SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

An introduction to the QR-method

av

Jacob Westlund

2019 - No K2

An introduction to the QR-method

Jacob Westlund

Självständigt arbete i matematik 15 högskolepoäng, grundnivå

Handledare: Yishao Zhou

2019

An introduction to the QR-method

Jacob Westlund

January 2019

Abstract

This thesis aims to provide an introduction to the QR-method, which is one of the most widely used algorithm for computing eigenvalues of matrices. The thesis starts by introducing fundamental concepts about matrices and eigenvalues which are then used as a theoretical framework throughout the thesis. In order to provide the reader with an understanding and a proof for the convergence of the QR-method, we introduce the Power method and show that the QR-method is equivalent to applying the Power method to multiple columns at the same time. We also show how the basic QR-method can be improved upon by reducing the initial matrix to Hessenberg form and by introducing shifts. Finally, we present the implicit QR-method, which is similar to the algorithms used in practice for computing eigenvalues.

1 Introduction

A problem that arises often in scientific computing applications is that of computing eigenvalues and their corresponding eigenvectors. As modern information processing deals with increasingly large matrices, designing efficient algorithms and methods for computing eigenvalues and eigenvectors is therefore an important problem in numerical analysis.

Abel proved already in 1824 that no analogue of the quadratic formula can exist for polynomials of degree 5 or greater. This means that even if computation could be done in exact arithmetic any eigenvalue solver or procedure must use iterative processes for computing eigenvalues rather than a direct one. The goal of an efficient eigenvalue solver or algorithm is therefore to produce a sequences of numbers which rapidly converges towards the eigenvalues. [6] This thesis aims to give an introduction to one such algorithm, the QR-method, which one of the most widely used algorithm for computing eigenvalues and eigenvectors, and it has been called one of the 10 most important algorithms of the 20-th century. [4]

This paper is divided in to the following sections; Section 2 is dedicated towards providing an introduction to useful concepts surrounding eigenvalues and matrices, most of which will be familiar to any one who has taken a introductory course in linear algebra. In section 3 we introduce the Power method and show how it can be used to compute individual eigenvectors and eigenvalues. In section 4 we introduce the QR-method for computing all eigenvalues in a matrix and we relate the QR-method to the Power method in order to prove its convergence and provide an intuition for how it works. In the same section we also show how modifications to the QR-method can lead to improvements in the convergence rate and we finally present the practical and the implicit versions of QR-method which is similar to the algorithms used in practice.

Throughout the paper we will use the following notation. Matrices are typeset in an uppercase font: A, Q; vectors appear as boldface, lowercase letters: $\mathbf{a}$, $\mathbf{q}$; scalars are depicted as lowercase letters, e.g., the elements of matrices and vectors: $A = [a_{m,n}]_{m,n}$ and $\mathbf{v} = [v_n]_n$.

2 Eigenvalues of matrices and prerequisites

A familiar problem in linear algebra is computing eigenvalues $\lambda \in \mathbb{C}$ and their corresponding eigenvectors $\mathbf{v}$ of matrices. Eigenvalues and eigenvectors are commonly defined as follows.

Definition 2.1 (Eigenvalues and eigenvectors)**.** Given a matrix $A \in \mathbb{C}^{n\times n}$, a vector $\mathbf{v} \in \mathbb{C}^n$ where $\mathbf{v} \neq 0$ is called an eigenvector and $\lambda \in \mathbb{C}$ is its corresponding eigenvalue if

$$A\mathbf{v} = \lambda\mathbf{v}.$$

In a geometric sense the eigenvector $\mathbf{v}$ can be thought of as an vector that is scaled by a scalar λ (the eigenvalue) but keeps its direction when we apply a transformation A to $\mathbf{v}$. We will start this chapter by talking about eigenvalues and eigenvector as well as some properties of matrices that will later be useful when we present methods for computing eigenvalues.

Definition 2.2 (Characteristic polynomial)**.** The characteristic polynomial of a matrix $A \in \mathbb{C}^{n\times n}$ is the polynomial

$$p_A(z) = \det(A - zI) = z^n + c_{n-1}z^{n-1} + ... + c_1 z + c_0$$

where I is the $n \times n$ identity matrix.

Theorem 2.1 (Characteristic equation)**.** *λ is an eigenvalue of the matrix $A \in \mathbb{C}^{n\times n}$ if and only if $p_A(\lambda) = 0$. The equation $p_A(\lambda) = 0$ is known as the characteristic equation of A.*

Proof. The theorem follows from definition 2.1 of eigenvalues and eigenvectors as well as the equivalence of several expressions. Given that λ is an eigenvalue it follows that there exists a non-zero vector $\mathbf{v}$ such that

$$A\mathbf{v} = \lambda\mathbf{v}, \tag{1}$$

the opposite relation is of course also true. We can write the above expression (1) as

$$A\mathbf{v} - \lambda\mathbf{v} = 0 \text{ or } (A - \lambda I)\mathbf{v} = 0.$$

Since we know that $\mathbf{v}$ is a non-zero vector it means that $A - \lambda I$ must have linear dependent columns and thus no inverse matrix, it is *singular*. The opposite relation is again true. Finally the determinant $\det(A - \lambda I)$ must then be zero.

We can write the equivalences as follows

$$\begin{aligned}\lambda \text{ is an eigenvalue} &\Leftrightarrow A\mathbf{v} - \lambda\mathbf{v} = 0 \\ &\Leftrightarrow A - \lambda I \text{ does not have a inverse matrix, it is singular} \\ &\Leftrightarrow \det(A - \lambda I) = 0\end{aligned}$$

■

From the definition 2.2 we see that a $n \times n$ matrix will result in an n-th degree polynomial (and equation when solving for λ in Theorem 2.1). From the fundamental theorem of algebra we know that this means that every $n \times n$ matrix will have n roots λ_i, $i = 1, 2, 3, \ldots n$. Of course, the eigenvalues does not have to be distinct since the roots in a polynomial can have greater multiplicity than 1.

Example 2.1 (Eigenvalues of a small matrix with the characteristic equation)**.** Let us look at an example of how to calculate the eigenvalues of a small matrix. Let

$$A = \begin{bmatrix} 1 & 4 \\ 1 & 1 \end{bmatrix},$$

which give us the equation

$$\det(A - \lambda I) = \begin{vmatrix} 1-\lambda & 4 \\ 1 & 1-\lambda \end{vmatrix} = (1-\lambda)^2 - 4 = 0.$$

We get $1 - \lambda = 2$ and $1 - \lambda = -2$, leading to the eigenvalues $\lambda = -1$ or $\lambda = 3$.

Example 2.2 (Eigenvalues of a upper triangular matrix)**.** Let us look at another example, this time with an upper triangular matrix. Let

$$A = \begin{bmatrix} 1 & 4 & 4 \\ 0 & -2 & 1 \\ 0 & 0 & 8 \end{bmatrix},$$

which give us the equation

$$\det(A - \lambda I) = \begin{vmatrix} 1-\lambda & 4 & 4 \\ 0 & -2-\lambda & 1 \\ 0 & 0 & 8-\lambda \end{vmatrix} = (1-\lambda)(-2-\lambda)(8-\lambda) = 0.$$

We get the eigenvalues $\lambda = 1$, $\lambda = -2$ or $\lambda = 8$.

In example 2.2 the eigenvalues equals the diagonal entries of A. This is not an accident, triangular matrices are special in the sense that eigenvalues can be read directly from the diagonal of the matrix.

Theorem 2.2 (Eigenvalues of triangular matrix)**.** *The eigenvalues of an upper or lower triangular matrix $A \in \mathbb{C}^{n\times n}$ are the diagonal entries of the matrix.*

Proof. Let $A \in \mathbb{C}^{n\times n}$ be an upper triangular matrix, that is

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ 0 & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{n,n} \end{bmatrix}.$$

Consider the expression $\det(A - \lambda I)$ from the characteristic polynomial in 2.2, for matrix A this can be written as

$$\prod_{i=1}^{n} a_{i,i} - \lambda_i. \tag{2}$$

From theorem 2.1 we know that expression (2) has to be zero for there to be eigenvalues. The expression equals zero if and only if, $a_{i,i} = \lambda_i$ for some i. Thus the diagonal entries of the matrix equals the eigenvalues precisely. To show that the same applies to lower triangular matrices it is enough to show that the transpose of A, written as A^T, has the same eigenvalues as A,

$$A^T = \begin{bmatrix} a_{1,1} & 0 & \dots & 0 \\ a_{2,1} & a_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix}$$

is a lower triangular matrix. Since A and A^T have the same characteristic polynomial,

$$\det(A - \lambda I) = \det(A - \lambda I)^T = \det(A^T - \lambda I)$$

they must also have the same eigenvalues with the same multiplicities. ∎

We will now move on to show some important properties of matrices. For example, some matrices represent the same linear operator but in different basis. They therefore share important characteristics such as the same eigenvalues. Such matrices are called *similar* matrices and are defined as follows.

Definition 2.3 (Similar matrices)**.** Two matrices $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{n\times n}$ are said to be similar if there is a square non-singular matrix $S \in \mathbb{C}^{n\times n}$ such that

$$B = S^{-1}AS. \tag{3}$$

The transformation from B to A (or the opposite) is called a similarity transformation. It defines an equivalence transformation, characterised, among other things, by transitivity; if A is similar to B and B is similar to C, then A is similar to C.

Theorem 2.3 (Eigenvalues of similar matrices)**.** *If $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{n\times n}$, and B is similar to A, $B = S^{-1}AS$. Then A and B have the same characteristic polynomial and thus the same eigenvalues, with the same multiplicities. If $\mathbf{v}$ is an eigenvector of A, then $S^{-1}\mathbf{v}$ is an eigenvector of B for the same eigenvalue λ.*

Proof. Using the product rule for determinants, we find that the characteristic polynomial is the same

$$\begin{aligned}\det(B-\lambda I) &= \det(S^{-1}AS-\lambda I) = \det(S^{-1}(A-\lambda I)S)\\ &= \det(S^{-1})\det(A-\lambda I)\det(S) = \det(A-\lambda I).\end{aligned} \tag{4}$$

From the definition of eigenvalues and eigenvectors $A\mathbf{v} = \lambda\mathbf{v}$ (definition 2.1) and similar matrices $A = SBS^{-1}$ (definition 2.3) it follows that $SBS^{-1}\mathbf{v} = \lambda\mathbf{v}$ and by left multiplication of both sides by S^{-1} we get $BS^{-1}\mathbf{v} = S^{-1}\lambda\mathbf{v} = \lambda S^{-1}\mathbf{v}$. Thus if $\mathbf{v}$ is an eigenvector of A, with the eigenvalue λ, then $S^{-1}\mathbf{v}$ is an eigenvector of B with the same eigenvalue λ. ■

Previously we concluded in theorem 2.2 that if a matrix is triangular (or just diagonal) then the eigenvalues can be read off the diagonal entries. In conjunction with what we have concluded about similar matrices in theorem 2.3 this means that if B is triangular and if A and B are similar, they have the same eigenvalues, and the eigenvalues of A can be read from the diagonal of B. However, it is important to note that matrices with the same eigenvalues are not necessarily similar. We show both these statements with examples.

Example 2.3 (Eigenvalues of similar matrices)**.** The matrices $A = \begin{bmatrix}1 & 2\\ 2 & 1\end{bmatrix}$ and $B = \begin{bmatrix}-1 & 0\\ 2 & 3\end{bmatrix}$ are similar. We have $A = S^{-1}BS$, with $S = \begin{bmatrix}1 & -1\\ 0 & 1\end{bmatrix}$. The eigenvalues of both A and B are $\lambda_1 = 3$ and $\lambda_2 = -1$.

Example 2.4 (Matrices with the same eigenvalues does not imply similarity)**.** It is of course not the case that two matrices with the same eigenvalues need to be similar. Let us look at an example of this. Let $A = I$ and $B = \begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$. Both A and B have eigenvalue $\lambda = 1$ but they cannot be similar since $A = S^{-1}BS$ is never true. If it was true then $SAS^{-1} = B$, but with $A = I$ this would mean that $SS^{-1} = B$ and thus $B = I$ which is a contradiction and not the case.

Definition 2.4 (Orthogonal matrix)**.** A matrix $A \in \mathbb{R}^{n\times n}$ is said to be orthogonal if

$$AA^T = I \tag{5}$$

where A^T denotes the transpose of A and I is the identity matrix. This means that the matrix A has an inverse that is equal to its transpose, $A^T = A^{-1}$.

There exists a corresponding definition for matrices with complex entries, known as unitary matrices. To define a unitary matrix we first need to define the conjugate transpose of a matrix.

Definition 2.5 (Conjugate transpose)**.** The conjugate transpose $A^* \in \mathbb{C}^{n\times m}$ of a matrix $A \in \mathbb{C}^{m\times n}$ is defined as

$$A^* = \overline{A^T} \tag{6}$$

where $\overline{A}$ represents the element-by-element conjugation of A and A^T is the element-by-element transposition $a_{i,j}$ to $a_{j,i}$ for $1 \leq i \leq n$ and $1 \leq j \leq m$ of A. In other words the conjugate transpose A^* is obtained by taking the transpose and then the complex conjugate of each entry in A.

Example 2.5 (Conjugate transpose)**.** To show what we mean we will provide a basic example. Let

$$A = \begin{bmatrix} 2 & -3-i & 10 \\ 2i & 1 & 2-i \end{bmatrix}$$

then the conjugate transpose of A is

$$A^* = \begin{bmatrix} 2 & -2i \\ -3+i & 1 \\ 10 & 2+i \end{bmatrix}.$$

It then follows naturally to define unitary matrices.

Definition 2.6 (Unitary matrix)**.** A matrix $A \in \mathbb{C}^{n\times n}$ is said to be unitary if

$$AA^* = I \tag{7}$$

where A^* is the conjugate transpose and I is the identity matrix. In other words, matrix A has an inverse that is equal to its conjugate transpose, $A^* = A^{-1}$.

Definition 2.7 (Symmetric matrix)**.** A matrix $A \in \mathbb{R}^{n\times n}$ which is equal to its transpose $A^T \in \mathbb{R}^{n\times n}$, that is $A = A^T$, is called a symmetric matrix.

Just as was the case for orthogonal matrices (with complex analogue unitary matrices) there exists a complex analogue of symmetric matrices.

Definition 2.8 (Hermitian matrix)**.** A matrix $A \in \mathbb{C}^{n\times n}$ which is equal to its complex transpose $A^* \in \mathbb{C}^{n\times n}$, that is $A = A^*$, is called a Hermitian matrix.

Lemma 2.1 (Eigenvalues of a Hermitian matrices are real)**.** *All eigenvalues of a Hermitian matrix are real.*

Proof. Let A be Hermitian matrix with a non-zero eigenvector $\mathbf{v}$ corresponding to an eigenvalue λ. Then from defintition 2.1 we have $A\mathbf{v} = \lambda\mathbf{v}$. Multiplying both sides of this expression with the conjugate transpose of $\mathbf{v}$, $\mathbf{v}^*$ we get

$$\mathbf{v}^* A\mathbf{v} = \mathbf{v}^*\lambda\mathbf{v} = \lambda\mathbf{v}^*\mathbf{v}.$$

Furthermore since $A = A^*$ (Hermitian) we can write the left side as

$$\mathbf{v}^* A\mathbf{v} = \mathbf{v}^* A^*\mathbf{v} = (A\mathbf{v})^*\mathbf{v} = (\lambda\mathbf{v})^*\mathbf{v} = \mathbf{v}^*\lambda^*\mathbf{v} = \lambda^*\mathbf{v}^*\mathbf{v}.$$

Written as a single expression we get

$$\lambda\mathbf{v}^*\mathbf{v} = \mathbf{v}^*\lambda\mathbf{v} = \mathbf{v}^* A\mathbf{v} = \mathbf{v}^* A^*\mathbf{v} = (A\mathbf{v})^*\mathbf{v} = (\lambda\mathbf{v})^*\mathbf{v} = \mathbf{v}^*\lambda^*\mathbf{v} = \lambda^*\mathbf{v}^*\mathbf{v}$$

Since $\mathbf{v}$ is non-zero, $\mathbf{v}^*\mathbf{v}$ is non-zero as well and so since $\lambda^* = \lambda$ the eigenvalue λ has to be real. ■

Gathering from what we have learned so far, we will now present a decomposition which will be elementary for this thesis going forward. The decomposition is referred to as the Schur form or the Schur decomposition.

Theorem 2.4 (Schur's theorem)**.** *If matrix $A \in \mathbb{C}^{n\times n}$, then A can be expressed as*

$$A = VUV^* \tag{8}$$

where V is a unitary matrix and U is an upper triangular matrix. The expression in the equation is known as the Schur form of A or Schur decomposition. Since A and U are similar the eigenvalues of A are the diagonal entries of U.

Proof. A proof of the existence of the Schur decomposition for a matrix $A \in \mathbb{C}^{n\times n}$ can be done with induction. For $n = 1$ the Schur decomposition will always exist since we can simply chose $V = \begin{bmatrix}1\end{bmatrix}$. We then assume that the Schur decomposition exists for all $(n-1)\times(n-1)$ matrices with complex entries. Now to our induction step, for $n > 1$ let $\mathbf{x}$ be a normalised eigenvector of A for the eigenvalue λ. We then form an unitary matrix $V = \begin{bmatrix}\mathbf{x}, \mathbf{v}_2, \ldots, \mathbf{v}_n\end{bmatrix} = \begin{bmatrix}\mathbf{x} & W\end{bmatrix}$ with $\mathbf{x}$ as the first column and W for the rest $(n-1)$ columns. Since $AV = A\begin{bmatrix}\mathbf{x} & W\end{bmatrix}$ we can write

$$V^* AV = \begin{bmatrix}\mathbf{x}^* \\ W^*\end{bmatrix} A \begin{bmatrix}\mathbf{x} & W\end{bmatrix} = \begin{bmatrix}\mathbf{x}^* A\mathbf{x} & \mathbf{x}^* AW \\ W^* A\mathbf{x} & W^* AW\end{bmatrix}. \tag{9}$$

Since $A\mathbf{x} = \lambda\mathbf{x}$ (definition 2.1) and columns W are orthogonal to $\mathbf{x}$, $W\mathbf{x}^* = 0$, it follows that $\mathbf{x}^* A\mathbf{x} = \lambda$ and $W^* A\mathbf{x} = \lambda W^*\mathbf{x} = 0$. We can write

$$A = \left[\begin{array}{c|c} \lambda & \mathbf{b} \\ \hline 0 & \hat{A} \end{array}\right] \tag{10}$$

where $\hat{A} = W^* AW \in \mathbb{C}^{(n-1)\times(n-1)}$ and $\mathbf{b}$ is a $1 \times n-1$ vector. By the induction hypothesis there exists a Schur decomposition of $\hat{A} = \hat{V}\hat{U}\hat{V}^*$ with $\hat{U}$ triangular and $\hat{V}$ unitary. Finally we can then form a unitary $n \times n$ unitary matrix

$$\bar{V} = V \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & \hat{V} \end{array}\right]. \tag{11}$$

Using the matrices A and $\bar{V}$ from expressions (10) and (11) we can write

$$\bar{V}^* A\bar{V} = \left[\begin{array}{c|c} \lambda & \mathbf{b}\hat{V} \\ \hline 0 & \hat{U} \end{array}\right] = U, \tag{12}$$

with U triangular. We have found the Schur decomposition of A. That fact that A and U are similar with the same eigenvalues follows directly from theorem 2.3 and definition 2.6. ■

Since matrix A and the upper triangular matrix U in the Schur form of A are similar, the eigenvalues of A are the diagonal entries of U. This fact makes the Schur decomposition extremely useful when computing the eigenvalues of matrices, since if we can compute the Schur decomposition of a given matrix we automatically get its eigenvalues and avoid the problem of solving the characteristic equation, theorem 2.1. In the the next sections we will introduce methods for computing the Schur decomposition and the rest of this thesis will very much strive towards that goal.

Corollary 2.1 (Spectral decomposition for Hermitian matrices)**.** *Every Hermitian matrix is unitarily similar to a diagonal real matrix*

$$V^*AV = D = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n).$$

Furthermore the columnvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ of V are the eigenvectors of A corresponding to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. The Schur decomposition of a Hermitian matrix A is the same as its spectral decomposition.

3 The Power method

Before we introduce the QR-Method we will talk about another method for computing eigenvalues and eigenvectors known as the Power method. We will later use the Power method when we try to give an intuition for and prove the convergence of the QR-method.

3.1 The basic Power method

In the previous chapter we introduced the characteristic polynomial in definition 2.2 as a way to obtain eigenvalues. This is however not an option for large matrices. Suppose that we are interested in finding the largest eigenvalue (in absolute terms), known as the *dominant eigenvalue*, of a matrix A then the Power method is useful. In this section we will give an short overview of the Power method and a proof for its convergence. We start by giving a definition of the dominant eigenvalue and dominant eigenvector

Definition 3.1 (Dominant eigenvalue and dominant eigenvector)**.** Let $A \in \mathbb{C}^{n\times n}$ be a matrix with eigenvalues that can be ordered

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$$

then the distinct maximum eigenvalue λ_1 is called a dominant eigenvalue while its associated eigenvector $\mathbf{x}_1$ is called a dominant eigenvector.

From the definition it is clear that not every matrix will have a dominant eigenvalue, and then of course not a dominant eigenvector either.

Example 3.1 (Matrix with no dominant eigenvalue or eigenvector)**.** For instance the matrix

$$A = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}$$

have eigenvalues $\lambda_1 = -2$ and $\lambda_2 = 2$ and thus no dominant eigenvalue and eigenvector.

Example 3.2 (Dominant eigenvalue and eigenvector of a matrix)**.** Matrices do however often have a dominant eigenvalue and eigenvector. We have a matrix

$$A = \begin{bmatrix} -5 & -2 \\ 4 & 1 \end{bmatrix},$$

and using the characteristic equation from theorem 2.1 we can calculate the eigenvalues of A,

$$0 = \lambda^2 + 4\lambda + 3 = (\lambda + 1)(\lambda + 3).$$

The eigenvalues are $\lambda_1 = -3$ and $\lambda_2 = -1$ where λ_1 is the dominant one. The corresponding dominant eigenvectors (to λ_1) are on the form

$$\mathbf{x} = t \begin{bmatrix} -1 \\ 1 \end{bmatrix} \text{ where } t \neq 0.$$

Theorem 3.1 (Convergence of the Power method)**.** *If $A \in \mathbb{C}^{n\times n}$ is a diagonalisable matrix with a dominant eigenvalue (λ_1) then given an initial non-zero vector $\mathbf{v}^{(0)}$ the Power method forms a sequence*

$$\mathbf{v}^{(0)}, A\mathbf{v}^{(0)}, A^2\mathbf{v}^{(0)}, \ldots, A^k\mathbf{v}^{(0)}, \ldots$$

that approaches a multiple of the dominant eigenvector $\mathbf{x}_i$ of matrix A.

Remark (Non-zero initial vector)**.** In theory only initial vectors $\mathbf{v}^{(0)}$ that contain a component in the direction of the dominant eigenvector, i.e. not orthogonal to the dominant eigenvector, will approach a multiple of the dominant eigenvector when the power method is applied. However, due to rounding errors this is not a problem in practice for any initial vector $\mathbf{v}^{(0)}$ except the zero vector which is always orthogonal to all vectors of the same type (size), even when the Power method is applied. [1]

Remark (Switching signs)**.** It should also be noted that the approximation for the dominant eigenvector can switch sign at each k step. In example 3.3 we can see that this is the case when $\lambda_1 < 0$ (negative) since λ_1^k switches sign for at each k in (17). If $\lambda_1 > 0$ (positive) then the approximation for the dominant eigenvector will either be positive or negative at each k step.

Proof. For simplification we assumed that the matrix A is diagonalisable, that is matrix A has a complete set of eigenpairs $(\lambda_i, \mathbf{x}_i)$ and thus linearly independent eigenvectors $\mathbf{x}_i$, each associated with an eigenvalue. We can therefore express any vector $\mathbf{v} \in \mathbb{C}^n$ as a linear combination of the eigenvectors. If we choose an arbitrary initial vector $\mathbf{v}^{(0)} = \mathbf{v}$ we can write

$$\mathbf{v}^{(0)} = \alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 + \cdots + \alpha_n\mathbf{x}_n = \sum_{i=1}^{n} \alpha_i\mathbf{x}_i. \tag{13}$$

Given the initial vector $\mathbf{v}^{(0)}$ the Power method forms a sequence $\mathbf{v}^{(1)} = A\mathbf{v}^{(0)}, \mathbf{v}^{(2)} = A^2\mathbf{v}^{(0)}, \ldots$ if we keep multiplying the initial vector by A. Recursively we get

$$\mathbf{v}^{(k)} = A\mathbf{v}^{(k-1)} \text{ for } k = 1, 2, 3, \ldots. \tag{14}$$

We know that $A\mathbf{v} = \lambda\mathbf{v}$ from definition 2.1, then combined with what we just showed in (14) we get $A^k\mathbf{v} = \lambda^k\mathbf{v}$ for any power k. We can build on (13) with what we now know to write

$$\mathbf{v}^{(k)} = A^k\mathbf{v}^{(0)} = \sum_{i=1}^{n} \alpha_i A^k\mathbf{x}_i = \sum_{i=1}^{n} \alpha_i\lambda^k\mathbf{x}_i. \tag{15}$$

We can develop (15) further and write

$$\begin{aligned}\mathbf{v}^{(k)} = A^k\mathbf{v}^{(0)} &= \sum_{i=1}^{n}\alpha_i\lambda^k\mathbf{x}_i = \lambda_1^k\sum_{i=1}^{n}\alpha_i\left(\frac{\lambda_i}{\lambda_1}\right)^k\mathbf{x}_i \\ &= \lambda_1^k\left(\alpha_1\mathbf{x}_1 + \sum_{i=2}^{n}\left(\frac{\lambda_i}{\lambda_1}\right)^k\mathbf{x}_i\right).\end{aligned} \tag{16}$$

Since we assumed $|\lambda_1| > |\lambda_i|$ for $i = 2, 3, 4, \ldots$ the right term in the last expression of (16) will approach zero when $k \to \infty$. This implies that the approximation

$$A^k\mathbf{v}^{(0)} \approx \lambda_1^k\alpha_1\mathbf{x}_1,\ \alpha_1 \neq 0 \tag{17}$$

improves as k increases. In other words if we keep multiplying or initial vector $\mathbf{v}^{(0)}$ by A we get an increasingly accurate approximation of a vector that points in the direction of the eigenvector $\mathbf{x}_1$.

Since $\mathbf{x}_1$ is a dominant eigenvector, it follows that any scalar multiple of it is also a dominant eigenvector. Thus we have shown that $A^k\mathbf{v}^{(0)}$ approaches a multiple of the dominant eigenvector of A when $k \to \infty$. ■

Corollary 3.1. *The convergence of the eigenvector will be linear with a rate equal to* $|\lambda_2| \,/\, |\lambda_1|$ *since* $|\lambda_2| \,/\, |\lambda_1| \geq |\lambda_3| \,/\, |\lambda_1| \geq \cdots \geq |\lambda_n| \,/\, |\lambda_1|$*, [3] pp. 476.*

Example 3.3 (Approximating a dominant eigenvector)**.** We will use the same matrix A as in example 3.2. We choose an initial non-zero vector $\mathbf{v}^{(0)} = \begin{bmatrix}1\\1\end{bmatrix}$.

Using the Power method we then obtain the following sequence of approximations:

$$\begin{aligned}\mathbf{v}^{(1)} = A\mathbf{v}^{(0)} &= \begin{bmatrix}-5 & -2\\ 4, & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}-7\\5\end{bmatrix} = 5\begin{bmatrix}1.4\\1\end{bmatrix} \\ \mathbf{v}^{(2)} = A\mathbf{v}^{(1)} &= \begin{bmatrix}-5 & -2\\ 4, & 1\end{bmatrix}\begin{bmatrix}-7\\5\end{bmatrix} = \begin{bmatrix}25\\-23\end{bmatrix} = -23\begin{bmatrix}-1.087\\1\end{bmatrix} \\ \mathbf{v}^{(3)} = A\mathbf{v}^{(2)} &= \begin{bmatrix}-5 & -2\\ 4, & 1\end{bmatrix}\begin{bmatrix}25\\-23\end{bmatrix} = \begin{bmatrix}-79\\77\end{bmatrix} = 77\begin{bmatrix}-1.026\\1\end{bmatrix} \\ \mathbf{v}^{(4)} = A\mathbf{v}^{(3)} &= \begin{bmatrix}-5 & -2\\ 4, & 1\end{bmatrix}\begin{bmatrix}-79\\77\end{bmatrix} = \begin{bmatrix}241\\-239\end{bmatrix} = -239\begin{bmatrix}-1.008\\1\end{bmatrix}.\end{aligned}$$

We see that the approximation approach a scalar multiple of the vector $\begin{bmatrix}-1, 1\end{bmatrix}$, which we of course know from example 3.2 is a dominant eigenvector of A. We also note that the convergence towards the scaled eigenvector is very quick in this case since the absolute values of the eigenvalues $|\lambda_1| = 3$ and $|\lambda_2| = 1$ are far apart, and thus $|\lambda_2| \,/\, |\lambda_1|$ is small and the convergence is quick as stated in corollary 3.1.

In our proof of theorem 3.1 we showed that $A^k\mathbf{v}^{(0)}$ will approach $\lambda_1^k\alpha_1\mathbf{x}_1$ when $k \to \infty$, but since $A^k\mathbf{v}^{(0)} \to \infty$ when $|\lambda_1| > 1$ and $A^k\mathbf{v}^{(0)} \to 0$ when $|\lambda_1| < 0$ we will want to apply some scaling to avoid overflow or underflow (values that a computer cannot represent). This is also evident from example 3.3 where the matrix elements grows in magnitude for every iteration. Since every multiple of a eigenvector is also an eigenvector, the magnitude of the eigenvector is unimportant

and we can apply any convenient scaling. Commonly the approximation of the eigenvector $\mathbf{x}_1$ is normalised in every step. Of course it is a good idea to choose the initial non-zero vector $\mathbf{v}^{(0)}$ to be normalised as well.

We still have not covered how this approximation of the the dominant eigenvector can help us compute the dominant eigenvalue. For this we need the Rayleigh quotient.

Definition 3.2 (Rayleigh quotient)**.** Given a matrix $A \in \mathbb{C}^{n\times n}$ and a vector $\mathbf{v} \in \mathbb{C}^n$ where $\mathbf{v}^*\mathbf{v} \neq 0$, the quantity

$$r(\mathbf{v}) = \frac{\mathbf{v}^* A\mathbf{v}}{\mathbf{v}^*\mathbf{v}} \tag{18}$$

is known as a Rayleigh quotient. The Rayleigh quotient is the unique quantity r that minimises $\|A\mathbf{v} - r\mathbf{v}\|_2$. [3] pp. 465.

Theorem 3.2 (Determining an eigenvalue from an eigenvector)**.** *If the vector* $\mathbf{v}$ *is an eigenvector of a matrix* $A \in \mathbb{C}^{n\times n}$ *then the corresponding eigenvalue is given by*

$$\lambda = \frac{\mathbf{v}^* A\mathbf{v}}{\mathbf{v}^*\mathbf{v}} \tag{19}$$

Proof. Since $\mathbf{v}$ is an eigenvector of A, we have that $A\mathbf{v} = \lambda\mathbf{v}$ and thus

$$r(\mathbf{v}) = \frac{\mathbf{v}^* A\mathbf{v}}{\mathbf{v}^*\mathbf{v}} = \frac{\lambda\mathbf{v}^*\mathbf{v}}{\mathbf{v}^*\mathbf{v}} = \lambda \tag{20}$$

■

Corollary 3.2. *The convergence rate of the eigenvalue* λ_1 *in the Power method is the same as for the eigenvector in corollary 3.1 in the general case. When A is Hermitian, have real eigenvalues, the convergence rate is instead quadratic* $(|\lambda_2| \,/\, |\lambda_1|)^2$*. [3] pp. 476.*

We now have all the tools we need to formulate a useful algorithm. This is the Power method:

Algorithm 1 Power method to find λ_1 and $\mathbf{x}_1$

$k = 0$
Choose $\mathbf{v}^{(0)}$ such that $\left\|\mathbf{v}^{(0)}\right\|_2 = 1$
while not converged **do**
  Compute $\mathbf{v}^{(k+1)} = A\mathbf{v}^{(k)}$
  Normalise $\mathbf{v}^{(k+1)} = \mathbf{v}^{(k+1)} \,/\, \left\|\mathbf{v}^{(k+1)}\right\|_2$
  Compute $\lambda_1^{(k)} = (\mathbf{v}^{(k+1)})^* A\mathbf{v}^{(k+1)}$
  Add $k = k + 1$
end while

Before we move on we should note what we mean with convergence in the algorithm above and all other algorithms in this section 3 about the Power method. We could of course choose to run the algorithm for some predeterined k number of iterations but a better way is to choose some convergence criteria.

Convergence criteria. Whenever we talk about the convergence in the algorithms in this section we iterate until

$$\left|\frac{\lambda^{(k)} - \lambda^{(k-1)}}{\lambda^{(k-1)}}\right| < \epsilon$$

where ϵ is some small number.

Example 3.4 (The complete Power method)**.** Let's run through the algorithm and compute both the dominant eigenvalue and eigenvector. Again, we use the same matrix

$$A = \begin{bmatrix} -5 & -2 \\ 4 & 1 \end{bmatrix}$$

from examples 3.2 and 3.3. We choose the same non-zero vector from example 3.3 but now we normalise it to get

$$\mathbf{v}^{(0)} = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}.$$

We then start the iteration from algorithm 1.

$$\mathbf{v}^{(1)} = A\mathbf{v}^{(0)} = \begin{bmatrix} -5 & -2 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{-7\sqrt{2}}{2} \\ \frac{5\sqrt{2}}{2} \end{bmatrix}$$

$$\text{Normalise } \mathbf{v}^{(1)} = \begin{bmatrix} \frac{-7}{\sqrt{74}} \\ \frac{5}{\sqrt{74}} \end{bmatrix}$$

$$\lambda_1^{(1)} = \begin{bmatrix} \frac{-7}{\sqrt{74}} & \frac{5}{\sqrt{74}} \end{bmatrix} \begin{bmatrix} -5 & -2 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} \frac{-7}{\sqrt{74}} \\ \frac{-7}{\sqrt{74}} \end{bmatrix} = -3.9189.$$

We keep iterating and the resulting approximations for the dominant eigenvector and eigenvalue are summarised in table 1 and table 2 respectively.

Table 1: Dominant eigenvector approximation

$\mathbf{v}^{(2)}$	$\mathbf{v}^{(3)}$	...	$\mathbf{v}^{(8)}$	$\mathbf{v}^{(9)}$	$\mathbf{v}^{(10)}$
$\begin{bmatrix} 0.7359 \\ -0.6770 \end{bmatrix}$	$\begin{bmatrix} -0.7161 \\ 0.6979 \end{bmatrix}$	...	$\begin{bmatrix} 0.7071 \\ -0.7070 \end{bmatrix}$	$\begin{bmatrix} -0.7071 \\ 0.7070 \end{bmatrix}$	$\begin{bmatrix} 0.7071 \\ -0.7071 \end{bmatrix}$

Table 2: Dominant eigenvalue approximation

$\lambda_1^{(2)}$	$\lambda_1^{(3)}$	...	$\lambda_1^{(8)}$	$\lambda_1^{(9)}$	$\lambda_1^{(10)}$
-3.2461	-3.0766	...	-3.0003	-3.0001	-3.0000

As expected the vector $\mathbf{v}^{(k)}$ approaches the normalised dominant eigenvector from example 3.2 with switching signs in every iteration and $\lambda_1^{(k)}$ approaches the dominant eigenvalue $\lambda_1 = -3$.

3.2 The inverse Power method

Opposite of the Power method, the inverse Power method is useful when trying to find the smallest eigenvalue (in absolute value).

Theorem 3.3 (Convergence of the inverse Power method)**.** *If $A \in \mathbb{C}^{n\times n}$ is a diagonalisable matrix with eigenvalues that can be ordered*

$$|\lambda_1| \geq |\lambda_2| \geq \ldots > |\lambda_n|. \tag{21}$$

Then λ_n^{-1} is the dominant eigenvalue of the inverse matrix A^{-1} and given an initial non-zero vector $\mathbf{v}^{(0)}$ the inverse Power method forms a recursive sequence

$$\mathbf{v}^{(k+1)} = A^{-1}\mathbf{v}^{(k)} \text{ for } k = 0, 1, 2, 3, \ldots \tag{22}$$

which approaches a multiple of the eigenvector $\mathbf{x}_n$.

Proof. The proof of the convergence of the inverse Power method is analogue to the proof of the Power method, theorem 3.1 and therefore we will not present it here. We will however comment on some differences. Note the difference in expression (21) from the corresponding expression in definition 3.1, that we now have a distinct minimum absolute eigenvalue and but no longer a distinct maximum absolute eigenvalue. Then λ_n^{-1} is the dominant eigenvalue of A^{-1}. It is not difficult to see why, if λ is an eigenvalue of A and $\mathbf{x}$ is its corresponding eigenvector, then we have $A\mathbf{x} = \lambda\mathbf{x}$, and so $\mathbf{x} = \lambda A^{-1}\mathbf{x}$, or $\lambda^{-1}\mathbf{x} = A^{-1}\mathbf{x}$. This shows that $\mathbf{x}$ is again an eigenvector of A^{-1} with eigenvalue λ^{-1}. ■

Corollary 3.3. *The convergence rate of the eigenvector* $\mathbf{x_n}$ *and eigenvalue* λ_n *are the same as for the normal Power method seen in corollary 3.1 and 3.2.*

Again, we can formulate the inverse Power method as an algorithm.

Algorithm 2 Inverse Power method to find λ_n and $\mathbf{x}_n$

$k = 0$
Choose $\mathbf{v}^{(0)}$ such that $\left\|\mathbf{v}^{(0)}\right\|_2 = 1$
while not converged **do**
  Solve $A\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)}$
  Normalise $\mathbf{v}^{(k+1)} = \mathbf{v}^{(k+1)} \,/\, \left\|\mathbf{v}^{(k+1)}\right\|_2$
  Compute $\lambda_n = (\mathbf{v}^{(k+1)})^* A\mathbf{v}^{(k+1)} = (\mathbf{v}^{(k+1)})^*\mathbf{v}^{(k)}$
  Add $k = k + 1$
end while

3.3 The shifted inverse Power method

The Power method has its drawbacks in that the convergence is arbitrarily slow since it depends on the absolute value difference between eigenvalues and so far we have only been able to compute the largest and smallest eigenvalues (in absolute values). What if we wanted to compute intermediate eigenvalues? We can modify the existing inverse Power method and introduce a shift μ to find an eigenvalue near μ.

The idea is simple. For any scalar μ that is not an eigenvalue in matrix A, the eigenvectors of the matrix $(A - \mu I)^{-1}$ are the same as for A with corresponding eigenvalues $(\lambda - \mu I)^{-1}$, similar to what we showed in the proof of theorem 3.3. If we introduce a shift μ close to the eigenvalue λ_i then $(\lambda_i - \mu I)^{-1}$ may be much larger than $(\lambda_j - \mu I)^{-1}$ for all $i \neq j$ and we can transform any eigenvalue into a dominant one and the dominance may be as large as we want. Since the convergence rate of the Power method depends on the absolute difference between eigenvalues as seen in corollary 3.3, convergence may be very rapid if we can choose a good shift.

A consequence of how the shifted inverse Power method works is that the eigenvector that we compute can be chosen by supplying the a shift close to a particular eigenvalue. The inverse shifted Power method is therefore very useful in the case we want to calculate one or more eigenvectors from a matrix where the eigenvalues are known. We state the shifted inverse Power method as an algorithm.

Algorithm 3 Shifted inverse Power method to find λ_n and $\mathbf{x}_n$

$k = 0$
Choose $\mathbf{v}^{(0)}$ such that $\left\|\mathbf{v}^{(0)}\right\|_2 = 1$
while not converged **do**
  Solve $(A - \mu I)\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)}$
  Normalise $\mathbf{v}^{(k+1)} = \mathbf{v}^{(k+1)} \,/\, \left\|\mathbf{v}^{(k+1)}\right\|_2$
  Compute $\lambda_n = (\mathbf{v}^{(k+1)})^* A \mathbf{v}^{(k+1)}$
  Add $k = k + 1$
end while

There is no reason why the shift could not be different in each iteration. For such a shift we could use the Rayleigh quotient from definition 3.2 which provides a much better approximation for the eigenvalue than a random shift. Let $\mathbf{v}^{(k)}$ be an approximation for the dominant eigenvector of the matrix A computed as the k-th iteration of the Power method. From theorem 3.2 we know that if $\mathbf{v}^{(k)}$, where $\|\mathbf{v}^{(k)}\|_2 = 1$, is an exact eigenvector of A then

$$A\mathbf{v} = \lambda\mathbf{v} \Rightarrow r = \mathbf{v}^* A\mathbf{v} = \lambda \mathbf{v}^*\mathbf{v} = \lambda. \tag{23}$$

Due to the continuous nature of matrix multiplication, the Rayleigh quotient will in every iteration provide an estimate of the eigenvalue and can therefore be used as a shift. For a more rigorous proof see [8] pp. 326. The use of the Rayleigh quotient as a shift greatly improves the convergence rate of the algorithm and it may be cubic under some assumptions (Hermitian matrix). [6] pp. 208.

Example 3.5 (Rayleigh shifted inverse Power method)**.** To see how fast the Rayleigh shifted inverse Power method is we will provide an example. Consider the matrix

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 6 & -1 & 0 \\ -1 & -2 & -1 \end{bmatrix},$$

with eigenvalues $\lambda_1 = 0$, $\lambda_2 = 3$, $\lambda_3 = -4$ and corresponding eigenvectors

$$\mathbf{x}_1 = \begin{bmatrix} -0.0769 \\ -0.4615 \\ 1 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} -1 \\ -1.5 \\ 1 \end{bmatrix} \quad \mathbf{x}_3 = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}$$

We then choose an initial eigenvector estimate

$$\mathbf{v}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

which is then normalised. The algorithm 3.3 generates the following sequence of eigenvector estimates and eigenvalue estimates respectively for the first 5 iterations seen in table 3 and table 4. The convergence is very rapid and already at iteration 5 we have a very good estimate of the eigenvalue $\lambda_2 = 3$ and its corresponding eigenvector $\mathbf{x}_2 = (-1, -1.5, 1)$ (normalised).

Table 3: Eigenvector approximation

$\mathbf{v}^{(1)}$	$\mathbf{v}^{(2)}$	$\mathbf{v}^{(3)}$	$\mathbf{v}^{(4)}$	$\mathbf{v}^{(5)}$
$\begin{bmatrix} 0.3296 \\ 0.6229 \\ -0.7095 \end{bmatrix}$	$\begin{bmatrix} 0.6622 \\ 0.7493 \\ 0.0085 \end{bmatrix}$	$\begin{bmatrix} -0.5229 \\ -0.7445 \\ 0.4151 \end{bmatrix}$	$\begin{bmatrix} 0.4882 \\ 0.7291 \\ -0.4798 \end{bmatrix}$	$\begin{bmatrix} -0.4851 \\ -0.7276 \\ 0.4850 \end{bmatrix}$

Table 4: Eigenvalue approximation

$\lambda^{(1)}$	$\lambda^{(2)}$	$\lambda^{(3)}$	$\lambda^{(4)}$	$\lambda^{(5)}$
1.7436	3.8337	3.2791	3.0002	3.0000

When using the Rayleigh quotient as a shift the algorithm can be formulated as follows.

Algorithm 4 Rayleigh shifted inverse Power method to find λ_n and $\mathbf{x}_n$

$k = 0$
Choose $\mathbf{v}^{(0)}$ such that $\left\|\mathbf{v}^{(0)}\right\|_2 = 1$
Compute $\lambda^{(0)} = (\mathbf{v}^{(0)})^* A\mathbf{v}^{(0)}$
while not converged **do**
  Solve $(A - \lambda^k I)\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)}$
  Normalise $\mathbf{v}^{(k+1)} = \mathbf{v}^{(k+1)} \,/\, \left\|\mathbf{v}^{(k+1)}\right\|_2$
  Compute $\lambda^{(k+1)} = (\mathbf{v}^{(k+1)})^* A\mathbf{v}^{(k+1)}$
  Add $k = k + 1$
end while

3.4 Subspace iteration

The Power method that we have covered in this section can be further generalised and thought of in terms of subspace iteration. We noted in example 3.3 that the sequence of eigenvectors approximations produced by the Power method does not approach the dominant eigenvector $\mathbf{x}_1$ itself but rather scalar multiples of it, and sometimes with switching sign in each iteration. The sequence thus does not convergence to the dominant eigenvector $\mathbf{x}_1$, instead we can say that it converges to the subspace spanned by the dominant eigenvector, span$\{\mathbf{x}_1\}$. In this sense each multiple of $\mathbf{x}_1$ can be seen as a representative of this eigenspace. Similarly the whole sequence generated by the Power method

$$\mathbf{v}^{(0)}, A\mathbf{v}^{(0)}, A^2\mathbf{v}^{(0)}, \ldots, A^k\mathbf{v}^{(0)}$$

can be seen as representatives for the space span$\{A^k\mathbf{v}^{(0)}\}$. Thus the Power method may be described as the process of iterating over subspaces.

Theorem 3.4 (Subspace iteration)**.** *Let $S = \text{span}\{\mathbf{v}^{(0)}\}$ be the one-dimensional subspace spanned by the initial vector $\mathbf{v}^{(0)}$. $S \in \mathbb{C}^n$. Then the sequence*

$$S, AS, A^2S, \ldots, A^kS \tag{24}$$

is formed by the Power method. The sequence converges to the eigenspace $T = \text{span}\{\mathbf{x}\}$, spanned by the dominant eigenvector $\mathbf{x}$. The convergence is true in the sense that the angle between T and A^kS converges to zero.

Remark (Angle definition)**.** To see what is meant with the angle between subspaces see for example [2]. To avoid the complications this brings with it and the need for further theory we will continue to use vectors and not subspaces in the rest of the thesis.

Proof. A formal proof requires a considerable amount of theory and will not be presented here. First, we would develop theory for what we mean by the angle between subspaces and provide some consequences of this. Then we need to prove the convergence itself. For a formal proof see for example Watkins [8]. ■

4 The QR algorithm

In the second section we introduced what eigenvalues are and showed some important characteristics of matrices. We ended by proving that every matrix $A \in \mathbb{C}^{n\times n}$ can be written as a Schur decomposition. Several methods exist for computing the Schur decomposition. One such method is the QR-method which we are know ready to introduce. Throughout the chapter we will build on the basic QR-method to finally make it a useful and efficient algorithm for computing the Schur decomposition and ultimately finding eigenvalues.

4.1 The basic QR-algorithm

The goal of the QR-method is computing the Schur decomposition by means of similarity transformations. As the name suggests the QR-method is tightly coupled with the QR-decomposition. Let us first define an iteration of the QR-method, we will call it a QR-step.

Definition 4.1 (QR step)**.** Let the QR-decomposition of a matrix $A \in \mathbb{C}^{n\times n}$ be

$$A = QR \tag{25}$$

where Q is a unitary matrix and R is an upper triangular matrix. For a proof of the QR-decomposition of all matrices $A \in \mathbb{C}^{n\times n}$ see for example section 5.2 of [5]. Let $A = A^{(k-1)}$, then we can define a QR-step as follows

$$A^{(k-1)} = Q^{(k)}R^{(k)},\ A^{(k)} = R^{(k)}Q^{(k)} = (Q^{(k)})^* A^{(k-1)}Q^{(k)} \text{ for every } k = 1, 2, \dots \tag{26}$$

Lemma 4.1 (Unitary similar in each QR-step)**.** *For every step k of the QR-method, the matrices $A^{(k)}$ and $A^{(k-1)}$ are unitary similar.*

Proof. From definition 4.1 it follows that $A^{(k)}$ can be written $A^{(k)} = (Q^{(k)})^* A^{(k-1)}Q^{(k)}$ and thus the matrices are unitary similar. ■

Corollary 4.1 (The unitary similarity is transitive over multiple QR-steps)**.** *In every step k, the matrix computed by the QR-algorithm, $A^{(k)}$, is unitary similar to the initial matrix A.*

Lemma 4.2 (Applying a QR-step to a Hermitian matrix with generate a new Hermitian matrix)**.** *A Hermitian matrix will remain unchanged under a QR step, that is if A is Hermitian then $A^{(k)}$ is also Hermitian in every step k*

Proof. From lemma 4.1 we saw that $A^{(k)}$ and $A^{(k-1)}$ are unitary similar. In conjunction with corollary 4.1, when the QR-step is applied repeatedly we get

$$\begin{aligned} A^{(k)} &= (Q^{(k)})^* A^{(k-1)}Q^{(k)} \\ &= (Q^{(k)})^* (Q^{(k-1)})^* A^{(k-2)}Q^{(k-1)}Q^{(k)} \\ &= \cdots = (Q^{(k)})^* \dots (Q^{(0)})^* A^{(0)}Q^{(0)} \dots Q^{(k)}. \end{aligned} \tag{27}$$

Then the conjugate transpose of $A^{(k)}$ will be

$$\begin{aligned} (A^{(k)})^* &= ((Q^{(k)})^* \dots (Q^{(0)})^* A^{(0)}Q^{(0)} \dots Q^{(k)})^* \\ &= (Q^{(k)})^* \dots (Q^{(0)})^* (A^{(0)})^* Q^{(0)} \dots Q^{(k)}, \end{aligned} \tag{28}$$

and if A is Hermitian then by applying expressions (27) and (28) we get

$$(A^{(k)})^* = (Q^{(k)})^* \dots (Q^{(0)})^* (A^{(0)})^* Q^{(0)} \dots Q^{(k)} = A^{(k)}. \tag{29}$$

■

We will end this section on the basic QR-method with a theorem about the convergence of the QR-method. When the QR-steps are applied repeatedly as in equation 27, then under some conditions $A^{(k)}$ converges to an upper triangular matrix. We can make this into a more formal theorem.

Theorem 4.1 (Basic QR-method)**.** *Let some matrix $A \in \mathbb{C}^{n\times n}$. Suppose that A has n eigenvalues with distinct absolute values, $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$, and the Schur decomposition $A = VUV^*$ from theorem 2.4. When the QR-step from definition 4.1 is iterated indefinitely the sequence $A^{(k)}$ converges to an upper triangular matrix with eigenvalues in monotone decreasing order of the absolute value down the diagonal.*

Proof. We will not provide a complete proof for the QR-method. Our approach will instead be to provide a partial proof by relating the QR-method to the Power method for the case when the matrix A is real and symmetric, see section 4.2. ■

Corollary 4.2. *If the matrix A is Hermitian then the sequence $A^{(k)}$ converges to a diagonal matrix.*

Proof. This follows directly from theorem 4.1 and corollary 2.1. ■

Before moving on we state the basic QR-method in its algorithmic form.

Algorithm 5 QR-method

$A = A^{(0)}$
while not converged **do**
  Compute $A^{(k-1)} = Q^{(k)}R^{(k)}$
  Compute $A^{(k)} = R^{(k)}Q^{(k)}$
  $k = k + 1$
end while

Convergence criteria. By convergence in this algorithm we mean that the $n \times n$ matrix $A^{(k)}$ generated by the algorithm is sufficiently upper triangular, and more precisely we iterate until all the elements of $A^{(k)}$ below the diagonal are sufficiently close to zero, that is $|a_{i,j}^{(k)}| < \epsilon$ when $i > j$ for all $i = 1, 2, \dots, n$ and a small ϵ.

4.2 The QR-method and the Power method

In section 3 we introduced the Power method and in section 3.4 we briefly touched on the idea that the Power method can be generalised as a type of subspace iteration. We will now relate the Power method to the QR-method and show how the QR-method is nothing more than applying the Power method to multiple vectors at the same time. This is known as simultaneous iteration or sometimes orthogonal iteration.

For the case of simplicity, in this section and going forward in the thesis, if nothing else is stated we will work with real, symmetric matrices with linearly independent column vectors which thus have real eigenvalues and complete sets of orthogonal eigenvectors , see corollary 2.1 and [8] pp. 341. For notation this means that $A = A^T \in \mathbb{R}^{n\times n}$, $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{v}^* = \mathbf{v}^T$ and $\|\mathbf{v}\| = \sqrt{\mathbf{v}^T\mathbf{v}}$.

Let A be such a matrix we just described, We can therefore, just as we showed in the Power method section, write the starting vector $\mathbf{v}^{(0)}$ as a linear combination of the eigenvectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of A:

$$\mathbf{v}^{(0)} = c_1\mathbf{x}_1 + \cdots + c_n\mathbf{x}_n$$

Note that only eigenvectors that are not orthogonal to $\mathbf{v}^{(0)}$ have a chance to be found with the Power method iteration, given that the calculations are done in exact arithmetic, see remark for theorem 3.1. This suggests that there is a possibility to find multiple eigenvectors and eigenvalues with the Power method if we start with several different initial vectors, each orthogonal to all the others. We begin with n linearly independent real vectors $V^{(0)} = \left[\mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_n^{(0)}\right]$. However, if we then apply k power iterations because of rounding errors in practice all such vectors will eventually converge to the dominant (scaled) eigenvector $\mathbf{x}_1$ and nothing is gained, see for example [1] pp. 107. The solution to keeping the vector ortoghonal to eachother is to orthonormalise at each step using the QR-decomposition where Q is a matrix with orthonormal column vectors and R is a diagonal matrix. The procedure can be written as follows in its algorithmic form.

Algorithm 6 Simultaneous iteration

Choose $V^{(0)}$ with orthonormal columns
while not converged **do**
  Compute $V^{(k)} = AV^{(k-1)}$
  Compute $V^{(k)} = \hat{Q}^{(k)}\hat{R}^{(k)}$
  Set $V^{(k)} = \hat{Q}^{(k)}$
  $k = k + 1$
end while

We conclude our discussion in a theorem from [6].

Theorem 4.2. *Let $\mathbf{v}_1^{(0)}, \mathbf{v}_2^{(0)}, \ldots, \mathbf{v}_n^{(0)}$ to be a set of initial orthonormal vectors that forms the $m \times n$ initial matrix*

$$V^{(0)} = \left[\mathbf{v}_1^{(0)}, \mathbf{v}_2^{(0)}, \ldots, \mathbf{v}_n^{(0)}\right] \tag{30}$$

and after k iterations of the algorithm we get

$$\hat{Q}^{(k)} = \left[\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n\right]. \tag{31}$$

Under the assumptions that:

1. *A is a real symmetric matrix where the first $n+1$ eigenvalues are distinct in absolute value*

$$|\lambda_1| > |\lambda_2| \ldots |\lambda_n| > |\lambda_{n+1}| \geq |\lambda_{n+2}| \geq \cdots \geq |\lambda_m|.$$

2. *The leading principal submatrices of the product $\left[\mathbf{q}_2, \ldots, \mathbf{q}_n\right]^T V^{(0)}$ are non-singular. (The leading principal submatrices of a matrix B are the top left square submatrices of B)*

Then the space spanned by $\hat{Q}^{(k)}$, span$\{\mathbf{q}_1, \mathbf{q}_2, \ldots \mathbf{q}_n\}$*, will converge to the space spanned by the first n eigenvectors of A,* span$\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n\}$.

Proof. For a proof see [6] pp. 214 ■

Now we have a method that under some assumptions computes all the eigenvectors of matrix A.

It is quite intuitive to see that if we choose the initial matrix $V^{(0)} = I$ where I is the identity matrix then algorithms in section 4.1 and section 4.2 are the same. We rewrite the two algorithms slightly with again slightly different notation.

Simultaneous iteration:

$$\begin{aligned} &\bar{Q}^{(0)} = I, \\ &\quad \text{Compute } W = A\bar{Q}^{(k-1)}, \\ &\quad \text{Compute } W = \bar{Q}^{(k)}R^{(k)}, \\ &\text{Let } A^{(k)} = (\bar{Q}^{(k)})^T A\bar{Q}^{(k)}, \\ &\text{Let } \bar{R}^{(k)} = R^{(k)}R^{(k-1)}\dots R^{(1)} \end{aligned} \tag{32}$$

Basic QR-method:

$$\begin{aligned} &A^{(0)} = A, \\ &\quad \text{Compute } A^{(k-1)} = Q^{(k)}R^{(k)}, \\ &\quad \text{Compute } A^{(k)} = R^{(k)}Q^{(k)}, \\ &\text{Let } \bar{Q}^{(k)} = Q^{(1)}Q^{(2)}\dots Q^{(k)}, \\ &\text{Let } \bar{R}^{(k)} = R^{(k)}R^{(k-1)}\dots R^{(1)} \end{aligned} \tag{33}$$

Theorem 4.3. *The simultaneous iteration method and the QR method in* (32) *and* (33) *above both generate the same sequence of matrices $A^{(k)}$, $\bar{R}^{(k)}$ and $\bar{Q}^{(k)}$, which satisfy the two properties:*

1. $A^k = \bar{Q}^{(k)}\bar{R}^{(k)}$
2. $A^{(k)} = (\bar{Q}^{(k)})^T A\bar{Q}^{(k)}$.

Proof. The proof proceeds by induction. The base case when $k = 0$ is true since this implies, $A^0 = \bar{Q}^{(0)} = \bar{R}^{(0)} = I$ and $A^{(0)} = A$. Next we consider the case $k \geq 1$. When $k-1$, suppose that both algorithms produce the same matrices $A^{(k-1)}$, $\bar{R}^{(k-1)}$ and $\bar{Q}^{(k-1)}$, and that property 1 and property 2 are satisfied. We can then move on to the case at the k-th iteration. For simultaneous iteration we have

$$\begin{aligned} A^k = AA^{k-1} &= A\bar{Q}^{(k-1)}\bar{R}^{(k-1)} \quad \text{induction assumption property 1} \\ &= \bar{Q}^{(k)}R^{(k)}\bar{R}^{(k-1)} \quad \text{row 2 and 3 from (32)} \\ &= \bar{Q}^{(k)}\bar{R}^{(k)} \end{aligned}$$

and thus property 1 is satisfied. Property 2 is satisfied directly from the definition in the simultaneous iteration algorithm, row 4 of (32).

In the case of the QR-method from (33) we can verify property 1 by the sequence

$$\begin{aligned} A^k = AA^{k-1} &= A\bar{Q}^{(k-1)}\bar{R}^{(k-1)} \quad \text{induction assumption property 1} \\ &= \bar{Q}^{(k-1)}A^{(k-1)}\bar{R}^{(k-1)} \quad \text{induction assumption property 2} \\ &= \bar{Q}^{(k-1)}Q^{(k)}R^{(k)}\bar{R}^{(k-1)} \quad \text{row 2 from (33)} \\ &= \bar{Q}^{(k)}\bar{R}^{(k)}. \end{aligned}$$

Finally we also have

$$\begin{aligned} A^{(k)} &= R^{(k)}Q^{(k)} \text{ row 3 from (33)} \\ &= (Q^{(k)})^T A^{(k-1)} Q^{(k)} \text{ row 2 from (33)} \\ &= (Q^{(k)})^T (\bar{Q}^{(k-1)})^T A \bar{Q}^{(k-1)} Q^{(k)} \text{ induction assumption property 2} \\ &= (\bar{Q}^{(k)})^T A \bar{Q}^{(k)}. \end{aligned}$$

which proves that property 2 is satisfied.

Since both algorithms satisfy both properties at the k-th iteration, the theorem has been proved by induction and the algorithms produce the same result and are in that sense equivalent. ■

Corollary 4.3. *Simultaneous **inverse** iteration applied to a permutated matrix $A^{-k}P$ where the permutation matrix*

$$P = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}$$

reverses the order of columns if applied from the right, is equivalent to the QR-method.

Since the QR-method behaves similar to the inverse Power method it would make sense if we could introduce a shift to speed up the convergence of the QR-method. We will discuss different shift strategies later in section 4.4.

4.3 The two step QR-algorithm

Although the basic QR-method in general converges to a Schur decomposition when the number of iterations $k \to \infty$, several improvements can be made to accelerate the convergence. One way to do this is to do some work on the matrix before applying our algorithm. For these improvements the Hessenberg matrix structure is crucial.

Definition 4.2 (Hessenberg matrix)**.** A matrix $H \in \mathbb{C}^{n\times n}$ is called a Hessenberg matrix if its elements below the lower off-diagonal are zero, that is $h_{i,j} = 0$ when $i > j + 1$. In addition, the matrix H is called an unreduced Hessenberg matrix if $h_{i,i+1} \neq 0$ for all $i = 1, 2, \ldots, n-1$

The structure of Hessenberg matrix H looks as follows in the general case

$$H = \begin{bmatrix} \times & \cdots & \cdots & \cdots & \times \\ \times & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \times \\ 0 & \cdots & 0 & \times & \times \end{bmatrix}$$

and in the case where the original matrix A is Hermitian then the upper Hessenberg matrix is also

Hermitian and thus tridiagonal, this follows from theorem 4.4,

$$T = \begin{bmatrix} \times & \times & 0 & \cdots & 0 \\ \times & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \times \\ 0 & \cdots & 0 & \times & \times \end{bmatrix}. \tag{34}$$

In both matrices the zeroes are zero elements and the $\times$ may be non-zero elements. We can define tridiagonal matrices more carefully.

Definition 4.3 (Tridiagonal matrix)**.** A matrix $T \in \mathbb{C}^{n\times n}$ is called tridiagonal if all non-zero elements are found on the diagonal and along the adjacent "slots" to the diagonal, i.e. along the subdiagonal, diagonal and superdiagonal. The matrix in (34) is tridiagonal.

Since the Hessenberg matrix is almost triangular it is cheaper to work with, see [8] for more details. In fact the Hessenberg matrix is as close as we come to an uppper triangular matrix with a direct method, since there is no direct method that gives an explicit solution for polynomials of higher degree than 4. What makes the Hessenberg form so useful in the QR-method is the fact that it is possible to do a direct transformation from any matrix A to a Hessenberg matrix H and keep the eigenvalues the same, we state this as a theorem, but first we define a reflector that we will use in the proof of the theorem.

Definition 4.4. Let $x, y \in \mathbb{C}^n$ be two distinct non-zero vectors with the same norm and let $\mathbb{H}$ denote the hyperplane orthogonal to $x - y$ that passes through the origin. Then we can think of a reflector as a transformation Q that maps (reflect) a vector to the other side of $\mathbb{H}$. Similar to the image of a mirror. Given x and y we have $x \to y$ and $y \to x$ under Q. See section 5.1 of [5] for more about reflectors and a precise definition.

Theorem 4.4. *Every matrix $A \in \mathbb{C}^{n\times n}$ is unitary similar to a upper Hessenberg matrix*

$$H = Q^* A Q$$

and thus have the same eigenvalues.

Proof. We will give an outline to a proof, for a full proof see for example [8]. We can construct Q (and Q^*) as the product of $n-2$ reflectors. A reflector is here a unitary transformation to introduce zeroes in the matrix A. We write the first reflector Q_1^* on the form

$$Q_1^* = \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q}_1^* \end{bmatrix}$$

where $\hat{Q}_1^*$ is an $(n-1)\times(n-1)$ matrix (reflector) such that

$$\hat{Q}_1^* \begin{bmatrix} a_{2,1} \\ a_{3,1} \\ \vdots \\ a_{n,1} \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{35}$$

We can see that the reflector is created in such a way so that the first row of A is left the same if left-multiplied by Q_1^*. From expression (35) it is also clear that the idea is to create the reflectors in such a way that all the elements below the subdiagonal are eliminated (set to zero) when a

similarity transformation by Q is applied to A. By left multiplication of A by Q_1^* we get

$$Q_1^*A = \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q}_1^* \end{bmatrix} \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{1,1} & a_{2,2} & & a_{2,n} \\ \vdots & & \ddots & \\ a_{1,1} & & & a_{1,n} \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & a_{1,n} \\ \alpha & \times & & & \vdots \\ 0 & \times & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & \times & \cdots & \times & \times \end{bmatrix}.$$

where $\times$ may be non-zero elements, just as before. If we then apply Q_1 to the right

$$Q_1^*AQ_1 = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & a_{1,n} \\ \alpha & \times & & & \vdots \\ 0 & \times & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & \times & \cdots & \times & \times \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q}_1 \end{bmatrix} = \begin{bmatrix} a_{1,1} & \times & \cdots & \cdots & \times \\ \alpha & \times & & & \vdots \\ 0 & \times & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & \times & \cdots & \times & \times \end{bmatrix}$$

the first column (including the zero entries) is kept. The process is repeated with a second reflector

$$Q_2^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \hat{Q}_2^* \end{bmatrix}$$

where $\hat{Q}_2^*$ is a $(n-2)\times(n-2)$ matrix that fulfills a condition analogue to (35) so that $Q_2^*Q_1^*AQ_1$ and $Q_2^*Q_1^*AQ_1Q_2$ have zeroes down the second column from the fourth row down. If the same process is repeated $n-2$ times, letting $Q^* = Q_1^*Q_2^*\ldots Q_{n-2}^*$ and $Q = Q_1Q_2\ldots Q_{n-2}$ we reach the desired results, $H = Q^*AQ$. Since the matrices A and H are similar their characteristic polynomial and eigenvalues are the same which follows from theorem 2.3. ■

The improved QR-method can be split into two phases or steps. The phases can be illustrated as follows for a matrix with dimension $n = 5$ in the general case:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

or in the case of a Hermitian matrix (see lemma 4.2)

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 \\ 0 & \times & \times & \times & 0 \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & 0 & 0 & 0 & 0 \\ 0 & \times & 0 & 0 & 0 \\ 0 & 0 & \times & 0 & 0 \\ 0 & 0 & 0 & \times & 0 \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}$$

where the first step is the reduction to a Hessenberg matrix and the second the step is the application of the QR-method.

4.4 Choosing shifts and deflation

We ended section 4.2 by implying that we could introduce shifts to improve the convergence rate of the QR-method. Introducing shifts to the QR-method accomplishes the same benefits as in the inverse shifted Power method, where the last column quickly converges to an eigenvector. We have previously introduced Rayleigh shifts in section 3.3 and the Rayleigh shift is a natural way to start but there exists several different shift strategies. In this section we will introduce the Wilkinson shift and compare the properties and convergence rate of both shift-strategies in the case for real symmetric tridiagonal matrices. Given a shift μ the basic shifted QR-method can be written as the following algorithm.

Algorithm 7 QR-method with shift

$T = A^{(0)}$
while not converged **do**
  Compute shift $\mu^{(k)}$
  Compute $(A^{(k)} - \mu^{(k)} I) = Q^{(k)} R^{(k)}$
  Compute $(A^{(k+1)} + \mu^{(k)} I) = R^{(k)} Q^{(k)}$
  $k = k + 1$
end while

The convergence criteria here is the same as for the algorithm in section 4.1.

To estimate the eigenvalue corresponding to the eigenvector that is approximated by the last corresponding eigenvector in $\bar{Q}^{(k)}$ from (33), $\mathbf{q}_m^{(k)}$, we apply the Rayleigh quotient from definition 3.2 to that last row and get

$$\mu^{(k)} = r(\mathbf{q}_m^{(k)}) = \frac{(\mathbf{q}_m^{(k)})^T A \mathbf{q}_m^{(k)}}{(\mathbf{q}_m^{(k)})^T \mathbf{q}_m^{(k)}} = (\mathbf{q}_m^{(k)})^T A \mathbf{q}_m^{(k)}.$$

Notice that the Rayleigh quotient $r(\mathbf{q}_m^{(k)})$ is the equal to $a_{m,m}^{(k)}$ since applying k QR-steps to A, $A^{(k)} = (\bar{Q}^{(k)})^T A \bar{Q}^{(k)}$. Therefore when using the Rayleigh quotient shift, the shift is simply set to $\mu^{(k)} = a_{m,m}^{(k)}$ which is the most right bottom element of the matrix A at iteration k .

A slightly more sophisticated method for choosing a shift is the Wilkinson shift, see [9]. Let the elements of a $m \times m$ symmetric tridiagonal matrix T, definition 4.3, at the k-th QR-step be defined as

$$T^{(k)} = \begin{bmatrix} \alpha_1^{(k)} & \beta_1^{(k)} & & \\ \beta_1^{(k)} & \alpha_2^{(k)} & \ddots & \\ & \ddots & \ddots & \beta_{m-1}^{(k)} \\ & & \beta_{m-1}^{(k)} & \alpha_m^{(k)} \end{bmatrix}. \tag{36}$$

We can then define the Wilkinson shift as follows.

Definition 4.5 (Wilkinson shift)**.** The Wilkinson shift $\mu^{(k)}$ is the eigenvalue of the left-bottom 2×2 submatrix of T from (36)

$$\begin{bmatrix} \alpha_{m-1}^{(k)} & \beta_{m-1}^{(k)} \\ \beta_{m-1}^{(k)} & \alpha_m^{(k)} \end{bmatrix}$$

that is closest to $\alpha_m^{(k)}$. Thus, per definition 2.1 it fulfills the following relations:

$$(\alpha_{m-1}^{(k)} - \mu^{(k)})(\alpha_m^{(k)} - \mu^{(k)}) = {\beta_{m-1}^{(k)}}^2,$$
$$|\alpha_m^{(k)} - \mu^{(k)}| \leq {\beta_{m-1}^{(k)}}^2 \leq |\alpha_{m-1}^{(k)} - \mu^{(k)}|.$$

With a formula, a numerically stable version of the Wilkinson shift can be obtained

$$\mu^{(k)} = \alpha_m^{(k)} - \frac{\text{sign}(\delta)(\beta_{m-1}^{(k)})^2}{(|\delta| + \sqrt{\delta^2 + (\beta_{m-1}^{(k)})^2})},$$

where $\delta = (\alpha_{m-1}^{(k)} - \alpha_m^{(k)})/\ 2$ and if $\delta = 0$, sign(δ) can be set either -1 or 1. [6]

One of Wilkinson's reasons for introducing his own shift method was that the Rayleigh shift does not guarantee global convergence. Let us look at a simple example.

Example 4.1 (Convergence of the shifted QR-method for matrices)**.** Let A be a symmetric matrix,

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

In its basic form the QR-method forms the sequence,

$$A = Q^{(1)}R^{(1)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
$$A^{(1)} = R^{(1)}Q^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = A$$

and it does not converge. Similarly applying a Rayleigh shift, which in this case is $\mu = 0$, does not give convergence since $A = (A - 0I)$.

In 1968 Wilkinson [9] showed that the Wilkinson shift guarantees convergence for all symmetric matrices and also quicker convergence than the Rayleigh shift in some instances, he also showed the convergence rates of both shift strategies. The convergence rates can be summarised in two theorems.

Theorem 4.5 ([9] [7] Convergence with Rayleigh shift)**.** *Suppose that the QR-method with the Rayleigh shift ($\mu^{(k)} = \alpha_m^{(k)}$) is applied to the tridiagonal matrix T, with notation like in* (36)*, k times, so that we have $T^{(k)}$, then the following facts are true*

- *Monotone decrease of non-diagonal element sequence $\{\beta_{m-1}^{(k)}\}$ for every iteration, that is $|\beta_{m-1}^{(k+1)}| \leq |\beta_{m-1}^{(k)}|$.*
- *Cubic rate of convergence, $\beta_{m-1}^{(k+1)} = \mathcal{O}((\beta_{m-1}^{(k)})^3)$ if convergence $\beta_{m-1}^{(k)} \to 0$ occurs when $k \to \infty$.*
- *Linear rate of convergence, $\beta_{m-2}^{(k)} \to 0$ if convergence $\beta_{m-1}^{(k)} \to \beta > 0$ when $k \to \infty$.*

Theorem 4.6 ([9] [7] Convergence with Wilkinson shift)**.** *Suppose that the QR-method with the Wilkinson shift from definition 4.5 is applied to the tridiagonal matrix T, with notation like in* (36)*, k times so that we have $T^{(k)}$, then the following facts are true*

- *Monotone decrease of non-diagonal element sequence $\{\beta_{m-1}^{(k)}\beta_{m-2}^{(k)}\}$ for every every iteration, that is $|\beta_{m-1}^{(k+1)}\beta_{m-2}^{(k+1)}| \leq |\beta_{m-1}^{(k)}\beta_{m-2}^{(k)}|$.*

- *Quadratic rate of convergence* $\beta_{m-1}^{(k+1)} = \mathcal{O}((\beta_{m-1}^{(k)})^2)$
- *For any* $\epsilon > 0$*, there exists a* k *such that* $|\beta_{m-1}^{(k)}| \leq \epsilon$*, global convergence is guaranteed* $\beta_{m-1}^{(k)} \to 0$ *when* $k \to \infty$.

Theorem 4.5 states that either both of $\beta_{m-1}^{(k)}$ and $\beta_{m-2}^{(k)}$ or just one of them tend to go towards 0 with cubic convergence rate for the algorithm. However, for some tridiagonal matrices T, the convergence rate is only linear. In the case of the Wilkinson shift in theorem 4.6, the convergence is guaranteed since $\beta_{m-1}^{(k)} \to 0$ and it is at least quadratic. The convergence rate is true in the sense that the problem can be deflated to a smaller one. For proofs of theorems 4.5 and 4.6 see for example [9] and [7]. We should emphasize that the convergence in the theorems only apply to symmetric matrices that have been reduced to tridiagonal form.

The QR-method with a good shift strategy and deflation is a useful algorithm for computing eigenvalues and it is therefore often referred to as the practical QR-method. In its algorithmic for we can write it as follows.

Algorithm 8 The practical QR-method

$T = A^{(0)}$
while not converged **do**
  Compute shift $\mu^{(k)}$
  Compute $(A^{(k)} - \mu^{(k)} I) = Q^{(k)} R^{(k)}$
  Compute $(A^{(k+1)} + \mu^{(k)} I) = R^{(k)} Q^{(k)}$
  Check for convergence $\beta_{m-1}^{(k)} \approx 0$
  Save the eigenvalue $\lambda_m = \alpha_m^{(k)}$
  Deflate problem $A^{(0)} = A_{1:m-1,1:m-1}$
  $k = k + 1$
end while

4.5 The implicit QR-method

In practice the explicit of decomposition of the shifted matrices, $(A^{(k)} - \mu^{(k)} I) = Q^{(k)} R^{(k)}$ and $(A^{(k+1)} + \mu^{(k)} I) = R^{(k)} Q^{(k)}$ is not computed, instead the same result can be accomplished implicitly. This is an implication of a theorem known as the implicit Q-theorem. We end this thesis by proving the Implicit Q-theorem and giving a short description for how the implicit algorithm works.

Theorem 4.7 ([3] Implicit Q-theorem)**.** *Let a matrix* $A \in \mathbb{R}^{n \times n}$ *and a orthogonal matrix* $Q = \begin{bmatrix} \mathbf{q}_1, \ldots, \mathbf{q}_n \end{bmatrix}$ *such that* $H = Q^T A Q$ *is an unreduced upper Hessenberg matrix with real positive subdiagonal entries. Then* H *and* Q *are uniquely (up to signs) determined by* A *and the first column* $\mathbf{q}_1 = Q\mathbf{e}_1$ *in* Q.

Before we prove the Implicit Q-theorem we can shortly explain or restate what it means. If we have two unreduced Hessenberg matrices $H = Q^T A Q$ and $G = V^T A V$, and $V_{:,1} = Q_{:,1}$ (the first columns are the same) then $V_{:,i} = Q_{:,i}$ for $i = 2, 3, \ldots, n$ (the rest of the columns). Furthermore, if the former is true and A in both Hessenberg matrices H and G are the same then $H = G$ since we are applying the same orthogonal transformation in both cases.

Proof. The proof follows very much from [3]. We assume that the k first columns of Q, $\mathbf{q}_1, \ldots, \mathbf{q}_k$ and the $k-1$ first columns of H have been computed. Since $\mathbf{q}_1$ is known this is valid for $k = 1$. Setting the k-th columns of $QH = AQ$ equal gives us

$$h_{1,k}\mathbf{q}_1 + \cdots + h_{k,k}\mathbf{q}_k + h_{k+1,k}\mathbf{q}_{k+1} = A\mathbf{q}_k, \text{ for } k = 1, \ldots, n-1.$$

If we multiply this expression by ${\mathbf{q}_i}^T$ and using the fact that Q is orthogonal this gives us $h_{i,k} = {\mathbf{q}_i}^T A\mathbf{q}_k$, for $i = 1, \ldots, k$. Since H is unreduced, $h_{k+1,k} \neq 0$ and we can write

$$\mathbf{q}_{k+1} = h_{k+1,k}^{-1}(A\mathbf{q}_k - \sum_{i=1}^{k} h_{i,k}\mathbf{q}_i),$$

This expression determines $\mathbf{q}_{k+1}$ uniquely up to signs ($\pm$), and if the subdiagonal elements $h_{k+1,k}$ are real and positive (like in the theorem) completely uniquely. ■

So how does the implicit algorithm work? Let T be a real symmetric tridiagonal matrix with size 4×4 with notation as in (36). We consider one step of the QR-method with a shift μ, we drop the iteration count for simplification

$$(T - \mu I) = QR, \quad (T^{'} + \mu I) = RQ. \tag{37}$$

Just like in the explicit form of the QR-method and in the section about Hessenberg matrices, and more specifically theorem 4.4 we want to introduce zeroes in our matrix. This can be done in a similar way as in the proof of theorem 4.4 with reflectors but a more precise alternative way is to do it with rotation matrices. [5] Let G be a rotation matrix defined as

$$G = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$$

where $c = \cos\theta$ and $s = \sin\theta$. We will use a special type of rotation matrix called Givens rotation, see section 5.1 of [5], which we define in the general case as

$$G_{i,j} = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & -s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}.$$

where again $c = \cos\theta$ and $s = \sin\theta$; the upper left c is found at $g_{i,i}$, the other c is found at $g_{j,j}$, s is found at $g_{j,i}$ and $-s$ is found at $g_{i,j}$. For our first Givens rotation, c and s are computed such that

$$G_{1,2}^T \mathbf{t}_1 = \pm \left\| \mathbf{t}_1 \right\|_2 \mathbf{e}_1, \;\; t_1 = (\alpha_1 - \mu I, \beta_1, 0, \ldots, 0)^T. \tag{38}$$

this is similar to the condition given in (35). If we then form $G_{1,2}^T T G_{1,2}$ (eigenvalues are preserved since it is a similarity transformation) the result is a matrix

$$\begin{bmatrix} \bar{\alpha_1} & \beta_1^{'} & \gamma & 0 \\ \beta_1^{'} & \alpha_2^{'} & \beta_2^{'} & 0 \\ \gamma & \beta_2^{'} & \alpha_3 & \beta_3 \\ 0 & 0 & \beta_3 & \alpha_4 \end{bmatrix} = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta_1 & 0 & 0 \\ \beta_1 & \alpha_2 & 0 & 0 \\ 0 & \beta_2 & \alpha_3 & \beta_3 \\ 0 & 0 & \beta_3 & \alpha_4 \end{bmatrix} \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Here we use a bar to denote if the an element will remain unaffected by further rotations in the one iteration and apostrophe to denote a different element at the same place. Next we apply matrices $G_{2,3}$ and $G_{2,3}^T$ computed in a similar way to (38) to zero-out the two new non-zero elements γ and to preserve the tridiagonal form. The result is a matrix

$$G_{2,3}^T G_{1,2}^T T G_{1,2} G_{2,3} = \begin{bmatrix} \bar{\alpha_1} & \bar{\beta_1} & 0 & 0 \\ \bar{\beta_1} & \bar{\alpha_2} & \beta_2^{''} & \gamma \\ 0 & \beta_2^{''} & \alpha_3^{'} & \beta_3^{'} \\ 0 & \gamma & \beta_3^{'} & \alpha_4 \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_2 & s_2 & 0 \\ 0 & -s_2 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{\alpha_1} & \beta_1^{'} & \gamma & 0 \\ \beta_1^{'} & \alpha_2^{'} & \beta_2^{'} & 0 \\ \gamma & \beta_2^{'} & \alpha_3 & \beta_3 \\ 0 & 0 & \beta_3 & \alpha_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_2 & -s_2 & 0 \\ 0 & s_2 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

If we then apply the rotation $G_{3,4}$, the two non-zero elements γ disappear and we obtain a tridiagonal matrix. Notice that our the rotations $G_{1,2}, G_{2,3} \ldots$ "chases" the non-zero elements μ down the diagonal until they disappear, this is called *bulge chasing*. Multiplying the Givens rotations together we obtain a matrix

$$Q = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_2 & -s_2 & 0 \\ 0 & s_2 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c_3 & -s_3 \\ 0 & 0 & s_3 & c_3 \end{bmatrix}.$$

It turns out and it can be shown that the first column of Q, $(c_1, s_1, 0, 0)^T$ is exactly equal to the first column of Q, had it been computed with the explicit shifted method from equation (37). Thus, by the Implicit Q Theorem 4.7 the tridiagonal matrix that result from one step of the of the implicit and explicit version of the QR-method respectively is the same.

Acknowledgement

I would like to thank my thesis advisor Yishao Zhou for her feedback and suggestions that kept my going to the end. I would also like to thank Alexander Berglund and my examiner Salvador Rodriguez-Lopez for taking the time to read my thesis.

References

[1] Peter Arbenz. *Chapter 6: Vector iteration Power method*. URL: `http://people.inf.ethz.ch/arbenz/ewp/Lnotes/2010/chapter6.pdf`.

[2] Peter Arbenz. *Chapter 7: Simultaneous vector or subspace iterations*. URL: `http://people.inf.ethz.ch/arbenz/ewp/Lnotes/2010/chapter7.pdf`.

[3] Åke Björck. *Numerical Methods in Matrix Computations*. 1st ed. Vol. 59. Texts in Applied Mathematics. Springer International Publishing, 2015. DOI: `10.1007/978-3-319-05089-8`.

[4] J. Dongarra and F. Sullivan. “Guest Editors Introduction to the top 10 algorithms”. In: *Computing in Science & Engineering* 2.1 (2000), pp. 22–23. DOI: 10.1109/mcise.2000.814652.

[5] Gene H. Golub and Van Loan Charles F. *Matrix computations*. Vol. 4. The Johns Hopkins University Press, 2013.

[6] Lloyd N. Trefethen and David Bau. *Numerical linear algebra*. Society for Industrial and Applied Mathematics, 1997.

[7] Tai-Lin Wang. “Convergence of the tridiagonal QR algorithm”. In: *Linear Algebra and its Applications* 322.1-3 (2001), pp. 1–17. DOI: 10.1016/s0024-3795(00)00171-3.

[8] David S. Watkins. *Fundamentals of matrix computations*. Vol. 3. Wiley, 2010.

[9] J.h. Wilkinson. “Global convergene of tridiagonal QR algorithm with origin shifts”. In: *Linear Algebra and its Applications* 1.3 (1968), pp. 409–420. DOI: 10.1016/0024-3795(68)90017-7.