \boldsymbol{x}^* \boldsymbol{x}$
  Solve $(\mathsf{A} - \mu \mathsf{I})\boldsymbol{y} = \boldsymbol{x}$
  $\boldsymbol{x} \leftarrow \boldsymbol{y} / \|\boldsymbol{y}\|$
end for
$\lambda \leftarrow \boldsymbol{x}^* \mathsf{A} \boldsymbol{x} / \boldsymbol{x}^* \boldsymbol{x}$
return $\boldsymbol{x}, \lambda$

It is possible to show that, when A is Hermitian, the Rayleigh quotient iteration converges to an eigenvector for almost every initial guess $\boldsymbol{x}_0$. Furthermore, if convergence to an eigenvector occurs, then μ converges cubically to the corresponding eigenvalue. See [11] and the references therein for more details.

6.3 Methods based on a subspace iteration

The subspace iteration resembles the power iteration but it is more general: not just one but several vectors are updated at each iteration.

6.3.1 Simultaneous iteration

Let $\mathsf{X}_0 = \begin{pmatrix} \boldsymbol{x}_1 & \dots & \boldsymbol{x}_p \end{pmatrix}$ denote an initial set of linearly independent vectors. Before we present the simultaneous iteration, we recall a statement concerning the QR decomposition of a matrix, which is related to the Gram–Schmidt orthonormalization process. We recall that the Gram–Schmidt method enables to construct, starting form an ordered set of vectors $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_p\}$ in $\mathbf{C}^n$, a new set of vectors $\{\boldsymbol{q}_1, \dots, \boldsymbol{q}_p\}$ which are *orthonormal* and span the same subspace of $\mathbf{C}^n$ as the original vectors.

Proposition 6.3 (Reduced QR decomposition)**.** *Assume that $\mathsf{X} \in \mathbf{C}^{n \times p}$ has linearly independent columns. Then there exist a matrix $\mathsf{Q} \in \mathbf{C}^{n \times p}$ with orthonormal columns and an upper triangular matrix $\mathsf{R} \in \mathbf{C}^{p \times p}$ such that the following factorization holds:*

$$\mathsf{X} = \mathsf{QR}. \tag{6.4}$$

This decomposition is known as a reduced QR decomposition *if $p < n$, or simply* QR decomposition *if $p = n$, in which case X is a square matrix and Q is a unitary matrix. The decomposition is unique if we require that the diagonal elements of R are real and positive.*

Proof. The statement is clear when $p = 1$. Reasoning by induction, we assume that the result is true up to $p - 1$, and prove that it then also holds true for p. We wish to show that there is a unique matrix $\mathsf{Q} \in \mathbf{C}^{n \times p}$ with orthonormal columns and a unique upper triangular matrix $\mathsf{R} \in \mathbf{C}^{p \times p}$ with real and positive diagonal elements such that (6.4) is satisfied. To this

end, we decompose the matrices Q and R as follows:

$$\mathsf{Q} = \begin{pmatrix} \mathsf{Q}_{p-1} & \boldsymbol{q} \end{pmatrix}, \qquad \mathsf{R} = \begin{pmatrix} \mathsf{R}_{p-1} & \boldsymbol{r} \\ \mathbf{0}_{p-1}^T & r \end{pmatrix}. \tag{6.5}$$

Here $\mathsf{Q}_{p-1} \in \mathbf{C}^{n\times(p-1)}$ is a matrix with orthonormal columns, $\mathsf{R} \in \mathbf{C}^{(p-1)\times(p-1)}$ is an upper triangular matrix with positive real diagonal elements, $\boldsymbol{q} \in \mathbf{C}^n$ is a normalized vector orthogonal to all the columns of Q_{p-1}, $\boldsymbol{r} \in \mathbf{C}^{n-1}$ is a vector and $r \in \mathbf{R}_{>0}$ is a scalar. Let us also denote by $\mathsf{X}_{p-1} \in \mathbf{C}^{n\times(p-1)}$ the matrix containing the $p-1$ first columns of X, and by $\boldsymbol{x}_p \in \mathbf{C}^n$ the p-th column of X. Substituting (6.5) into (6.4), we then obtain

$$\begin{pmatrix} \mathsf{X}_{p-1} & \boldsymbol{x}_p \end{pmatrix} = \begin{pmatrix} \mathsf{Q}_{p-1}\mathsf{R}_{p-1} & \mathsf{Q}_{p-1}\boldsymbol{r} + \boldsymbol{q}r \end{pmatrix}, \tag{6.6}$$

By the induction hypothesis, there exist a unique choice of matrices Q_{p-1} and R_{p-1} with the required structure such that $\mathsf{X}_{p-1} = \mathsf{Q}_{p-1}\mathsf{R}_{p-1}$. Comparing the last column of both sides in (6.6), we obtain

$$\boldsymbol{x}_p = \mathsf{Q}_{p-1}\boldsymbol{r} + \boldsymbol{q}r. \tag{6.7}$$

Left-multiplying both sides by Q_{p-1}^* and employing the orthogonality between $\boldsymbol{q}$ and the columns of Q_{p-1}, we deduce that necessarily $\boldsymbol{r} = \mathsf{Q}_{p-1}^*\boldsymbol{x}_p$. It then follows from (6.7) that

$$\boldsymbol{q} = \frac{1}{r}\left(\boldsymbol{x}_p - \mathsf{Q}_{p-1}\mathsf{Q}_{p-1}^*\boldsymbol{x}_p\right), \qquad r = \left\|\boldsymbol{x}_p - \mathsf{Q}_{p-1}\mathsf{Q}_{p-1}^*\boldsymbol{x}_p\right\|.$$

It is simple to check that $\boldsymbol{q}$ is indeed orthogonal to the columns of Q, which concludes the proof. Note that $\mathsf{Q}_{p-1}\mathsf{Q}_{p-1}^*\boldsymbol{x}_p$ is the orthogonal projection of $\boldsymbol{x}_p$ onto the subspace spanned by the column of Q_{p-1}. □

Note that the columns of the matrix Q of the decomposition coincide with the vectors that would be obtained by applying the Gram–Schmidt method to the columns of the matrix X. In fact, the Gram–Schmidt process is one of several methods by which the QR decomposition can be calculated in practice.

Algorithm 11 Simultaneous iteration

$\mathsf{X} \leftarrow \mathsf{X}_0$
for $k \in \{1, 2, \dots\}$ **do**
  $\mathsf{Q}_k\mathsf{R}_k = \mathsf{A}\mathsf{X}_{k-1}$ (QR decomposition).
  $\mathsf{X}_k \leftarrow \mathsf{Q}_k$.
end for

The simultaneous iteration method is presented in Algorithm 11. Like the normalization in the power iteration Algorithm 8, the QR decomposition performed at each step in Algorithm 11 enables to avoid overflow errors. Notice that when $p = 1$, the simultaneous iteration reduces to the power iteration. We emphasize that the factorization step at each iteration does not influence the subspace spanned by the columns of X. Therefore, this subspace after k iterations coincides with that spanned by the columns of the matrix $\mathsf{A}^k\mathsf{X}_0$. In fact, in exact arithmetic, it would be equivalent to perform the QR decomposition only once as a final step, after the **for**

loop. Indeed, denoting by $\mathsf{Q}_k\mathsf{R}_k$ the QR decomposition of $\mathsf{A}\mathsf{X}_{k-1}$, we have

$$\begin{aligned} &\mathsf{X}_k = \mathsf{A}\mathsf{X}_{k-1}\mathsf{R}_k^{-1} = \mathsf{A}^2\mathsf{X}_{k-2}\mathsf{R}_{k-1}^{-1}\mathsf{R}_k^{-1} = \cdots = \mathsf{A}^k\mathsf{X}_0\mathsf{R}_1^{-1}\dots\mathsf{R}_k^{-1} \\ \Leftrightarrow \quad &\mathsf{X}_k(\mathsf{R}_k\dots\mathsf{R}_1) = \mathsf{A}^k\mathsf{X}_0. \end{aligned}$$

Since X_k has orthonormal columns and $\mathsf{R}_k\dots\mathsf{R}_1$ is an upper triangular matrix (see Exercise 4.3) with real positive elements on the diagonal (check this!), it follows that X_k can be obtained by QR factorization of $\mathsf{A}^k\mathsf{X}_0$. In order to show the convergence of the simultaneous iteration, we begin by proving the following preparatory lemma.

Lemma 6.4 (Continuity of the QR decomposition)**.** *If $\mathsf{Q}_k\mathsf{R}_k \to \mathsf{Q}\mathsf{R}$, where Q is orthogonal and R is upper triangular with positive real entries on the diagonal, then $\mathsf{Q}_k \to \mathsf{Q}$.*

Proof. We reason by contradiction and assume there is $\varepsilon > 0$ and a subsequence $(\mathsf{Q}_{k_n})_{n\geqslant 0}$ such that $\|\mathsf{Q}_{k_n} - \mathsf{Q}\| \geqslant \varepsilon$ for all n. Since the set of unitary matrices is a compact subset of $\mathsf{C}^{n\times n}$, there exists a further subsequence $(\mathsf{Q}_{k_{n_m}})_{m\geqslant 0}$ that converges to a limit Q_∞ which is also a unitary matrix and at least ε away in norm from Q. But then

$$\mathsf{R}_{k_{n_m}} = \mathsf{Q}_{k_{n_m}}^{-1}(\mathsf{Q}_{k_{n_m}}\mathsf{R}_{k_{n_m}}) = \mathsf{Q}_{k_{n_m}}^{*}(\mathsf{Q}_{k_{n_m}}\mathsf{R}_{k_{n_m}}) \xrightarrow[m\to\infty]{} \mathsf{Q}_\infty^{*}(\mathsf{Q}\mathsf{R}) =: \mathsf{R}_\infty.$$

Since R_k is upper triangular with positive diagonal elements for all k, clearly R_∞ is also upper triangular with positive diagonal elements. But then $\mathsf{Q}_\infty\mathsf{R}_\infty = \mathsf{Q}\mathsf{R}$, and by uniqueness of the decomposition we deduce that $\mathsf{Q} = \mathsf{Q}_\infty$, which is a contradiction. □

Before presenting the convergence theorem, we introduce the following terminology: we say that $\mathsf{X}_k \in \mathbf{C}^{n\times p}$ converges essentially to a matrix X_∞ if each column of X_k converges essentially to the corresponding column of X_∞. We prove the convergence in the Hermitian case for simplicity. In the general case of $\mathsf{A} \in \mathbf{C}^{n\times n}$, it cannot be expected that X_k converges essentially to V, because the columns of X_k are orthogonal but eigenvectors may not be orthogonal. In this case, the columns of X_k converge not to the eigenvectors but to the so-called Schur vectors of A; see [11] for more information.

Theorem 6.5 (Convergence of the simultaneous iteration)**.** *Assume that $\mathsf{A} \in \mathbf{C}^{n\times n}$ is Hermitian, that $\mathsf{X}_0 \in \mathbf{C}^{n\times p}$ has linearly independent columns, and finally that the subspace spanned by the column of X_0 satisfies*

$$\operatorname{col}(\mathsf{X}_0) \cap \operatorname{Span}\{\boldsymbol{v}_{p+1}, \dots, \boldsymbol{v}_n\} = \varnothing. \tag{6.8}$$

If it holds that

$$\lambda_1 > \lambda_2 > \cdots > \lambda_p > \lambda_{p+1} \geqslant \lambda_{p+2} \geqslant \dots \geqslant \lambda_n, \tag{6.9}$$

then X_k converges essentially *to $\mathsf{V}_1 := \begin{pmatrix} \boldsymbol{v}_1 & \dots & \boldsymbol{v}_p \end{pmatrix}$.*

Proof. Let $\mathsf{B} = \mathsf{V}^{-1}\mathsf{X}_0 \in \mathbf{C}^{n\times p}$, so that $\mathsf{X}_0 = \mathsf{V}\mathsf{B}$, and note that $\mathsf{A}^k\mathsf{X}_0 = \mathsf{V}\mathsf{D}^k\mathsf{B}$. We denote by $\mathsf{B}_1 \in \mathbf{C}^{p\times p}$ and $\mathsf{B}_2 \in \mathbf{C}^{(n-p)\times p}$ the upper $p\times p$ and lower $(n-p)\times p$ blocks of B, respectively. The matrix B_1 is nonsingular, otherwise the assumption (6.8) would not hold. Indeed, if there was a nonzero vector $\boldsymbol{z} \in \mathbf{C}^p$ such that $\mathsf{B}_1\boldsymbol{z} = 0$, then

$$\mathsf{X}_0\boldsymbol{z} = \mathsf{V}\begin{pmatrix}\mathsf{B}_1\\ \mathsf{B}_2\end{pmatrix}\boldsymbol{z} = \begin{pmatrix}\mathsf{V}_1 & \mathsf{V}_2\end{pmatrix}\begin{pmatrix}\mathbf{0}\\ \mathsf{B}_2\boldsymbol{z}\end{pmatrix} = \mathsf{V}_2\mathsf{B}_2\boldsymbol{z}.$$

implying that $\mathsf{X}_0\boldsymbol{z} \in \mathrm{col}(\mathsf{X}_0)$ is a linear combination of the vectors in $\mathsf{V}_2 = \begin{pmatrix}\boldsymbol{v}_{p+1} & \dots \boldsymbol{v}_n\end{pmatrix}$, which contradicts the assumption. We also denote by D_1 and D_2 the $p\times p$ upper-left and the $(n-p)\times(n-p)$ lower-right blocks of D, respectively. From the expression of $\mathsf{A}^k\mathsf{X}_0$, we have

$$\begin{aligned}\mathsf{A}^k\mathsf{X}_0 &= \begin{pmatrix}\mathsf{V}_1 & \mathsf{V}_2\end{pmatrix}\begin{pmatrix}\mathsf{D}_1^k & \\ & \mathsf{D}_2^k\end{pmatrix}\begin{pmatrix}\mathsf{B}_1\\ \mathsf{B}_2\end{pmatrix} = \mathsf{V}_1\mathsf{D}_1^k\mathsf{B}_1 + \mathsf{V}_2\mathsf{D}_2^k\mathsf{B}_2,\\ &= \left(\mathsf{V}_1 + \mathsf{V}_2\mathsf{D}_2^k\mathsf{B}_2\mathsf{B}_1^{-1}\mathsf{D}_1^{-k}\right)\mathsf{D}_1^k\mathsf{B}_1. \end{aligned} \tag{6.10}$$

The second term in the bracket on the right-hand side converges to zero in the limit as $k\to\infty$ by (6.9). Let $\widetilde{\mathsf{Q}_k}\widetilde{\mathsf{R}_k}$ denote the reduced QR decomposition of the bracketed term. By Lemma 6.4, which we proved for the standard QR decomposition but also holds for the reduced one, we deduce from $\widetilde{\mathsf{Q}_k}\widetilde{\mathsf{R}_k} \to \mathsf{V}_1$ that $\widetilde{\mathsf{Q}_k} \to \mathsf{V}_1$. Rearranging (6.10), we have

$$\mathsf{A}^k\mathsf{X}_0 = \widetilde{\mathsf{Q}_k}(\widetilde{\mathsf{R}_k}\mathsf{D}_1^k\mathsf{B}_1).$$

Since the matrix between brackets is a $p\times p$ square invertible matrix, this equation implies that $\mathrm{col}(\mathsf{A}^k\mathsf{X}_0) = \mathrm{col}(\widetilde{\mathsf{Q}_k})$. Denoting by $\mathsf{Q}_k\mathsf{R}_k$ the QR decomposition of $\mathsf{A}_k\mathsf{X}_0$, we therefore have $\mathrm{col}(\mathsf{Q}_k) = \mathrm{col}(\widetilde{\mathsf{Q}_k})$, and so the projectors on these subspaces are equal. We recall that, for a set of orthonormal vectors $\boldsymbol{r}_1,\dots,\boldsymbol{r}_p$ gathered in a matrix $\mathsf{R} = \begin{pmatrix}\boldsymbol{r}_1 & \dots & \boldsymbol{r}_p\end{pmatrix}$, the projector on $\mathrm{col}(\mathsf{R}) = \mathrm{Span}\{\boldsymbol{r}_1,\dots,\boldsymbol{r}_p\} \subset \mathbf{C}^n$ is the square $n\times n$ matrix

$$\mathsf{R}\mathsf{R}^* = \boldsymbol{r}_1\boldsymbol{r}_1^* + \cdots + \boldsymbol{r}_p\boldsymbol{r}_p^*.$$

Consequently, the equality of the projectors implies $\mathsf{Q}_k\mathsf{Q}_k^* = \widetilde{\mathsf{Q}_k}\widetilde{\mathsf{Q}_k}^*$. Now, we want to establish the essential convergence of Q_k to V_1. To this end, we reason by induction, relying on the fact that the first k columns of X_0 undergo a simultaneous iteration independent of the other columns. For example, the first column simply undergoes a power iteration, and so it converges essentially to $\boldsymbol{v}_1$. Assume now that the columns 1 to $p-1$ of Q_k converge essentially to $\boldsymbol{v}_1,\dots,\boldsymbol{v}_{p-1}$. Then the p-th column of Q_k at iteration k, which we denote by $\boldsymbol{q}_p^{(k)}$, satisfies

$$\begin{aligned}\boldsymbol{q}_p^{(k)}\boldsymbol{q}_p^{(k)*} &= \mathsf{Q}_k\mathsf{Q}_k^* - \boldsymbol{q}_1^{(k)}\boldsymbol{q}_1^{(k)*} - \cdots - \boldsymbol{q}_{p-1}^{(k)}\boldsymbol{q}_{p-1}^{(k)*} = \widetilde{\mathsf{Q}}_k\widetilde{\mathsf{Q}}_k^* - \boldsymbol{q}_1^{(k)}\boldsymbol{q}_1^{(k)*} - \cdots - \boldsymbol{q}_{p-1}^{(k)}\boldsymbol{q}_{p-1}^{(k)*}\\ &\xrightarrow[k\to\infty]{} \mathsf{V}_1\mathsf{V}_1^* - \boldsymbol{v}_1\boldsymbol{v}_1^* - \cdots - \boldsymbol{v}_{p-1}\boldsymbol{v}_{p-1}^* = \boldsymbol{v}_p\boldsymbol{v}_p^*.\end{aligned}$$

Therefore, noting that $|a| = \sqrt{a\overline{a}}$ for every $a \in \mathbf{C}$, we deduce

$$|\boldsymbol{v}_p^* \boldsymbol{q}_p^{(k)}| = \sqrt{\boldsymbol{v}_p^* \boldsymbol{q}_p^{(k)} \boldsymbol{q}_p^{(k)^*} \boldsymbol{v}_p} \xrightarrow[k\to\infty]{} \sqrt{\boldsymbol{v}_p^* \boldsymbol{v}_p \boldsymbol{v}_p^* \boldsymbol{v}_p} = 1,$$

which shows that $\angle(\boldsymbol{q}_p^{(k)}, \boldsymbol{v}_p)$ converges to 0. Finally, observing that

$$\left\| \mathrm{e}^{-i\theta_k} \boldsymbol{q}_p^{(k)} - \boldsymbol{v}_p \right\|^2 = 2 - 2|\boldsymbol{v}_p^* \boldsymbol{q}_p^{(k)}| \xrightarrow[k\to\infty]{} 0, \qquad \theta_k = \frac{\boldsymbol{v}_p^* \boldsymbol{q}_p^{(k)}}{|\boldsymbol{v}_p^* \boldsymbol{q}_p^{(k)}|},$$

we conclude that $\boldsymbol{q}_p^{(k)}$ converges essentially to $\boldsymbol{v}_p$. □

In addition to this convergence result, it is possible to show that the error satisfies

$$\angle\Big(\mathrm{col}(\mathsf{X}_k), \mathrm{col}(\mathsf{V}_1)\Big) = \mathcal{O}\left(\left|\frac{\lambda_{p+1}}{\lambda_p}\right|^k\right).$$

Here, the angle between two subspaces $\mathcal{A}$ and $\mathcal{B}$ of $\mathbf{C}^n$ is defined as

$$\angle(\mathcal{A}, \mathcal{B}) = \max_{\boldsymbol{a}\in\mathcal{A}\setminus\{\mathbf{0}\}} \left(\min_{\boldsymbol{b}\in\mathcal{B}\setminus\{\mathbf{0}\}} \angle(\boldsymbol{a}, \boldsymbol{b}) \right).$$

6.3.2 The QR algorithm

The QR algorithm, which is based on the QR decomposition, is one of the most famous algorithms for calculating *all* the eigenpairs of a matrix. We first present the algorithm and then relate it to the simultaneous iteration in Section 6.3.1. The method is presented in Algorithm 12.

Algorithm 12 QR algorithm

$\mathsf{X}_0 = \mathsf{A}$
for $i \in \{1, 2, \dots\}$ **do**
  $\mathsf{Q}_k\mathsf{R}_k = \mathsf{X}_{k-1}$ (QR decomposition)
  $\mathsf{X}_k = \mathsf{R}_k\mathsf{Q}_k$
end for

Successive iterates of the QR algorithm are related by the equation

$$\mathsf{X}_k = \mathsf{Q}_k^{-1}\mathsf{X}_{k-1}\mathsf{Q}_k = \mathsf{Q}_k^*\mathsf{X}_{k-1}\mathsf{Q}_k = \cdots = (\mathsf{Q}_1 \dots \mathsf{Q}_k)^*\mathsf{X}_0(\mathsf{Q}_1 \dots \mathsf{Q}_k) \tag{6.11}$$

Therefore, all the iterates are related by a unitary similarity transformation, and so they all have the same eigenvalues as $\mathsf{X}_0 = \mathsf{A}$. Rearranging (6.11), we have

$$(\mathsf{Q}_1 \dots \mathsf{Q}_k)\mathsf{X}_k = \mathsf{A}(\mathsf{Q}_1 \dots \mathsf{Q}_k),$$

and so, introducing $\widetilde{\mathsf{Q}}_k = \mathsf{Q}_1 \dots \mathsf{Q}_k$ and noting that $\mathsf{X}_k = \mathsf{Q}_{k+1}\mathsf{R}_{k+1}$ by the algorithm, we deduce

$$\widetilde{\mathsf{Q}}_{k+1}\mathsf{R}_{k+1} = \mathsf{A}\widetilde{\mathsf{Q}}_k.$$

This reveals that the matrix sequence $(\widetilde{\mathsf{Q}}_k)_{k\geqslant 1}$ undergoes a simultaneous iteration and so, assuming that A is Hermitian with n distinct nonzero eigenvalues, we deduce that $\widetilde{\mathsf{Q}}_k \to \mathsf{V}$ essentially in the limit as $k \to \infty$, by Theorem 6.5. As a consequence, by (6.11), it holds that $\mathsf{X}_k \to \mathsf{V}^*\mathsf{X}_0\mathsf{V} = \mathsf{D}$; in other words, the matrix X_k converges to a diagonal matrix with the eigenvalues of A on the diagonal.

6.4 Projection methods

In this section, we begin by presenting a general method for constructing an approximation of the eigenvectors of A in a given subspace $\mathcal{U}$ of $\mathbf{C}^n$. We then discuss a particular choice for the subspace $\mathcal{U}$ as a Krylov subspace, which is very useful in practice.

Assume that $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_p\}$ is an orthonormal basis of $\mathcal{U}$. Then for any vector $\boldsymbol{v} \in \mathbf{C}^n$, the vector of $\mathcal{U}$ that is closest to $\boldsymbol{v}$ in the Euclidean distance is given by the orthogonal projection

$$\mathsf{P}_{\mathcal{U}}\boldsymbol{v} := \mathsf{U}\mathsf{U}^*\boldsymbol{v} = (\boldsymbol{u}_1\boldsymbol{u}_1^* + \cdots + \boldsymbol{u}_p\boldsymbol{u}_p^*)\boldsymbol{v}.$$

In practice, the eigenvectors of A are unknown, and so it is impossible to calculate approximations using this formula. The Rayleigh–Ritz method, which we present hereafter, is an alternative and practical method for constructing approximations of the eigenvectors and eigenvalues. In general, the subspace $\mathcal{U}$ does not contain any eigenvector of A, and so the problem

$$\mathsf{A}\boldsymbol{v} = \lambda\boldsymbol{v}, \qquad \boldsymbol{v} \in \mathcal{U} \tag{6.12}$$

does not admit a solution. Let us denote by U the matrix with columns $\boldsymbol{u}_1, \dots, \boldsymbol{u}_p$. Since any vector $\boldsymbol{v} \in \mathcal{U}$ is equal to $\mathsf{U}\boldsymbol{z}$ for some vector $\boldsymbol{z} \in \mathbf{C}^p$, equation (6.12) is equivalent to the problem

$$\mathsf{A}\mathsf{U}\boldsymbol{z} = \lambda\mathsf{U}\boldsymbol{z},$$

which is a system of n equations with $p < n$ unknowns. The Rayleigh–Ritz method is based on the idea that, in order to obtain a problem with as many unknowns as there are equations, we can multiply this equation by U^*, which leads to the problem

$$\mathsf{B}\boldsymbol{z} := (\mathsf{U}^*\mathsf{A}\mathsf{U})\boldsymbol{z} = \lambda\boldsymbol{z}. \tag{6.13}$$

This is standard eigenvalue problem for the matrix $\mathsf{U}^*\mathsf{A}\mathsf{U} \in \mathbf{C}^{p\times p}$, which is much easier to solve than the original problem if $p \ll n$. Equivalently, equation (6.13) may be formulated as follows: find $\boldsymbol{v} \in \mathcal{U}$ such that

$$\boldsymbol{u}^*(\mathsf{A}\boldsymbol{v} - \lambda\boldsymbol{v}), \qquad \forall \boldsymbol{u} \in \mathcal{U}. \tag{6.14}$$

The solutions to (6.13) and (6.14) are related by the equation $\boldsymbol{v} = \mathsf{U}\boldsymbol{z}$. Of course, the eigenvalues of B in problem (6.13), which are called the Ritz values of A relative to $\mathcal{U}$, are in general different from those of A. Once an eigenvector $\boldsymbol{y}$ of B has been calculated, an approximate eigenvector of A, called a *Ritz vector* of A relative to $\mathcal{U}$, is obtained from the equation $\widehat{\boldsymbol{v}} = \mathsf{U}\boldsymbol{y}$. The Rayleigh–Ritz algorithm is presented in full in Algorithm 13.

Algorithm 13 Rayleigh–Ritz

Choose $\mathcal{U} \subset \mathbf{C}^n$
Construct a matrix U whose columns are orthonormal and span $\mathcal{U}$
Find the eigenvalues $\widehat{\lambda}_i$ and eigenvectors $\boldsymbol{y}_i \in \mathbf{C}^p$ of $\mathsf{B} := \mathsf{U}^*\mathsf{A}\mathsf{U}$
Calculate the corresponding Ritz vectors $\widehat{\boldsymbol{v}}_i = \mathsf{U}\boldsymbol{y}_i \in \mathbf{C}^n$.

It is clear that if $\boldsymbol{v}_i \in \mathcal{U}$, then λ_i is an eigenvalue of B in (6.13). In fact, we can show the following more general statement.

Proposition 6.6. *If $\mathcal{U}$ is an invariant subspace of A, meaning that $\mathsf{A}\mathcal{U} \subset \mathcal{U}$, then each Ritz vector of A relative to $\mathcal{U}$ is an eigenvector of A.*

Proof. Let $\mathsf{U} \in \mathbf{C}^{n\times p}$ and $\mathsf{W} \in \mathbf{C}^{n\times(n-p)}$ be matrices whose columns form orthonormal bases of $\mathcal{U}$ and $\mathcal{U}^\perp$, respectively. Here $\mathcal{U}^\perp$ denotes the orthogonal complement of $\mathcal{U}$ with respect to the Euclidean inner product. Then, since $\mathsf{W}^*\mathsf{A}\mathsf{U} = \mathsf{0}$ by assumption, it holds that

$$\mathsf{Q}^*\mathsf{A}\mathsf{Q} = \begin{pmatrix} \mathsf{U}^*\mathsf{A}\mathsf{U} & \mathsf{U}^*\mathsf{A}\mathsf{W} \\ \mathsf{W}^*\mathsf{A}\mathsf{U} & \mathsf{W}^*\mathsf{A}\mathsf{W} \end{pmatrix} = \begin{pmatrix} \mathsf{U}^*\mathsf{A}\mathsf{U} & \mathsf{U}^*\mathsf{A}\mathsf{W} \\ \mathsf{0} & \mathsf{W}^*\mathsf{A}\mathsf{W} \end{pmatrix}, \qquad \mathsf{Q} = \begin{pmatrix} \mathsf{U} & \mathsf{W} \end{pmatrix}.$$

If $(\boldsymbol{y}, \widehat{\lambda})$ is an eigenvector of $\mathsf{U}^*\mathsf{A}\mathsf{U}$, then

$$\mathsf{Q}^*\mathsf{A}\mathsf{Q}\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix} = \begin{pmatrix} (\mathsf{U}^*\mathsf{A}\mathsf{U})\boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix} = \widehat{\lambda}\begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{pmatrix} =: \widehat{\lambda}\boldsymbol{x},$$

and so $(\boldsymbol{x}, \widehat{\lambda})$ is an eigenpair of $\mathsf{Q}^*\mathsf{A}\mathsf{Q}$. But then $(\mathsf{Q}\boldsymbol{x}, \widehat{\lambda}) = (\mathsf{U}\boldsymbol{y}, \widehat{\lambda})$ is an eigenpair of A, which proves the statement. □

If $\mathcal{U}$ is close to being an invariant subspace of A, then it is expected that the Ritz vectors and Ritz values of A relative to $\mathcal{U}$ will provide good approximations of some of the eigenpairs of A. Quantifying this approximation is difficult, so we only present without proof the following error bound. See [10] for more information.

Proposition 6.7. *Let A be a full rank Hermitian matrix and $\mathcal{U}$ a p-dimensional subspace of $\mathbf{C}^n$. Then there exists eigenvalues $\lambda_{i_1}, \ldots, \lambda_{i_p}$ of A which satisfy*

$$\forall j \in \{1, \ldots, p\}, \qquad |\lambda_{i_j} - \widehat{\lambda}_j| \leqslant \|(\mathsf{I} - \mathsf{P}_\mathcal{U})\mathsf{A}\mathsf{P}_\mathcal{U}\|_2.$$

In the case where A is Hermitian, it is possible to show that the Ritz values are bounded from above by the eigenvalues of A. The proof of this result relies on the Courant–Fisher theorem for characterizing the eigenvalues of a Hermitian matrix, which is recalled in Theorem A.5 in the appendix.

Proposition 6.8. *If* $\mathsf{A} \in \mathbf{C}^{n\times n}$ *is Hermitian, then*

$$\forall i \in \{1,\dots,p\}, \qquad \widehat{\lambda}_i \leqslant \lambda_i$$

Proof. By the Courant–Fisher theorem, it holds that

$$\widehat{\lambda}_i = \max_{\mathcal{S}\subset\mathbf{C}^p,\dim(\mathcal{S})=i} \left(\min_{\boldsymbol{x}\in\mathcal{S}\setminus\{0\}} \frac{\boldsymbol{x}^*\mathsf{B}\boldsymbol{x}}{\boldsymbol{x}^*\boldsymbol{x}} \right)$$

Letting $\boldsymbol{y} = \mathsf{U}\boldsymbol{x}$ and then $\mathcal{R} = \mathsf{U}\mathcal{S}$, we deduce that

$$\begin{aligned}\widehat{\lambda}_i &= \max_{\mathcal{S}\subset\mathbf{C}^p,\dim(\mathcal{S})=i} \left(\min_{\boldsymbol{y}\in\mathsf{U}\mathcal{S}\setminus\{0\}} \frac{\boldsymbol{y}^*\mathsf{A}\boldsymbol{y}}{\boldsymbol{y}^*\boldsymbol{y}} \right) \\ &= \max_{\mathcal{R}\subset\mathcal{U},\dim(\mathcal{R})=i} \left(\min_{\boldsymbol{y}\in\mathcal{R}\setminus\{0\}} \frac{\boldsymbol{y}^*\mathsf{A}\boldsymbol{y}}{\boldsymbol{y}^*\boldsymbol{y}} \right) \leqslant \max_{\mathcal{R}\subset\mathbf{C}^n,\dim(\mathcal{R})=i} \left(\min_{\boldsymbol{y}\in\mathcal{R}\setminus\{0\}} \frac{\boldsymbol{y}^*\mathsf{A}\boldsymbol{y}}{\boldsymbol{y}^*\boldsymbol{y}} \right) = \lambda_i,\end{aligned}$$

where we used the Courant–Fisher for the matrix A in the last equality. □

This projection approach is sometimes combined with a simultaneous subspace iteration: an approximation X_k of the p first eigenvector is first calculated using Algorithm 11, and then the matrix X_k is used in place of U in Algorithm 13.

6.4.1 Projection method in a Krylov subspace

The power iteration constructs at iteration k an approximation of $\boldsymbol{v}_1$ in the one-dimensional subspace spanned by the vector $\mathsf{A}^k\boldsymbol{x}_0$, and only the previous iteration $\boldsymbol{x}_k$ is employed to construct $\boldsymbol{x}^{k+1}$. One may wonder whether, by employing all the previous iterates rather than only the previous one, a better approximation of $\boldsymbol{v}_1$ can be constructed. More precisely, instead of looking for an approximation in the subspace $\mathrm{Span}\{\mathsf{A}^k\boldsymbol{x}_0\}$, would it be useful to extend the search area to the Krylov subspace

$$\mathcal{K}_{k+1}(\mathsf{A},\boldsymbol{x}_0) := \mathrm{Span}\big\{\boldsymbol{x}_0, \mathsf{A}\boldsymbol{x}_0, \dots, \mathsf{A}^k\boldsymbol{x}_0\big\}?$$

The answer to this question is positive, and the resulting method is often much faster than the power iteration. This is achieved by employing the Rayleigh–Ritz projection method Algorithm 13 with the choice $\mathcal{U} = \mathcal{K}_{k+1}(\mathsf{A},\boldsymbol{x}_0)$. Applying this method requires to calculate an orthonormal basis of the Krylov subspace and to calculate the reduced matrix $\mathsf{U}^*\mathsf{A}\mathsf{U}$. The *Arnoldi method* enables to achieve these two goals simultaneously.

6.4.2 The Arnoldi iteration

This Arnoldi iteration is based on the Gram–Schmidt process and presented in Algorithm 14. The iteration breaks down if $h_{j+1,j} = 0$, which indicates that $\mathcal{A}\boldsymbol{u}_j$ belongs to the Krylov subspace $\mathrm{Span}\{\boldsymbol{u}_1,\dots,\boldsymbol{u}_j\} = \mathcal{K}_j(\mathsf{A},\boldsymbol{u}_1)$, implying that $\mathcal{K}_{j+1}(\mathsf{A},\boldsymbol{u}_1) = \mathcal{K}_j(\mathsf{A},\boldsymbol{u}_1)$. In this case, the subspace $\mathcal{K}_j(\mathsf{A},\boldsymbol{u}_1)$ is an invariant subspace of A because, by Exercise 6.2, we have

$$\mathsf{A}\mathcal{K}_j(\mathsf{A},\boldsymbol{u}_1) \subset \mathcal{K}_{j+1}(\mathsf{A},\boldsymbol{u}_1) = \mathcal{K}_j(\mathsf{A},\boldsymbol{u}_1).$$

Algorithm 14 Arnoldi iteration for constructing an orthonormal basis of $\mathcal{K}_p(\mathsf{A}, \boldsymbol{u}_1)$

Choose $\boldsymbol{u}_1$ with unit norm.
for $j \in \{1, \dots p\}$ **do**
  $\boldsymbol{u}_{j+1} \leftarrow \mathsf{A}\boldsymbol{u}_j$
  for $i \in \{1, \dots, j\}$ **do**
    $h_{i,j} \leftarrow \boldsymbol{u}_i^* \boldsymbol{u}_{j+1}$
    $\boldsymbol{u}_{j+1} \leftarrow \boldsymbol{u}_{j+1} - h_{i,j}\boldsymbol{u}_i$
  end for
  $h_{j+1,j} \leftarrow \|\boldsymbol{u}_{j+1}\|$
  $\boldsymbol{u}_{j+1} \leftarrow \boldsymbol{u}_{j+1}/h_{j+1,j}$
end for

Therefore, applying the Rayleigh–Ritz with $\mathcal{U} = \mathrm{Span}\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_j\}$ yields exact eigenpairs Proposition 6.6. If the iteration does not break down then, by construction, the vectors $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_p\}$ at the end of the algorithm are orthonormal. It is also simple to show by induction that they form a basis of $\mathcal{K}_p(\mathsf{A}, \boldsymbol{u}_1)$. The scalar coefficients $h_{i,j}$ can be collected in a matrix square $p \times p$ matrix

$$\mathsf{H} = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,p} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,p} \\ 0 & h_{3,2} & h_{3,3} & \cdots & h_{3,p} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{p,p-1} & h_{p,p} \end{pmatrix}.$$

This matrix contains only zeros under the first subdiagonal; such a matrix is called a *Hessenberg* matrix. Inspecting the algorithm, we notice that the j-th column contains the coefficients of the projection of the vector $\mathsf{A}\boldsymbol{u}_j$ onto the basis $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_p\}$. In other words,

$$\mathsf{U}^*\mathsf{A}\mathsf{U} = \mathsf{H}, \tag{6.15}$$

We have thus shown that the Arnoldi algorithm enables to construct both an orthonormal basis of a Krylov subspace and the associated reduced matrix. In fact, we have the following equation

$$\mathsf{A}\mathsf{U} = \mathsf{U}\mathsf{H} + h_{p+1,p}(\boldsymbol{v}_{p+1}\boldsymbol{e}_p^*), \qquad \boldsymbol{e}_p = \begin{pmatrix} 0 \\ \vdots \\ 1 \end{pmatrix} \in \mathbf{C}^p. \tag{6.16}$$

The Arnoldi algorithm, coupled with the Rayleigh–Ritz method, has very good convergence properties in the limit as $p \to \infty$, in particular for eigenvalues with a large modulus. The following result shows that the residual $\boldsymbol{r} = \mathsf{A}\widehat{\boldsymbol{v}} - \widehat{\lambda}\boldsymbol{v}$ associated with a Ritz vector can be estimated inexpensively. Specifically, the norm of the residual is equal to the last component of the associated eigenvector of H multiplied by $h_{p+1,p}$.

Proposition 6.9 (Formula for the residual). *Let $\boldsymbol{y}_i$ be an eigenvector of H associated with*

the eigenvalues $\widehat{\lambda}_i$, and let $\widehat{\boldsymbol{v}}_i = \mathsf{U}\boldsymbol{y}_i$ denote the corresponding eigenvector. Then

$$\mathsf{A}\widehat{\boldsymbol{v}}_i - \widehat{\lambda}\boldsymbol{v}_i = h_{p+1,p}(\boldsymbol{y}_i)_p \boldsymbol{v}_{p+1}.$$

Consequently, it holds that

$$\|\mathsf{A}\widehat{\boldsymbol{v}}_i - \widehat{\lambda}\boldsymbol{v}_i\| = |h_{p+1,p}(\boldsymbol{y}_i)_p|.$$

Proof. Multiplying both sides of (6.16) by $\boldsymbol{y}_i$, we obtain

$$\mathsf{A}\mathsf{U}\boldsymbol{y}_i = \mathsf{U}\mathsf{H}\boldsymbol{y}_i + h_{p+1,p}(\boldsymbol{v}_{p+1}\boldsymbol{e}_p^*)\boldsymbol{y}_i.$$

Using the definition of $\widehat{\boldsymbol{v}}_i$ and rearranging the equation, we have

$$\mathsf{A}\widehat{\boldsymbol{v}}_i - \widehat{\lambda}_i\boldsymbol{y}_i = h_{p+1,p}(\boldsymbol{v}_{p+1}\boldsymbol{e}_p^*)\boldsymbol{y}_i,$$

which immediately gives the result. □

In practice, the larger the dimension p of the subspace $\mathcal{U}$ employed in the Rayleigh–Ritz method, the more memory is required for storing an orthonormal basis of $\mathcal{U}$. In addition, for large values of p, computing the reduced matrix (6.15) and its eigenpairs becomes computationally expensive; the computational cost of computing the matrix H scales as $\mathcal{O}(p^2)$. To remedy these potential issues, the algorithm can be restarted periodically. For example, Algorithm 15 can be employed as an alternative to the power iteration in order to find the eigenvector associated with the eigenvalue with largest modulus.

Algorithm 15 Restarted Arnoldi iteration

Choose $\boldsymbol{u}_1 \in \mathbf{C}^n$ and $p \ll n$
for $i \in \{1, 2, \dots\}$ **do**
  Perform p iterations of the Arnoldi iteration and construct $\mathcal{U}$;
  Calculate the Ritz vector $\widehat{\boldsymbol{v}}_1$ associated with the largest Ritz value relative to $\mathcal{U}$;
  If this vector is sufficiently accurate, then stop. Otherwise, restart with $\boldsymbol{u}_1 = \widehat{\boldsymbol{v}}_1$.
end for

6.4.3 The Lanczos iteration

The Lanczos iteration may be viewed as a simplified version of the Arnoldi iteration in the case where the matrix A is Hermitian. Let us denote by $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_p\}$ the orthonormal vectors generated by the Arnoldi iteration. When A is Hermitian, it holds that

$$h_{i,j} = \boldsymbol{u}_i^*(\mathsf{A}\boldsymbol{u}_j) = (\mathsf{A}\boldsymbol{u}_i)^*\boldsymbol{u}_j = \overline{h_{j,i}}.$$

Therefore, the matrix H is Hermitian. This is not surprising, since we showed that $\mathsf{H} = \mathsf{U}^*\mathsf{A}\mathsf{U}$ and the matrix A is Hermitian. Since H is also of Hessenberg form, we deduce that H is tridiagonal. An inspection of Algorithm 14 shows that the subdiagonal entries of H are real. Since A is Hermitian, the diagonal entries $h_{i,i} = \boldsymbol{u}_i^*(\mathsf{A}\boldsymbol{u}_j)$ are also real, and so we conclude that

all the entries of the matrix H are in fact real. This matrix if of the form

$$\mathsf{H} = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_p \\ & & & \beta_p & \alpha_p \end{pmatrix}$$

Adapting the Arnoldi iteration to this setting leads to Algorithm 16.

Algorithm 16 Lanczos iteration for constructing an orthonormal basis of $\mathcal{K}_p(\mathsf{A}, \boldsymbol{u}_1)$

Choose $\boldsymbol{u}_1$ with unit norm.
$\beta_1 \leftarrow 0$, $\boldsymbol{u}_0 \leftarrow \mathbf{0} \in \mathbf{C}^n$
for $j \in \{1, \dots p\}$ **do**
  $\boldsymbol{u}_{j+1} \leftarrow \mathsf{A}\boldsymbol{u}_j - \beta_j \boldsymbol{u}_{j-1}$
  $\alpha_j \leftarrow \boldsymbol{u}_j^* \boldsymbol{u}_{j+1}$
  $\boldsymbol{u}_{j+1} \leftarrow \boldsymbol{u}_{j+1} - \alpha_j \boldsymbol{u}_j$
  $\beta_{j+1} \leftarrow \|\boldsymbol{u}_{j+1}\|$
  $\boldsymbol{u}_{j+1} \leftarrow \boldsymbol{u}_{j+1}/\beta_{j+1}$
end for

6.5 Exercises

Exercise 6.1. *PageRank is an algorithm for assigning a rank to the vertices of a directed graph. It is used by many search engines, notably Google, for sorting search results. In this context, the directed graph encodes the links between pages of the World Wide Web: the vertices of the directed graph are webpages, and there is an edge going from page i to page j if page i contains a hyperlink to page j.*

Let us consider a directed graph $G(V, E)$ with vertices $V = \{1, \dots, n\}$ and edges E. The graph can be represented by its adjacency matrix $\mathsf{A} \in \{0, 1\}^{n \times n}$, whose entries are given by

$$a_{ij} = \begin{cases} 1 & \text{if there is an edge from } i \text{ to } j, \\ 0 & \text{otherwise.} \end{cases}$$

Let r_i denote the "value" assigned to vertex i. The idea of PageRank, in its simplest form, is to assign values to the vertices by solving the following system of equations;

$$\forall i \in V, \qquad r_i = \sum_{j \in \mathcal{N}(i)} \frac{r_j}{o_j}. \tag{6.17}$$

where o_j is the outdegree of vertex j, i.e. the number of edges leaving from j. Here the sum is over the set of nodes $\mathcal{N}(i)$, which denotes all the "incoming" neighbors of i, i.e. those that have an edge pointing towards node i.

- *Read the Wikipedia page on* PageRank *to familiarize yourself with the algorithm.*

- *Let* $\boldsymbol{r} = \begin{pmatrix} r_1 & \dots & r_n \end{pmatrix}^T$. *Show using* (6.17) *that* $\boldsymbol{r}$ *satisfies*

$$\boldsymbol{r} = \mathsf{A}^T \begin{pmatrix} \frac{1}{o_1} & & \\ & \ddots & \\ & & \frac{1}{o_n} \end{pmatrix} \boldsymbol{r} =: \mathsf{A}^T \mathsf{O}^{-1} \boldsymbol{r}.$$

  In other words, $\boldsymbol{r}$ *is an eigenvector with eigenvalue* 1 *of the matrix* $\mathsf{M} = \mathsf{A}^T \mathsf{O}^{-1}$.

- *Show that* M *is a left-stochastic matrix, i.e. that each column sums to 1.*

- *Prove that the eigenvalues of any matrix* $\mathsf{B} \in \mathbf{R}^{n \times n}$ *coincide with those of* B^T. *You may use the fact that* $\det(\mathsf{B}) = \det(\mathsf{B}^T)$.

- *Using the previous items, show that 1 is an eigenvalue and that* $\rho(\mathsf{M}) = 1$. *For the second part, find a subordinate matrix norm such that* $\|\mathsf{M}\| = 1$.

- *Implement PageRank in order to rank pages from a 2013 snapshot of English Wikipedia. You can use either the simplified version of the algorithm given in* (6.17) *or the improved version with a damping factor described on Wikipedia. In the former case, the following are both sensible stopping criteria:*

$$\frac{\|\mathsf{M}\widehat{\boldsymbol{r}} - \widehat{\boldsymbol{r}}\|_1}{\|\widehat{\boldsymbol{r}}\|_1} < 10^{-15} \qquad \textit{or} \qquad \frac{\|\mathsf{M}\widehat{\boldsymbol{r}} - \widehat{\lambda}\widehat{\boldsymbol{r}}\|_1}{\|\boldsymbol{r}\|_1} < 10^{-15}, \qquad \widehat{\lambda} = \frac{\widehat{\boldsymbol{r}}^T \mathsf{M} \widehat{\boldsymbol{r}}}{\widehat{\boldsymbol{r}}^T \widehat{\boldsymbol{r}}},$$

  where $\widehat{\boldsymbol{v}}$ *is an approximation of the eigenvector corresponding to the dominant eigenvalue. A dataset is available on the course website to complete this part. This dataset contains a subset of the data publicly available here, and was generated from the full dataset by retaining only the 5% best rated articles. After decompressing the archive, you can load the dataset into Julia by using the following commands:*

```
import CSV
import DataFrames

# Data (nodes and edges)
nodes = CSV.read("names.csv", DataFrames.DataFrame)
edges = CSV.read("edges.csv", DataFrames.DataFrame)

# Convert data to matrices
nodes = Matrix(nodes)
edges = Matrix(edges)
```

  After you have assigned a rank to all the pages, print the 10 pages with the highest ranks. My code returns the following entries:

1. *United States*
2. *United Kingdom*
3. *World War II*
4. *Latin*
5. *France*
6. *Germany*
7. *English language*
8. *China*
9. *Canada*
10. *India*

- ***Extra credit:*** *Write a function* `search(keyword)` *that can be employed for searching the database. Here is an example of what it could return:*

```
julia> search("New York")
481-element Vector{String}:
 "New York City"
 "New York"
 "The New York Times"
 "New York Stock Exchange"
 "New York University"
 …
```

Exercise 6.2. *Show the following properties of the Krylov subspace* $\mathcal{K}_p(\mathsf{A}, \boldsymbol{x})$.

- $\mathcal{K}_p(\mathsf{A}, \boldsymbol{x}) \subset \mathcal{K}_{p+1}(\mathsf{A}, \boldsymbol{x})$.
- $\mathsf{A}\mathcal{K}_p(\mathsf{A}, \boldsymbol{x}) \subset \mathcal{K}_{p+1}(\mathsf{A}, \boldsymbol{x})$.
- *The Krylov subspace* $\mathcal{K}_p(\mathsf{A}, \boldsymbol{x})$ *is invariant under rescaling: for all* $\alpha \in \mathbf{C}$,

$$\mathcal{K}_p(\mathsf{A}, \boldsymbol{x}) = \mathcal{K}_p(\alpha\mathsf{A}, \boldsymbol{x}) = \mathcal{K}_p(\mathsf{A}, \alpha\boldsymbol{x}).$$

- *The Krylov subspace* $\mathcal{K}_p(\mathsf{A}, \boldsymbol{x})$ *is invariant under shift of the matrix* A*: for all* $\alpha \in \mathbf{C}$,

$$\mathcal{K}_p(\mathsf{A}, \boldsymbol{x}) = \mathcal{K}_p(\mathsf{A} - \alpha\mathsf{I}, \boldsymbol{x}).$$

- *Similarity transformation: If* $\mathsf{T} \in \mathbf{C}^{n\times n}$ *is nonsingular, then*

$$\mathcal{K}_p(\mathsf{T}^{-1}\mathsf{A}\mathsf{T}, \mathsf{T}^{-1}\boldsymbol{x}) = \mathsf{T}^{-1}\mathcal{K}_p(\mathsf{A}, \boldsymbol{x}).$$

Exercise 6.3. *The minimal polynomial of a matrix* $\mathsf{A} \in \mathbf{C}^{n\times n}$ *is the monic polynomial* p *of lowest degree such that* $p(\mathsf{A}) = 0$. *Prove that, if* A *is Hermitian with* $m \leqslant n$ *distinct eigenvalues, then the minimal polynomial is given by*

$$p(t) = \prod_{i=1}^{m}(t - \lambda_i).$$

Exercise 6.4. *The minimal polynomial for a general matrix* $\mathsf{A} \in \mathbf{C}^{n\times n}$ *is given by*

$$p(t) = \prod_{i=1}^{m}(t - \lambda_i)^{s_i}.$$

where s_i is the size of the largest Jordan block associated with the eigenvalue λ_i in the normal Jordan form of A. Verify that $p(\mathsf{A}) = 0$.

Exercise 6.5. *Let d denote the degree of the minimal polynomial of A. Show that*

$$\forall p \geqslant d, \qquad \mathcal{K}_{p+1}(\mathsf{A}, \boldsymbol{x}) = \mathcal{K}_p(\mathsf{A}, \boldsymbol{x}).$$

Deduce that, for $p \geqslant n$, the subspace $\mathcal{K}_p(\mathsf{A}, \boldsymbol{x})$ is an invariant subspace of A.

Exercise 6.6. *Let $\mathsf{A} \in \mathbf{C}^{n\times n}$. Show that $\mathcal{K}_n(\mathsf{A}, \boldsymbol{x})$ is the smallest invariant subspace of A that contains $\boldsymbol{x}$.*

Exercise 6.7. *Consider the matrix*

$$\mathsf{M} = \begin{pmatrix} 0 & 1 & 2 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

- *Find the dominant eigenvalue of M by using the power iteration.*
- *Find the eigenvalue of M closest to 1 by using the inverse iteration.*
- *Find the other two eigenvalues of M by using a method of your choice.*

Exercise 6.8 (A posteriori error bound)**.** *Assume that $\mathsf{A} \in \mathbf{C}^{n\times n}$ is Hermitian, and that $\widehat{\boldsymbol{v}}$ is a normalized approximation of an eigenvector which satisfies*

$$\|\widehat{\boldsymbol{z}}\| := \|\mathsf{A}\widehat{\boldsymbol{v}} - \widehat{\lambda}\widehat{\boldsymbol{v}}\| = \delta, \qquad \widehat{\lambda} = \frac{\widehat{\boldsymbol{v}}^*\mathsf{A}\widehat{\boldsymbol{v}}}{\widehat{\boldsymbol{v}}^*\widehat{\boldsymbol{v}}}.$$

Prove that there is an eigenvalue λ of A such that

$$|\widehat{\lambda} - \lambda| \leqslant \delta.$$

Hint: *Consider first the case where A is diagonal.*

Exercise 6.9 (Bauer–Fike theorem)**.** *Assume that $\mathsf{A} \in \mathbf{C}^{n\times n}$ is diagonalizable: $\mathsf{AV} = \mathsf{VD}$. Show that, if $\widehat{\boldsymbol{v}}$ is a normalized approximation of an eigenvector which satisfies*

$$\|\boldsymbol{r}\| := \|\mathsf{A}\widehat{\boldsymbol{v}} - \widehat{\lambda}\widehat{\boldsymbol{v}}\| = \delta$$

for some $\widehat{\lambda} \in \mathbf{C}$, then there is an eigenvalue λ of A such that

$$|\widehat{\lambda} - \lambda| \leqslant \kappa_2(\mathsf{V})\delta.$$

Hint*: Rewrite*

$$\|\widehat{\boldsymbol{v}}\| = \|(\mathsf{A} - \widehat{\lambda}\mathsf{I})^{-1}\boldsymbol{r}\| = \|\mathsf{V}(\mathsf{D} - \widehat{\lambda}\mathsf{I})^{-1}\mathsf{V}^{-1}\boldsymbol{r})\|.$$

Exercise 6.10. *In Exercise 6.8 and Exercise 6.9, we saw examples* a posteriori *error estimates which guarantee the existence of an eigenvalue of* A *within a certain distance of the approximation* $\widehat{\lambda}$*. In this exercise, our aim is to provide an answer to the following different question: given an approximate eigenpair* $(\widehat{\boldsymbol{v}}, \widehat{\lambda})$*, what is the smallest perturbation* E *that we need to apply to* A *in order to guarantee that* $(\widehat{\boldsymbol{v}}, \widehat{\lambda})$ *is an exact eigenpair, i.e. that*

$$(\mathsf{A} + \mathsf{E})\widehat{\boldsymbol{v}} = \widehat{\lambda}\widehat{\boldsymbol{v}}?$$

Assume that $\widehat{\boldsymbol{v}}$ *is normalized and let* $\mathcal{E} = \left\{\mathsf{E} \in \mathbf{C}^{n\times n} : (\mathsf{A} + \mathsf{E})\widehat{\boldsymbol{v}} = \widehat{\lambda}\widehat{\boldsymbol{v}}\right\}$*. Prove that*

$$\min_{\mathsf{E}\in\mathcal{E}} \|\mathsf{E}\|_2 = \|\boldsymbol{r}\|_2 := \|\mathsf{A}\widehat{\boldsymbol{v}} - \widehat{\lambda}\widehat{\boldsymbol{v}}\|. \tag{6.18}$$

To this end, you may proceed as follows:

- *Show first that any* $\mathsf{E} \in \mathcal{E}$ *satisfies* $\mathsf{E}\widehat{\boldsymbol{v}} = -\boldsymbol{r}$*.*
- *Deduce from the first item that*
$$\inf_{\mathsf{E}\in\mathcal{E}} \|\mathsf{E}\|_2 \geqslant \|\boldsymbol{r}\|_2.$$
- *Find a rank one matrix* E_* *such that* $\|\mathsf{E}_*\|_2 = \|\boldsymbol{r}\|_2$*, and then conclude. Recall that any rank 1 matrix can be written in the form* $\mathsf{E}_* = \boldsymbol{u}\boldsymbol{w}^*$*, with norm* $\|\boldsymbol{u}\|_2\|\boldsymbol{w}\|_2$*.*

Equation (6.18) *is a simplified version of the Kahan–Parlett–Jiang theorem and is an example of a* backward error estimate*. Whereas forward error estimates quantify the distance between an approximation and the exact solution, backward error estimates give the size of the perturbation that must be applied to a problem so that an approximation is exact.*

Exercise 6.11. *Assume that* $(\boldsymbol{x}_k)_{k\geqslant 0}$ *is a sequence of normalized vectors in* $\mathbf{C}^n$*. Show that the following statements are equivalent:*

- $(\boldsymbol{x}_k)_{k\geqslant 0}$ *converges essentially to* $\boldsymbol{x}_\infty$ *in the limit as* $k \to \infty$*.*
- *The angle* $\angle(\boldsymbol{x}_k, \boldsymbol{x}_\infty)$ *converges to zero in the limit as* $k \to \infty$*.*
- *The projector* $\mathsf{P}_{\boldsymbol{x}_k}$ *converges to* $\mathsf{P}_{\boldsymbol{x}_\infty}$ *in the limit as* $k \to \infty$*.*

Exercise 6.12. *Assume that* $\mathsf{A} \in \mathbf{C}^{n\times n}$ *is skew-Hermitian. Derive a Lanczos-like algorithm for constructing an orthonormal basis of* $\mathcal{K}_p(\mathsf{A}, \boldsymbol{x})$ *and calculating the reduced matrix*

$$\mathsf{U}^*\mathsf{A}\mathsf{U},$$

where $\mathsf{U} \in \mathbf{C}^{n\times p}$ *contains the vectors of the basis as columns. Implement your algorithm with* $p = 20$ *in order to approximate the dominant eigenvalue of the matrix* A *constructed by the following piece of code:*

```
n = 5000
A = rand(n, n) + im * randn(n, n)
A = A - A'  # A is now skew-Hermitian
```

Exercise 6.13. *Assume that* $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_p\}$ *and* $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$ *are orthonormal bases of the same subspace* $\mathcal{U} \subset \mathbf{C}^n$*. Show that the projectors* $\mathsf{U}\mathsf{U}^*$ *and* $\mathsf{W}\mathsf{W}^*$ *are equal.*

Exercise 6.14. *Assume that* $\mathsf{A} \in \mathbf{C}^{n\times n}$ *is a Hermitian matrix with distinct eigenvalues, and suppose that we know the dominant eigenpair* $(\boldsymbol{v}_1, \lambda_1)$*, with* $\boldsymbol{v}_1$ *normalized. Let*

$$\mathsf{B} = \mathsf{A} - \lambda_1 \boldsymbol{v}_1 \boldsymbol{v}_1^*.$$

If we apply the power iteration to this matrix, what convergence can we expect?

Exercise 6.15. *Assume that* $\widehat{\boldsymbol{v}_1}$ *and* $\widehat{\boldsymbol{v}_2}$ *are two Ritz vectors of a Hermitian matrix* A *relative to a subspace* $\mathcal{U} \subset \mathbf{C}^n$*. Show that, if the associated Ritz values are distinct, then* $\widehat{\boldsymbol{v}_1}^* \widehat{\boldsymbol{v}_2} = 0$*.*

6.6 Discussion and bibliography

The content of this chapter is inspired mainly from [14] and also from [11]. The latter volume contains a detailed coverage of the standard methods for eigenvalue problems. Some of the exercises are taken from [16], and others are inspired from [11].