Numerics II

Yoann Le Hénaff, Yanyan Shi

Summer semester, 2025
May 28, 2025

Forefront

This document is a translation of the notes taken by Markus Klein in 2009-2010, of the lecture given in German by Prof. Dr. Christian Lubich.

Any mistake is most likely due to YS or YLH, please email us so that we can fix it.

Contents

Chapter 1

Fast Fourier Transform

1.1 Fourier Series

Definition 1.1 – Fourier Transform

Let $(c_n)_{n\in\mathbb{Z}}$ be an absolutely summable sequence of complex numbers, i.e., $\sum_{n\in\mathbb{Z}} |c_n| < \infty$. The Fourier transform of $(c_n)_{n\in\mathbb{Z}}$ is given by:

$$\hat{c}(t) = \sum_{n=-\infty}^{\infty} c_n e^{int}, \quad t \in \mathbb{R}.$$

Proposition 1.1 – Properties of the Fourier Transform

The Fourier transform $\hat{c}(t)$ satisfies:

1. $\hat{c}$ is 2π-periodic.
2. $\hat{c}$ is continuous.
3. It satisfies the orthogonality relation:

$$\frac{1}{2\pi}\int_0^{2\pi} e^{-int}e^{imt}dt = \begin{cases} 1, & m = n, \\ 0, & m \neq n. \end{cases}$$

4. The inverse formula holds:

$$c_n = \frac{1}{2\pi}\int_0^{2\pi} e^{-int}\hat{c}(t)dt.$$

5. The Parseval equation holds:

$$\sum_{n=-\infty}^{\infty} |c_n|^2 = \frac{1}{2\pi}\int_0^{2\pi} |\hat{c}(t)|^2 dt.$$

6. Let (c_n), (d_n) be absolutely summable sequences. Then, for the convolution, we have:

$$(c * d)_n := \sum_{j=-\infty}^{\infty} c_{n-j}d_j$$

defined as the convolution, and the convolution theorem holds:

$$\widehat{c * d}(t) = \hat{c}(t) \cdot \hat{d}(t)$$

Proof. 2. $\sum_{n=-N}^{N} c_n e^{int} \to \hat{c}(t)$ is uniformly convergent for $N \to \infty$ because

$$\left| \hat{c}(t) - \sum_{n=-N}^{N} c_n e^{int} \right| \leq \sum_{|n|>N} |c_n| \to 0$$

and this convergence is independent of t, implying that $\hat{c}$ is continuous.

□

Definition 1.2 – Fourier Coefficients

Let f be a 2π-periodic continuous function. Then for $n \in \mathbb{Z}$,

$$c_n = \frac{1}{2\pi} \int_0^{2\pi} f(t) e^{-int} \, dt$$

is the n-th Fourier coefficient of f, and we denote $c_n = \hat{f}(n)$.

Theorem 1.1 – Calculation of Fourier Coefficients and Estimation

Let f be 2π-periodic and p-times differentiable, with $f^{(p)}$ absolutely integrable (it suffices that $f \in W^{p,1}$). Then the following hold:

1. The n-th Fourier coefficient of $f^{(p)}$ is $(in)^p \cdot c_n$.
2. $c_n = O(|n|^{-p})$, i.e., $|c_n| \leq M \cdot |n|^{-p}$ with

$$M = \frac{1}{2\pi} \int_0^{2\pi} |f^{(p)}(t)| \, dt$$

Proof. 1. We obtain by (multiple) integration by parts:

$$\frac{1}{2\pi} \int_0^{2\pi} f^{(p)}(t) e^{-int} \, dt = -\frac{1}{2\pi} \int_0^{2\pi} f^{(p-1)}(t)(-in) e^{-int} \, dt = (in)^p \frac{1}{2\pi} \int_0^{2\pi} f(t) e^{-int} \, dt$$

where the boundary terms vanish due to periodicity.

2. From the above, we have:

$$|c_n| \cdot |n|^p = \left| \frac{1}{2\pi} \int_0^{2\pi} f^{(p)}(t) e^{-int} \, dt \right| \leq \frac{1}{2\pi} \int_0^{2\pi} |f^{(p)}(t)| \, dt$$

□

Remark 1.1

In particular, (c_n) is absolutely summable if $f \in C^2$ (or $f \in W^{2,1}$). With greater effort, it can also be shown that $f \in C^1$ would suffice.

Theorem 1.2 – Convolution Theorem

Let f, g be continuous and 2π-periodic. Then the convolution of f and g, defined by

$$(f * g)(t) = \frac{1}{2\pi} \int_0^{2\pi} f(t - \tau) g(\tau) \, d\tau$$

is again 2π-periodic and continuous. For the Fourier coefficients of the convolution, we have:

$$\widehat{f * g}(n) = \hat{f}(n) \cdot \hat{g}(n), \quad n \in \mathbb{Z}$$

Proof. The periodicity and continuity are clear from Analysis II. Now we calculate:

$$\begin{aligned}\widehat{f*g}(n) &= \frac{1}{2\pi}\int_0^{2\pi}\left(\frac{1}{2\pi}\int_0^{2\pi} f(t-\tau)g(\tau)\,d\tau\right)e^{-int}\,dt \\ &= \left(\frac{1}{2\pi}\right)^2\int_0^{2\pi}\int_0^{2\pi} f(t-\tau)e^{-in(t-\tau)}\cdot g(\tau)e^{-in\tau}\,dtd\tau\end{aligned}$$

Set $s := t-\tau$, then we obtain:

$$\begin{aligned}&=\frac{1}{2\pi}\int_0^{2\pi} f(s)e^{-ins}ds\cdot\frac{1}{2\pi}\int_0^{2\pi} g(\tau)e^{-in\tau}d\tau \\ &=\hat{f}(n)\cdot\hat{g}(n)\end{aligned}$$

Thus, the claim follows. □

Remark 1.2 – Generalization

The Fourier transform is also defined for $n\in\mathbb{R}$, and the convolution theorem holds for this case as well.

Remark 1.3 – Outlook

Can a continuous function f be recovered from its Fourier coefficients? That is, does

$$f(t) \stackrel{?}{=} \sum_{n=-\infty}^{\infty} c_n e^{int}$$

We will see that without additional conditions on f, the sequence (c_n) is not absolutely summable. Even if (c_n) is absolutely summable, does $f(t) = \hat{c}(t)$? We will later see that this is indeed the case.

Theorem 1.3 – Fejér's Theorem (Fejér, 1904)

Let $f:\mathbb{R}\to\mathbb{C}$ be continuous and 2π-periodic with Fourier coefficients

$$(c_n)_{n\in\mathbb{Z}} = \frac{1}{2\pi}\int_0^{2\pi} e^{-int}f(t)\,dt$$

Then the following holds:

$$\sum_{k=-n}^{n}\left(1-\frac{|k|}{n+1}\right)c_k e^{ikt}\to f(t)\quad\text{uniformly in } t$$

Proof. 1. We consider the convolution with the convolution theorem:

$$(e^{in\tau}*f)(t) = \frac{1}{2\pi}\int_0^{2\pi} e^{in(t-\tau)}f(\tau)\,d\tau = e^{int}c_n$$

Thus, by linearity:

$$\sum_{j=-n}^{n}\left(1-\frac{|j|}{n+1}\right)e^{ijt}c_j := (K_n*f)(t)$$

with the so-called Fejér kernel K_n, for which we have:

$$K_n(t) = \sum_{j=-n}^{n}\left(1-\frac{|j|}{n+1}\right)e^{ijt}$$

2. We show that the reduction formula holds:

$$K_n(t) = \frac{1}{n+1}\left(\frac{\sin\left(\frac{n+1}{2}t\right)}{\sin\left(\frac{t}{2}\right)}\right)^2$$

This is shown using the definition of the sine function, where we use the identity:

$$\sin^2\left(\frac{t}{2}\right) = \frac{1}{2}(1-\cos t) = -\frac{1}{4}e^{-it} + \frac{1}{2} - \frac{1}{4}e^{it}$$

Direct computation then gives the identity:

$$\left(-\frac{1}{4}e^{it} + \frac{1}{2} - \frac{1}{4}e^{it}\right)\sum_{j=-n}^{n}\left(1-\frac{|j|}{n+1}\right)e^{ijt} = \frac{1}{n+1}\left(-\frac{1}{4}e^{-i(n+t)t} + \frac{1}{2} - \frac{1}{4}e^{i(n+1)t}\right)$$

This simplifies to:

$$= \frac{1}{n+1}\sin^2\left(\frac{(n+1)t}{2}\right)$$

3. We now consider some properties:

   (a) Due to the orthogonality relation:

$$\frac{1}{2\pi}\int_0^{2\pi} K_n(t)\,dt = \sum_{j=-n}^{n}\left(1-\frac{|j|}{n+1}\right)\frac{1}{2\pi}\underbrace{\int_0^{2\pi} e^{ijt}}_{=\delta_{j0}}\,dt = 1$$

   (b) $K_n(t) \geq 0$ $\forall t$, n due to the reduction formula.

   (c) $\forall \delta \in (0,\pi)$, we have:

$$\lim_{n\to\infty}\int_\delta^{2\pi-\delta} K_n(t)\,dt = 0$$

   because

$$K_n(t) \leq \frac{1}{n+1}\frac{1}{\sin^2\left(\frac{\delta}{2}\right)} \to 0 \quad \text{as} \quad n\to\infty$$

4. It follows that:

$$(K_n * f)(t) - f(t) = \frac{1}{2\pi}\int_0^{2\pi} K_n(\tau)\left(f(t-\tau)-f(t)\right)d\tau$$

and thus:

$$\int_0^{2\pi} K_n(\tau)\left(f(t-\tau)-f(t)\right)d\tau = I_1 + I_2$$

where

$$I_1 = \int_{-\delta}^{\delta}\left(f(t-\tau)-f(t)\right)K_n(\tau)d\tau \quad \text{and} \quad I_2 = \int_\delta^{2\pi-\delta}\left(f(t-\tau)-f(t)\right)K_n(\tau)d\tau$$

We estimate:

$$I_1 \leq \max_{|\tau|\leq\delta}|f(t-\tau)-f(t)| \cdot \frac{1}{2\pi}\int_{-\delta}^{\delta} K_n(\tau)d\tau \leq \frac{1}{2\pi}\int_{-\pi}^{\pi} K_n(\tau)d\tau = 1$$

We estimate the second part:

$$I_2 \leq \frac{1}{2}\cdot 2\max_{0\leq\theta\leq 2\pi}|f(\theta)| \cdot \underbrace{\int_\delta^{2\pi-\delta}|K_n(\tau)|d\tau}_{\to 0}$$

Thus, for $n \to \infty$, we have:

$$\limsup_{n\to\infty} |(K_n * f)(t) - f(t)| \leq \max_{|\tau|\leq\delta} |f(t-\tau) - f(t)| \quad \text{for} \quad \forall\delta \in (0,\pi)$$

As $\delta \to 0$, the term tends to 0 due to continuity. Since f is uniformly continuous, the convergence is uniform. Thus, $K_n * f(t) \to f(t)$ uniformly in t.

□

Remark 1.4

It should be noted that

$$\sum_{k=-n}^{n} \left(1 - \frac{|k|}{n+1}\right) c_k e^{int} = \frac{1}{n+1} \sum_{m=0}^{n} \left(\sum_{k=-m}^{m} c_k e^{ikt}\right)$$

is the arithmetic mean of all partial sums, where the partial sums need not necessarily converge.

Theorem 1.4 – Uniqueness Theorem

Let f and g be 2π-periodic and continuous functions with the same Fourier coefficients. Then $f = g$.

Proof. We have

$$f(t) = \lim_{n\to\infty} \sum_{j=-n}^{n} \left(1 - \frac{|j|}{n+1}\right) c_j e^{ijt} = g(t) \quad \forall t.$$

□

Theorem 1.5 – Representation via Fourier Coefficients

Let f be a 2π-periodic and continuous function. If the Fourier coefficients are absolutely summable, then:

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{int} \quad \forall t.$$

Proof. The function f and the series have the same Fourier coefficients, and by the uniqueness theorem, both functions must coincide. □

Remark 1.5

This is particularly true if f is once continuously differentiable (i.e., $f \in W^{1,1}$).

Theorem 1.6 – Interpretation of Fejér's Theorem

Every continuous 2π-periodic function can be uniformly approximated by trigonometric polynomials.

Proof. This follows directly from the statement of Fejér's theorem. □

Theorem 1.7 – Weierstrass Approximation Theorem

Every continuous function on a compact interval $g : [a,b] \to \mathbb{R}$ can be uniformly approximated by polynomials, i.e., $\forall\epsilon > 0$, there exists a polynomial p such that

$$\max_{x\in[a,b]} |p(x) - g(x)| < \epsilon.$$

Proof. Without loss of generality, assume that $[a, b] = [-1, 1]$. Set $f(t) = g(x)$ for $x = \cos t$, or more precisely $t = \arccos x \in [0, \pi]$. Next, extend f as an even function, i.e., $f(-t) = f(t)$, and note that f is continuous.

For even functions, we have $c_{-n} = c_n$, because:

$$c_{-n} = \frac{1}{2\pi}\int_0^{2\pi} e^{int} f(t)\,dt = \frac{1}{2\pi}\int_{-2\pi}^{0} e^{-in\tau} f(-\tau)\,d\tau = \frac{1}{2\pi}\int_0^{2\pi} e^{-in\tau} \underbrace{f(-\tau)}_{f(\tau)}\,d\tau = c_n$$

due to the transformation of variables and the 2π-periodicity.

Furthermore, by Fejér's theorem, we know:

$$\sum_{j=-n}^{n}\left(1 - \frac{|j|}{n+1}\right) c_j e^{ijt} = c_0 + 2\sum_{k=1}^{n}\left(1 - \frac{k}{n+1}\right) c_k \cos(kt).$$

We also know that the left-hand side converges uniformly to $f(t)$. Since $\cos(k \arccos x) = T_k(x)$, the k-th Chebyshev polynomial, we obtain:

$$c_0 + 2\sum_{k=1}^{n}\left(1 - \frac{k}{n+1}\right) c_k T_k(x) \to g(x).$$

□

Remark 1.6

The convergence in the last theorem can be arbitrarily slow. It will be faster if the coefficients decay rapidly, which is particularly the case when the function is frequently differentiable.

1.2 Discrete Fourier Transform

We consider finite sequences $x = (x_0, \ldots, x_{N-1}) \in \mathbb{C}^N$ periodically extended to arbitrary integer indices (if needed). That is, we set $x_k = x_\ell$ if $k \equiv \ell \mod N$.

Definition 1.3 – Discrete Fourier Transform

The mapping $\mathcal{F}_N : \mathbb{C}^N \to \mathbb{C}^N$ is defined by $\mathcal{F}_N x = \hat{x}$ with

$$\hat{x}_k = \sum_{j=0}^{N-1} w_N^{k\cdot j} x_j,$$

where $w_N = e^{i\frac{2\pi}{N}}$ is the primitive N-th complex root of unity, i.e., $w_N^N = 1$.

Remark 1.7

If N is clear from context, we simply write w for the root of unity.

Remark 1.8 – Computational Complexity

The direct computation of the discrete Fourier transform requires about N^2 operations (multiplications and additions). We will later see that the Fast Fourier Transform (FFT) requires about $N \log_2 N$ operations if $N = 2^L$.

Lemma 1.1 – Orthogonality Relation of the Discrete Fourier Transform

It holds that:

$$\sum_{k=0}^{N-1} w_N^{k\ell}\bar{w}_N^{km} = \begin{cases} N, & \ell \equiv m \mod N \\ 0, & \text{otherwise} \end{cases}$$

Proof. Let $\bar{w} = w^{-1}$. For $\ell \equiv m$,

$$\sum_{k=0}^{N-1} 1 = N.$$

Otherwise, it follows that

$$\sum_{k=0}^{N-1} w_N^{k\ell}w_N^{-km} = \sum_{k=0}^{N-1} w_N^{k(\ell-m)} = \frac{1-(w_N^{\ell-m})^N}{1-w_N^{\ell-m}} = 0.$$

□

Theorem 1.8 – Parseval's Equation

Let $\mathbb{C}^N$ be equipped with the Euclidean norm. Then we have:

$$\frac{1}{\sqrt{N}}\|\mathcal{F}_N x\| = \|x\|, \quad \forall x \in \mathbb{C}^N,$$

i.e., this transformation is an isometry/unitary mapping. Explicitly, this means:

$$\frac{1}{N}\sum_{k=0}^{N-1} |\hat{x}_k|^2 = \sum_{j=0}^{N-1} |x_j|^2.$$

Proof. We compute:

$$\begin{aligned}
\|\hat{x}\|^2 &= \sum_{k=0}^{N-1} \hat{x}_k\bar{\hat{x}}_k \\
&= \sum_{k=0}^{N-1}\sum_{\ell=0}^{N-1} w_N^{k\ell}x_\ell \sum_{m=0}^{N-1} \bar{w}_N^{km}\bar{x}_m \\
&= \sum_\ell \sum_m x_\ell \bar{x}_m \underbrace{\sum_k w_N^{k\ell}\bar{w}_N^{km}}_{=N\delta_{lm}} \\
&= N\sum_{\ell=0}^{N-1} x_\ell\bar{x}_\ell = N\|x\|^2,
\end{aligned}$$

which proves the equality. □

Notation 1.1 – Fourier Transform Matrix

The mapping $\mathcal{F}_N : \mathbb{C}^N \to \mathbb{C}^N$ is linear and represented by the matrix $(w_N^{kj})_{k,j=0}^{N-1}$. Similarly, we define:

$$\bar{\mathcal{F}}_N : \mathbb{C}^N \to \mathbb{C}^N, \quad \text{with matrix } (\bar{w}_N^{kj})_{k,j=0}^{N-1} = (w_N^{-kj}).$$

Theorem 1.9 – Inverse Discrete Fourier Transform

We have:

$$\mathcal{F}_N^{-1} = \frac{1}{N}\bar{\mathcal{F}}_N,$$

or explicitly,

$$x_j = \frac{1}{N} \sum_{k=0}^{N-1} \bar{w}_N^{kj} \hat{x}_k, \quad \text{for } j = 0, \ldots, N-1.$$

Proof. From the orthogonality lemma, we know that $\mathcal{F}_N \cdot \bar{\mathcal{F}}_N = NI_N$. Explicitly,

$$\sum_{k=0}^{N-1} \bar{w}_N^{kj} \sum_{\ell=0}^{N-1} w_N^{k\ell} x_\ell = \sum_{\ell=0}^{N-1} x_\ell \sum_{k=0}^{N-1} \bar{w}_N^{kj} w_N^{k\ell} = Nx_j.$$

Thus, dividing by N gives the result. □

Definition 1.4 – Pointwise Product

For sequences $x, y \in \mathbb{C}^N$, the pointwise multiplication is defined as:

$$(x \cdot y)_k = x_k y_k.$$

Definition 1.5 – Convolution Multiplication

For N-periodic sequences $x, y \in \mathbb{C}^N$, we define the convolution product $x * y \in \mathbb{C}^N$ as:

$$(x * y)_k = \sum_{j=0}^{N-1} x_{k-j} y_j.$$

Theorem 1.10 – Convolution Theorem

The Fourier transform converts convolution into pointwise multiplication:

$$\mathcal{F}_N(x * y) = (\mathcal{F}_N x) \cdot (\mathcal{F}_N y).$$

Proof. We consider the m-th component of the left-hand side:

$$(\mathcal{F}_N(x * y))_m = \sum_{k=0}^{N-1} w_N^{mk} \sum_{j=0}^{N-1} x_{k-j} y_j.$$

Using index shift $k - j = \ell$:

$$\sum_{j=0}^{N-1} \sum_{\ell=0}^{N-1} w_N^{m(\ell+j)} x_\ell y_j.$$

Since sequences are N-periodic,

$$\sum_{j=0}^{N-1} w_N^{mj} y_j \sum_{\ell=0}^{N-1} w_N^{m\ell} x_\ell = \hat{y}_m \hat{x}_m = (\mathcal{F}_N y) \cdot (\mathcal{F}_N x),$$

proving the claim. □

Corollary 1.1 – Properties of Convolution

Since pointwise multiplication is commutative and associative, convolution is also commutative and associative.

Corollary 1.2 – Direct Computation of Convolution

We have:

$$x * y = \frac{1}{N}\bar{\mathcal{F}}_N(\mathcal{F}_N x \cdot \mathcal{F}_N y),$$

which provides an efficient way to compute convolution using the Fourier transform.

Remark 1.9 – Computational Effort for Convolution

A direct computation of the convolution requires approximately N^2 multiplications and additions. With the FFT, we need $N \log_2 N$ operations for the transformation, only N operations for the pointwise multiplication, and another $N \log_2 N$ operations for the inverse FFT. Thus, using the FFT, we require only $3N \log_2 N + 2N$ operations.

1.3 Fast Fourier Transform (FFT)

Given a vector $x = (x_0, \dots, x_{N-1}) \in \mathbb{C}^N$, we aim to compute $\hat{x} = \mathcal{F}_N x$.

Theorem 1.11 – Reduction Formula

We split the vector x into two vectors u and v, where u contains the even indices and v the odd indices:

$$x = (u_0, v_0, u_1, v_1, \dots, u_{N/2-1}, v_{N/2-1}) \in \mathbb{C}^N.$$

Then, for $k = 0, \dots, N/2 - 1$, we have:

$$\begin{aligned}(\mathcal{F}_N x)_k &= (F_{N/2}u)_k + w_N^k (F_{N/2}v)_k,\\ (\mathcal{F}_N x)_{k+N/2} &= (F_{N/2}u)_k - w_N^k (F_{N/2}v)_k.\end{aligned}$$

Proof. We compute the k-th entry of $\mathcal{F}_N x$:

$$(\mathcal{F}_N x)_k = \sum_{\ell=0}^{N-1} w_N^{k\ell} x_\ell = \sum_{j=0}^{N/2-1} w_N^{k(2j)} u_j + \sum_{j=0}^{N/2-1} w_N^{k(2j+1)} v_j.$$

Since $w_N^{2j} = w_{N/2}^j$, this simplifies to:

$$(\mathcal{F}_N x)_k = \sum_{j=0}^{N/2-1} w_{N/2}^{kj} u_j + w_N^k \sum_{j=0}^{N/2-1} w_{N/2}^{kj} v_j = (F_{N/2}u)_k + w_N^k (F_{N/2}v)_k.$$

Due to periodicity, we obtain the second equation using $w_N^{k+N/2} = w_N^k w_N^{N/2} = -w_N^k$. □

Remark 1.10

If $F_{N/2}u$ and $F_{N/2}v$ are known, we require $N/2$ multiplications and N additions to compute $\mathcal{F}_N x$.

Remark 1.11 – Historical Note

This formula dates back to Cooley-Tukey (1965). Similar ideas were found by Danilson and Lanzos (1942), Runge (1925), Gauss, and even Caesar's "divide et impera."

Remark 1.12 – Algorithm Complexity

If $N = 2^L$, we can recursively divide the vector L times until we obtain vectors of length 1. This results in a computational cost of $L \cdot N/2$ multiplications, where $L = \log_2 N$.

Theorem 1.12 – Computational Complexity of FFT

For $N = 2^L$, computing $\mathcal{F}_N x$ requires:

1. $\frac{1}{2} N \log_2 N$ complex multiplications,
2. $N \log_2 N$ complex additions.

Remark 1.13 – Order of Elements

During the execution of the FFT algorithm, the order of elements is permuted. Using binary representation, we obtain the order by reversing the binary digits:

Input (Decimal)	Input (Binary)	Output (Binary)	Output (Decimal)
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Thus, the order is obtained by mirroring the binary digits.

Remark 1.14 – Comparison with Direct Computation

We summarize key values that highlight the advantage of FFT:

N	N^2	$N \log_2 N$	Quotient
$2^5 = 32$	10^3	160	6.4
$2^{10} \approx 10^3$	10^6	10^4	100
$2^{20} \approx 10^6$	10^{12}	$2 \cdot 10^7$	$50,000$

1.4 Approximation of Fourier coefficients, trigonometric interpolation

We would like to clarify the relationship between the discrete and continuous Fourier transforms. The Fourier coefficients for a 2π-periodic, continuous function f are given by:

$$\hat{f}(n) = c_n = \frac{1}{2\pi} \int_0^{2\pi} e^{-int} f(t)\, dt.$$

We approximate the integral using the trapezoidal rule with step size $h = \frac{2\pi}{N}$, obtaining:

$$\hat{f}_N(n) = \frac{1}{2\pi} \left(\frac{h}{2} e^{-in0} f(0) + h e^{-inh} f(h) + \cdots + h e^{-in(N-1)h} f((N-1)h) + \frac{h}{2} e^{-inNh} f(Nh) \right).$$

Due to the periodicity, we have $he^{-in0} f(0) = he^{-inNh} f(Nh)$, so the trapezoidal rule effectively becomes the rectangular rule. Thus, we obtain:

$$\hat{f}_N(n) = \frac{1}{N} \sum_{j=0}^{N-1} e^{-inj\frac{2\pi}{N}} f(t_j) = \frac{1}{N} \sum_{j=0}^{N-1} w_N^{-nj} f(t_j),$$

where $t_j = jh = j\frac{2\pi}{N}$. Therefore, $\hat{f}_N(n)$ is N-periodic as a vector. This leads to the following representation:

$$\hat{f}_N = \mathcal{F}_N^{-1} (f(t_j))_{j=0}^{N-1},$$

which can be computed using the Fast Fourier Transform (FFT).

Theorem 1.13 – Aliasing Formula

Let $(\hat{f}(n))_{n\in\mathbb{Z}}$ be absolutely summable. Then,

$$\hat{f}_N(n) - \hat{f}(n) = \sum_{0\neq\ell\in\mathbb{Z}} \hat{f}(n+\ell N).$$

Proof. From the previous chapter, we know that if the Fourier coefficients are absolutely summable, we have:

$$f(t) = \sum_{k=-\infty}^{\infty} \hat{f}(k)e^{ikt}.$$

Now, consider:

$$\hat{f}_N(n) = \frac{1}{N}\sum_{j=0}^{N-1} w_N^{-nj} \sum_{k=-\infty}^{\infty} \hat{f}(k)e^{ikj\frac{2\pi}{N}} = \sum_{k=-\infty}^{\infty} \hat{f}(k)\frac{1}{N}\sum_{j=0}^{N-1} w_N^{-nj}w_N^{kj},$$

where $w_N = e^{i\frac{2\pi}{N}}$. The inner sum can be simplified as follows:

$$\frac{1}{N}\sum_{j=0}^{N-1} w^{-nj}w^{kj} = \begin{cases} 1 & \text{if } k = n \mod N, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, we obtain:

$$\hat{f}_N(n) = \sum_{\ell=-\infty}^{\infty} \hat{f}(n+\ell N),$$

and for $\ell = 0$, this reduces to $\hat{f}(n)$, which confirms the statement. □

Corollary 1.3 – Error Estimation for the Numerical Fourier Coefficients

Let f be p-times continuously differentiable (with $p \geq 2$) and 2π-periodic. Then, for $|n| \leq \frac{N}{2}$, we have:

$$|\hat{f}_N(n) - \hat{f}(n)| \leq C \cdot N^{-p}.$$

Proof. From the previous section, we know that:

$$|\hat{f}(n)| \leq C_0|n|^{-p},$$

with

$$C_0 = \frac{1}{2\pi}\int_0^{2\pi} |f^{(p)}(t)|\,dt.$$

On the other hand, using the aliasing formula, we have:

$$|\hat{f}_N(n) - \hat{f}(n)| = \sum_{\ell\neq 0} |\hat{f}(n+\ell N)| \leq \sum_{\ell\neq 0} C_0 \cdot |n+\ell N|^{-p}.$$

For $|n| \leq \frac{N}{2}$, we have $|n+\ell N| \geq \frac{N}{2}$. So $|n+\ell N| < N$ for only one $\ell \neq 0$. This is

$$l = \begin{cases} -1 & n > 0, \\ 1 & n < 0. \end{cases}$$

For every interval $I = [N, 2N), [2N, 3N), \cdots$ and $(-2N, -N], (-3N, -2N], \cdots$ there is exactly one ℓ such that $n + \ell N \in I$. This gives us the following in total:

$$\sum_{\ell\neq 0} |n+\ell N|^{-p} \leq (N/2)^{-p} + 2\sum_{m=1}^{\infty} (mN)^{-p} \leq c_p \cdot N^{-p},$$

where c_p is a constant derived from the converging sums. Hence, $C = C_0c_p$ and the result follows. □

Remark 1.15

For the special case $n = 0$, we obtain for the grid spacing $h = \frac{2\pi}{N}$:

$$\frac{h}{2\pi}\sum_{j=0}^{N-1} f(t_j) - \frac{1}{2\pi}\int_0^{2\pi} f(t)\,dt = O(h^p).$$

Thus, the accuracy of the trapezoidal rule depends on the smoothness of the function, i.e., the trapezoidal rule is very accurate for smooth and periodic integrands.

Theorem 1.14 – Trigonometric Interpolation

The trigonometric polynomial

$$f_N(t) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}}{}' \hat{f}_N(n)e^{int}$$

interpolates f at the points $t_j = j\frac{2\pi}{N}$ for $j = 0, 1, \dots, N-1$.

Proof. Due to the periodicity, we have (in particular $e^{int_j} = w_N^{nj}$):

$$f_N(t_j) = \sum_{n=0}^{N-1} \hat{f}_N(n) w_N^{nj}.$$

We have used that the left and right boundary terms are equal, so the $\sum'$ term vanishes. This formula means that

$$f_N(t_j))_{j=0}^{N-1} = \mathcal{F}_N\left(\hat{f}_N(n)\right)_{n=0}^{N-1} = \mathcal{F}_N\mathcal{F}_N^{-1}(f(t_j))_{j=0}^{N-1} = (f(t_j))_{j=0}^{N-1}.$$

Thus, $f_N(t)$ already interpolates the grid points t_j. □

Notation $\sum'$ means that the first and the last term are each taken with the factor $\frac{1}{2}$, i.e., for our sum we obtain:

$$\sum_{n=-\frac{N}{2}}^{\frac{N}{2}}{}' \hat{f}_N(n) := \frac{1}{2}\hat{f}_N\left(-\frac{N}{2}\right) + \hat{f}_N\left(-\frac{N}{2}+1\right) + \dots + \hat{f}_N\left(\frac{N}{2}-1\right) + \frac{1}{2}\hat{f}_N\left(\frac{N}{2}\right)$$

This is the interpolation at the nodes $t_j = j\frac{2\pi}{N}$.
It would also be permissible to form one of the following sums instead:

$$\sum_{n=-\frac{N}{2}+1}^{\frac{N}{2}}, \quad \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1}, \quad \sum_{n=-\frac{N}{2}+k}^{\frac{N}{2}-1+k}$$

Using this method, we could also prove the theorem. However, we will see that our variant is the least oscillatory due to symmetry reasons.

Algorithm 1.1 – Trigonometric Interpolation

We assume that f is 2π-periodic. Then we proceed as follows:

1. Compute $f(t_j)$ for $j = 0, \dots, N-1$ and $t_j = j \cdot \frac{2\pi}{N}$.
2. Compute $\hat{f}_N = \mathcal{F}_N^{-1}(f(t_j))_{j=0}^{N-1}$ using the FFT with a computational complexity of $O(N \log N)$ operations.
3. Obtain the trigonometric interpolation polynomial as in the previous theorem.

Theorem 1.15 – Error Estimate for Trigonometric Interpolation

If $(\hat{f}(n))_n$ is absolutely summable, then:

$$|f_N(t) - f(t)| \leq 2 \sum_{|n| \geq \frac{N}{2}} {}' |\hat{f}(n)|$$

Proof. We use the aliasing formula and obtain:

$$\begin{aligned} |f_N(t) - f(t)| &= \left| \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} {}' \hat{f}_N(n) e^{int} - \sum_{n=-\infty}^{\infty} \hat{f}(n) e^{int} \right| \\ &= \left| \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} {}' (\hat{f}_N(n) - \hat{f}(n)) e^{int} - \sum_{|n| \geq \frac{N}{2}} {}' \hat{f}(n) e^{int} \right| \end{aligned}$$

Using the aliasing formula:

$$\begin{aligned} &= \left| \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} {}' \sum_{\ell \neq 0} \hat{f}(n + \ell N) e^{int} - \sum_{|n| \geq \frac{N}{2}} {}' \hat{f}(n) e^{int} \right| \\ &\leq \sum_{|m| \geq \frac{N}{2}} {}' |\hat{f}(m)| + \sum_{|n| \geq \frac{N}{2}} {}' |\hat{f}(n)| \\ &\leq 2 \sum_{|n| \geq \frac{N}{2}} {}' |\hat{f}(n)| \end{aligned}$$

which proves the theorem. □

Remark 1.16

If $f \in C^p$ with $p \geq 2$, then it is well known that $\hat{f}(n) = O(|n|^{-p})$. Thus, we obtain: $|f_N(t) - f(t)| = O(N^{-p+1})$.

1.5 Inverse Convolution Problem, Regularization, Filtering

Motivation (Problem Statement)
An input signal $u = u(x)$ for $x \in \mathbb{R}$ enters a device, and a signal b is measured, which no longer corresponds to the input signal.

We make the following assumptions:

1. *$u \mapsto b$ is linear.*
2. *$u \mapsto b$ is shift-invariant.*

We know that such mappings are convolutions.

Remark 1.17

This problem arises in many areas, such as image or signal processing.

Model Construction (Device Model)
We consider the following model:

$$\int_{-\infty}^{\infty} a(x-y) u(y) \, dy + \varepsilon(x) = b(x)$$

where $u(y)$ is the unknown input signal, $b(x)$ is the observed signal, and the term $a(x-y)$ represents the device function. The term $\varepsilon(x)$ represents the noise or errors in the model, including measurement errors, rounding errors, and model errors. This noise is typically unknown, but we may know an upper bound for it, so that $|\varepsilon(x)| \leq M$ pointwise or in the quadratic mean, i.e. $\int_{-\infty}^{\infty} |\varepsilon(x)|^2 \, dx \leq M$ or similar. We aim to reconstruct the input signal u from the observed signal.

Remark 1.18 – Reduction of the problem

In general, a, u, and b have compact support, so that $\text{supp}(a) = \overline{\{x : a(x) \neq 0\}}$. After a variable transformation, we can assume without loss of generality that:

$$\text{supp}(a) \subseteq \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \quad \text{supp}(u) \subseteq \left[-\frac{\pi}{2}, \frac{\pi}{2}\right].$$

This implies:

$$\text{supp}(b - \varepsilon) \subseteq [-\pi, \pi]$$

since if $b(x) \neq \varepsilon(x)$, then $\exists y$ such that $a(x-y) \cdot u(y) \neq 0$, and thus $x - y \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, and therefore $x \in [-\pi, \pi]$.

We now extend a, u, ε, and b to be 2π-periodic on $\mathbb{R}$. This reduces the problem to:

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} a(x-y)u(y) \, dy + \varepsilon(x) = b(x) \quad \text{for } x \in [-\pi, \pi],$$

which we can express briefly as:

$$a * u + \varepsilon = b$$

or for the linear operator $Au = a * u$, we have:

$$Au + \varepsilon = b$$

Remark 1.19

The "usual" solution method is to solve the linear system $Au + \varepsilon = b$ in $\mathbb{R}^n$ by neglecting the disturbance ε and solving $Av = b$ using standard methods. Then the error $v - u = A^{-1}\varepsilon$ holds, provided $\|A^{-1}\varepsilon\| \leq \|v\|$, which is not necessarily true if the matrix is ill-conditioned.
Unfortunately, this is the case here.

Reminder (Convolution Theorem) It is known that: $\widehat{a * u}(n) = \hat{a}(n)\hat{u}(n)$

Remark 1.20

For the Fourier coefficients, we have:

$$\hat{a}(n)\hat{u}(n) + \hat{\varepsilon}(n) = \hat{b}(n) \quad \text{for } n \in \mathbb{Z}$$

Thus,

$$\hat{u}(n) = \frac{\hat{b}(n)}{\hat{a}(n)} - \frac{\hat{\varepsilon}(n)}{\hat{a}(n)} = \hat{v}(n) - \frac{\hat{\varepsilon}(n)}{\hat{a}(n)}$$

where $\hat{v}(n)$ solves $\hat{a}(n)\hat{u}(n) = \hat{b}(n)$.
However, since the Fourier coefficients of ε remain approximately constant, while the Fourier coefficients of a decay rapidly (if a is smooth), the expression at the back will dominate the term, leading to the result that

$$u(x) = \sum_{n=-\infty}^{\infty} \hat{u}(n)e^{inx} = \sum_{n=-\infty}^{\infty} \hat{v}(n)e^{inx} - \sum_{n=-\infty}^{\infty} \frac{\hat{\varepsilon}(n)}{\hat{a}(n)} e^{inx}$$

For large n, we observe catastrophic error amplification, meaning the problem is ill-conditioned. This is also referred to as a poorly posed problem, or an ill-posed problem.

Remark 1.21 – Alternative Approach (Minimization Problem)

We do not want to solve $Au = b$, but rather we want to minimize $\|Au - b\| \leq \|\varepsilon\|$, i.e., if $\|\varepsilon\| \approx \delta$, we require that $\|Au - b\| \leq \|\delta\|$.
In general, there are infinitely many such u that satisfy this. We wish to choose u such that $\|u''\|$ is minimal (e.g., using cubic splines). Alternatively, we might minimize $\|u\|$. Thus, we are looking for the smoothest or smallest possible u, which leads to a minimization problem.

Remark 1.22

We assume that, unless stated otherwise, we consider the L^2-norm, i.e.

$$\|f\| = \|f\|_{L^2} := \left(\frac{1}{2\pi}\int_{-\pi}^{\pi} |f(x)|^2 \, dx\right)^{1/2}.$$

Theorem 1.16 – General Form of the Minimization Problem

We want to minimize $\|Lu\|$ and assume that L is linear, e.g., $Lu = u''$. Additionally, we require the constraint $\|Au - b\| \leq \delta$.
The minimum is achieved for $\|Au - b\| = \delta$.

Proof. Assume this is not the case, i.e., $\|Lu\|$ is minimal for some u with $\|Au - b\| < \delta$. Consider $\tilde{u} = (1 - \rho)u$ for some $\rho > 0$. Since L is linear, we have:

$$\|L\tilde{u}\| = (1 - \rho)\|Lu\| < \|Lu\|$$

and for $\tilde{u}$, we get:

$$\|A\tilde{u} - b\| = \|(1 - \rho)(Au - b) - \rho b\| \leq (1 - \rho)\|Au - b\| + \rho\|b\| \leq \delta$$

This leads to a contradiction for sufficiently small ρ, so the minimum must occur for $\|Au - b\| = \delta$. □

Remark 1.23

In practice, there are often additional constraints, such as considering only solutions that are monotonic or positive.

Remark 1.24

It is a reasonable assumption that $\|b\| > \delta$, otherwise, the observed signal would be weaker than the noise. This directly implies that $u \neq 0$, and hence the problem cannot be trivially solved.

Definition 1.6 – Tikhonov Regularization

We consider a fixed $\alpha > 0$ as a regularization parameter. We solve the minimization problem without the constraint:

$$\|Au - b\|^2 + \alpha\|Lu\|^2 = \min$$

As $\alpha \to 0$, we get $\|Au - b\| = 0$, but $\|Lu\|$ becomes arbitrarily large. As $\alpha \to \infty$, we get $\|Lu\| = 0$, but $\|Au - b\|$ becomes arbitrarily large. The optimal α should be chosen so that $\|Au - b\| \approx \|\varepsilon\|$, if ε is known. Otherwise, it must be determined empirically until the result "looks good".

Remark 1.25

We now clarify the connection between the minimization and regularization problems: We restrict ourselves to a finite-dimensional problem, i.e., $b \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$. Then A, L are matrices, and $\|\cdot\|$ denotes the Euclidean norm. We have:

$$\|Lu\|_2^2 = u^T L^T Lu, \quad \|Au-b\|_2^2 = (Au-b)^T(Au-b) = u^T A^T Au - 2u^T A^T b + b^T b.$$

The general solution $f(u) = \min$ and $g(u) = 0$ satisfies:

$$f'(u) + g'(u)^T\lambda = 0, \quad g(u) = 0,$$

where λ is a Lagrange multiplier. We then obtain the following two conditions:

$$2L^T Lu + (2A^T Au - 2A^T b)^T\lambda = 0, \quad \lambda \in \mathbb{R}, \quad \|Au-b\|_2^2 - \delta^2 = 0.$$

Thus, we have $(L^T L + \lambda A^T A)u = \lambda A^T b$. For a fixed λ, this is the solution to a minimization problem without constraints:

$$\|Lu\|_2^2 + \lambda\|Au-b\|_2^2 = \min, \quad \text{so for } \alpha = \frac{1}{\lambda}, \quad \text{we have : } \|Au-b\|_2^2 + \alpha\|Lu\|_2^2 = \min.$$

Theorem 1.17 – Tikhonov Regularization

Let $Au = a * u$ and $Lu = u^{(p)}$. The solution to the minimization problem for $\alpha > 0$ as a given regularization parameter is:

$$\|a * u - b\|_2^2 + \alpha\|u^{(p)}\|_2^2 = \min,$$

for u a 2π-periodic function with a square-integrable p-th derivative, and is given by the Fourier coefficients:

$$\hat{u}(n) = \begin{cases} \frac{|\hat{a}(n)|^2}{|\hat{a}(n)|^2 + \alpha n^{2p}} \cdot \frac{\hat{b}(n)}{\hat{a}(n)} & \text{if } \hat{a}(n) \neq 0, \\ 0 & \text{if } \hat{a}(n) = 0. \end{cases}$$

We define the regularization filter $\Phi_\alpha(n)$:

$$\Phi_\alpha(n) := \frac{|\hat{a}(n)|^2}{|\hat{a}(n)|^2 + \alpha n^{2p}}.$$

Proof. By the Parseval formula, the regularization problem is equivalent to:

$$\sum_{n=-\infty}^{\infty} \underbrace{\left(\left|\hat{a}(n)\hat{u}(n) - \hat{b}(n)\right|^2 + \alpha n^{2p}|\hat{u}(n)|^2\right)}_{=:\Lambda_n} = \min,$$

where we have used the convolution theorem and the fact that $\widehat{u^{(p)}}(n) = (in)^p\hat{u}(n)$. The expression becomes minimal if each individual summand is minimized.

For the n-th summand, which we define as Λ_n, we compute:

$$\begin{aligned} \Lambda_n &= |\hat{a}(n)|^2|\hat{u}(n)|^2 - \hat{a}(n)\hat{u}(n)\overline{\hat{b}(n)} - \overline{\hat{a}(n)\hat{u}(n)}\hat{b}(n) + |\hat{b}(n)|^2 + \alpha n^{2p}|\hat{u}(n)|^2 \\ &= \underbrace{\left(|\hat{a}(n)|^2 + \alpha n^{2p}\right)}_{=:r} \cdot \underbrace{|\hat{u}(n)|^2}_{=:z^2} - 2\mathrm{Re}\left(\underbrace{\overline{\hat{u}(n)}}_{=:\bar{z}}\underbrace{\overline{\hat{a}(n)} \cdot \hat{b}(n)}_{=:s}\right) + |\hat{b}(n)|^2. \end{aligned}$$

Thus, we must have:

$$r|z|^2 - 2\mathrm{Re}(\bar{z}s) = \min, \quad \text{or for} \quad q = \frac{s}{r} \quad \text{then applies} \quad |z|^2 - 2\mathrm{Re}(\bar{z}q) = \min$$

but due to quadratic completion it applies:

$$|z|^2 - 2\mathrm{Re}(\bar{z}q) \geq |z|^2 - 2|z| \cdot |q| + |q|^2 - |q|^2 \geq -|q|^2$$

i.e.

$$\hat{u}(n) = \frac{|\hat{a}(n)|^2}{|\hat{a}(n)|^2 + \alpha n^{2p}} \frac{\hat{b}(n)}{\hat{a}(n)}.$$

□

1.6 Numerical Deconvolution, Smoothing of Measured Data

Remark 1.26 – Motivation (Problem Statement)

We have the same assumptions as in the previous section and the problem $a * u + \epsilon = b$ is given, where a, u, and b are 2π-periodic and $\epsilon \in L^2$. We are given the conditions:

$$\|u^{(p)}\|_{L^2} = \min, \quad \|a \cdot u - b\|_{L^2} \leq \delta.$$

Now, b is measured at discrete points $x_j = \frac{2\pi}{N} j$, so we replace b with the trigonometric interpolation polynomial b_N, leading to the following conditions:

$$\|u_N^{(p)}\|_{L^2} = \min, \quad \|a * u_N - b_N\|_{L^2} \leq \delta.$$

Definition 1.7 – Regularization

We choose $\alpha > 0$ and obtain the regularization problem:

$$\|a * u_N - b_N\|_{L^2}^2 + \alpha \|u_N^{(p)}\|_{L^2}^2 = \min$$

among all trigonometric polynomials $u_N(x)$:

$$u_N(x) = \sum_{n=-N/2}^{N/2}{}' \, \hat{u}_N(n) e^{inx}.$$

From the theorem in Section 5, we know that:

$$\hat{u}_N(n) = \Phi_\alpha(n) \frac{\hat{b}_N(n)}{\hat{a}(n)}, \quad n = -\frac{N}{2}, \ldots, \frac{N}{2} - 1.$$

Algorithm 1.2 – Practical Computation

We are given $b(x_j)$ for $j = 0, \ldots, N-1$, as well as $a(x)$ or $\hat{a}(n)$.

1. Compute with the FFT, as in Section 4:

   $$\left(\hat{b}_N(n)\right)_{n=-\frac{N}{2}}^{\frac{N}{2}-1} = \frac{1}{N} \overline{\mathcal{F}_N} (b(x_j))_{j=0}^{N-1}$$

   with a total of $N \log_2 N$ operations.

2. The Fourier coefficients of the apparatus function $\hat{a}(n)$ are either given (often only $\hat{a}(n)$ is provided, not a), or we approximate $\hat{a}_M(n)$ with $M \geq N$, potentially even $M \gg N$.

3. Then calculate:

   $$\hat{u}_N(n) = \frac{\overline{\hat{a}(n)} \hat{b}_N(n)}{|\hat{a}(n)|^2 + \alpha n^{2p}}, \quad n = -\frac{N}{2}, \ldots, \frac{N}{2} - 1.$$

4. Next, compute the discrete Fourier transform using FFT:

   $$(u_N(x_j))_{j=0}^{N-1} = \mathcal{F}_N (\hat{u}_N(n))_{n=-\frac{N}{2}}^{\frac{N}{2}-1}.$$

   with $N \log_2 N$ operations.

Remark 1.27 – Choice of Regularization Parameter

We want to know how to determine or approximate the regularization parameter α. If the estimate (this term is also called the variance):

$$\delta \approx \left(\frac{1}{N} \sum_{j=0}^{N-1} |\varepsilon(x_j)|^2 \right)^{\frac{1}{2}} = \|\varepsilon\|_{L^2}$$

is known, we start with some α and compute (using the Parseval formula):

$$d_\alpha^2 = \|a * u_N - b_N\|_{L^2}^2 = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}}{}' |\hat{a}(n)\hat{u}_N(n) - \hat{b}_N(n)|^2 = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}}{}' (1 - \Phi_\alpha(n))^2 |\hat{b}_N(n)|^2 .$$

We then choose α such that $d_\alpha \approx \delta$. Note that $\alpha \mapsto d_\alpha$ is monotonically increasing, so this process can be iterated relatively easily. At the optimal α, we compute $\hat{u}_N(n)$, and then, using FFT, obtain $u_N(x_j)$.

Remark 1.28

If the variance is unknown, the procedure makes no sense. However, statistical methods can determine an optimal regularization parameter λ. This can be found in the literature under the term "generalized cross-validation."

Remark 1.29 – Smoothing of Data

We have measured values $b(x_j)$ and a variance of the measurement error that is approximately δ. We are looking for a trigonometric polynomial u_N with:

$$\|u_N - b_N\|_{L^2}^2 = \frac{1}{N} \sum_{j=0}^{N-1} |u_N(x_j) - b(x_j)|^2 \le \delta^2 = \min, \quad \|u_N^{(p)}\|_{L^2} = \min .$$

Using our formula, we can compute this directly. For the special case $\hat{a}(n) = 1$, we have $a * u = u$, which corresponds to convolution with the Dirac delta function. For $p = 2$, we obtain:

$$\hat{u}_N(n) = \frac{1}{1 + \alpha n^4} \hat{b}_N(n).$$

We note that the high-frequency components of $\hat{b}_N(n)$ (for large n) are filtered out by this formula, which results in smoothing. By the inverse Fourier transform, we eventually obtain the desired u_N.

Remark 1.30 – Approach for Non-Periodic Data

If the data are not periodic, we subtract a fitting line such that it fluctuates around a level and then extend it periodically.

Remark 1.31

An alternative approach would be to smooth the data using splines instead of Fourier transforms.

Remark 1.32 – Differentiation of noisy data

We want to find the derivative of the data, i.e., $u = b'$. Thus, we have:

$$\int_0^x u(t)\,dt = b(x) - b(0).$$

We have $\hat{u}(n) = in\hat{b}(n)$, so:

$$\underbrace{\frac{1}{in}}_{=:\hat{a}(n)} \hat{u}(n) = \hat{b}(n)$$

This leads to a new minimization problem:

$$\sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \left| \frac{1}{in}\hat{u}_N(n) - \hat{b}_N(n) \right|^2 \le \delta^2, \quad \|u_N''\|_{L^2}^2 = \min.$$

For the special case $\hat{a}(n) = \frac{1}{in}$, we get:

$$\hat{u}_N(n) = \frac{n^{-2}}{n^{-2} + \alpha n^4} in\hat{b}_N(n) = \frac{in}{1 + \alpha n^6}\hat{b}_N(n).$$

Chapter 2

Eigenvalue Problems

In this chapter, n denotes a positive integer.

Review – Left/right eigenvectors

Let $A \in \mathbb{C}^{n\times n}$. If there is $(\lambda, v) \in \mathbb{C} \times \mathbb{C}^n$ such that $Av = \lambda v$, then λ is called an *eigenvalue of* A and v an *eigenvector of* A *associated to* λ. If there is $(\kappa, u) \in \mathbb{C} \times \mathbb{C}^n$ such that $u^*A = \kappa u^*$, then κ is also called an eigenvalue of A and u an *eigenvector of* A *associated to* κ. When it is not specified, an eigenvector usually refers to a right eigenvector.

Question
Why don't we need to distinguish left and right eigenvalues?

Answer
Due to the fact that the determinant of a matrix B is equal to the determinant of its transpose, by taking $B = A - \lambda I$ we get $\chi_A(\lambda) = \det(A - \lambda I) = \det(A^T - \lambda I) = \chi_{A^T}(\lambda)$, where χ_A is the characteristic polynomial of A. Notice that $u^*A = \kappa u^* \iff A^T\bar{u} = \kappa\bar{u}$, hence κ is an eigenvalue of A^T. In other words, κ is also an eigenvalue of A. Hence, all left eigenvalues are right eigenvalues, and there is no need to distinguish them.

Note
The eigenvectors are conventionally taken to be of unit ℓ^2 norm, since for any $c \in \mathbb{C}$,

$$Av = \lambda v \iff A(cv) = \lambda(cv).$$

Taking $c = 1/\|v\|_2$, the eigenvector $u = cv$ is of unit ℓ^2 norm.

In this chapter, we will generally assume that eigenvectors are normalized.

Review – Diagonalization

A matrix $A \in \mathbb{C}^{n\times n}$ is said to be *diagonalizable* if there is an invertible matrix $P \in \mathbb{C}^{n\times n}$ and a diagonal matrix $\Lambda \in \mathbb{C}^{n\times n}$ such that

$$P^{-1}AP = \Lambda.$$

2.1 Fundamentals

Motivation
There are numerous applications where eigenvalues are required:

1. *In mechanics (physics), for example, one is interested in the natural vibrations of membranes. If $u(x)$ represents the deflection on a domain Ω, then we require that*

$$-\Delta u = \lambda u \quad in \quad \Omega, \qquad u = 0 \quad on \quad \partial\Omega.$$

For a grid over Ω, we obtain the discretization

$$Av = \lambda v.$$

2. *In biology, the Lotka-Volterra predatorprey model is frequently used to describe the dynamics of two interacting species:*

$$\begin{cases} \frac{dx}{dt} = \alpha x - \beta xy \\ \frac{dy}{dt} = -\gamma y + \delta xy, \end{cases}$$

where x (resp. y) is the population density of prey (resp. predator). The system's equilibrium is obtained when $\frac{dx}{dt} = \frac{dy}{dt} = 0$, which yields two points:

$$\{x = 0,\, y = 0\} \quad \text{or} \quad \left\{x = \frac{\gamma}{\delta},\, y = \frac{\alpha}{\beta}\right\}.$$

The stability of fixed points is studied by looking at eigenvalues of a certain matrix.

- $\{x = 0,\, y = 0\}$*: there are always one positive and one negative eigenvalues, hence the equilibrium is unstable*
- $\left\{x = \frac{\gamma}{\delta},\, y = \frac{\alpha}{\beta}\right\}$*: there are always two complex conjuguate eigenvalues, hence the population oscillates with time around that point, and the system is stable in a certain sense.*

3. *Search engines: "The 25,000,000,000 Dollar Eigenvalue Problem", which we will discuss later in the context of Google's mechanism.*

2.1.1 Review (Characteristic Polynomial)

Review – Characteristic polynomial

For $A \in \mathbb{C}^{n\times n}$, the *characteristic polynomial* χ_A of A is defined by $\chi_A(\lambda) := \det(A - \lambda I)$.

The condition $Av = \lambda v$ is equivalent to $\det(A - \lambda I) = \chi_A(\lambda) = 0$. One might consider first computing the characteristic polynomial and then finding its roots to obtain the eigenvalues.

Note
The equivalence is shown as follows:

$$(A - \lambda I)v = 0 \text{ for some } v \neq 0 \iff (A - \lambda I) \text{ is singular} \iff \det(A - \lambda I) = 0.$$

Example 2.1 – Poor Conditioning

Let $A = \text{diag}(10, 11, \ldots, 16)$ be a 7×7 matrix. Here, it is clear what the eigenvalues are, and the characteristic polynomial is:

$$\chi_A(\lambda) = (10 - \lambda)(11 - \lambda)\cdots(16 - \lambda) \quad = -\lambda^7 + 91\lambda^6 - 3535\lambda^5 + \ldots - 31813200\lambda + 57657600.$$

If one computes the roots of χ_A in single precision (i.e. $\texttt{eps} = 10^{-8}$), the result[a] is:

$$9.97,\ 11.31 - 0.30i,\ 11.37 + 0.33i,\ 13.47 - 0.76i,\ 13.57 + 0.76i,\ 15.51 - 0.09i,\ 15.80 + 0.06i.$$

This does not correspond to the actual eigenvalues. The problem is that computing the roots of a polynomial from its coefficients is a poorly conditioned problem.

[a] Results obtained via the Julia programming language with Float32 precision (giving between 6 and 9 digits of precision)

Remark 2.1 – How to explain poor conditioning of the root-finding problem?

Let

$$p(\lambda) = \sum_{k=0}^{n} a_k \lambda^k,$$

and we suppose its roots are simple (i.e. they are all distinct). The coefficients $\{a_k\}$ are the "real" ones, but numerically they are only known up to a certain precision η: the computer only sees coefficients $\{a_k(1+\varepsilon_k)\}$ for some $|\varepsilon_k| \le \eta$, and the polynomial seen by the computer is

$$p(\lambda,\eta) = \sum_{k=0}^{n} (a_k + a_k\varepsilon_k)\lambda^k = \sum_{k=0}^{n} \left(a_k + \underbrace{a_k \frac{\varepsilon_k}{\eta}}_{=:b_k} \eta \right) \lambda^k = p(\lambda) + q(\lambda)\eta$$

with $q(\lambda) = \sum_{k=0}^{n} b_k \lambda^k$ and $|b_k| \le |a_k|$. We study the roots $\lambda(\eta)$ of $p(\lambda,\eta) = p(\lambda) + \eta q(\lambda)$ as a function of η. Let $\lambda(0) = \lambda^*$ be a simple root of p. We consider the differentiable function $\eta \mapsto \lambda(\eta)$ defined by $p(\lambda(\eta),\eta) = 0$ for all η with $|\eta| \le \eta_0$. It exists and is unique by the implicit function theorem, hence:

$$\underbrace{\frac{\partial p}{\partial \lambda}(\lambda(\eta),\eta)}_{p'(\lambda(\eta))\lambda'(\eta)+\mathcal{O}(\eta)} + \underbrace{\frac{\partial p}{\partial \eta}(\lambda(\eta),\eta)}_{q(\lambda(\eta))} = 0.$$

Thus,

$$\lambda'(\eta) \approx -\frac{q(\lambda(\eta))}{p'(\lambda(\eta))}, \quad \lambda(\eta) \approx \lambda^* + \eta\lambda'(0),$$

and the relative error is:

$$\frac{|\lambda(\eta) - \lambda^*|}{|\lambda^*|} \approx \frac{|\eta\lambda'(0)|}{|\lambda^*|} \approx |\eta| \cdot \left| \frac{q(\lambda^*)}{\lambda^* p'(\lambda^*)} \right|.$$

The term $\left| \frac{q(\lambda^*)}{\lambda^* p'(\lambda^*)} \right|$ can become very large. Coming back to Example 2.1, we have

$$q(\lambda) = \frac{1}{\eta}\left(-\varepsilon_0 \lambda^7 + 91\varepsilon_1\lambda^6 + \cdots - 31813200\varepsilon_6\lambda + 57657600\varepsilon_7 \right),$$

and thus for $\lambda^* = 10$ we obtain

$$\begin{aligned}
|q(\lambda^*)| &\le \frac{1}{\eta}\left(|\varepsilon_0(\lambda^*)^7| + |91\varepsilon_1(\lambda^*)^6| + \cdots + |31813200\varepsilon_6\lambda^*| + |57657600\varepsilon_7| \right) \\
&\le |(\lambda^*)^7| + |91(\lambda^*)^6| + \cdots + |31813200\lambda^*| + |57657600| \\
&\le 10^7 + 91 \cdot 10^7 + \cdots + 31.8132 \cdot 10^7 + 5,76576 \cdot 10^7 \\
&\approx 10^8,
\end{aligned}$$

and $|p'(\lambda^*)| = 720 \approx 10^3$. So the relative error is $\eta \cdot 10^4$, which means that we lose about four significant digits in the decimal expansion of the roots. Therefore, it is numerically unreasonable to compute the coefficients of the characteristic polynomial with a coarse precision.

Proposition 2.1 – Obtaining eigenvectors by similarity transformations

Let $A \in \mathbb{C}^{n\times n}$, and $T \in \mathbb{C}^{n\times n}$ invertible. If $B = T^{-1}AT$, then

$$Av = \lambda v \iff TBT^{-1}v = \lambda v \iff B(T^{-1}v) = \lambda(T^{-1}v).$$

Thus, B and A have the same eigenvalues, and v is an eigenvector of A if and only if $T^{-1}v$ is an eigenvector of B.

Definition 2.1 – Unitary/orthogonal matrix

The matrix $U \in \mathbb{C}^{n\times n}$ is *unitary* if $U^*U = UU^* = I$, where $U^* = \overline{U^T}$. In other words, $U^{-1} = U^*$. If U is real, then $U^* = U^T$ and U is said to be *orthogonal*.

Exercise
Let $U_1, \dots, U_k \in \mathbb{C}^{n\times n}$ unitary matrices. Show that $V = U_1 \cdots U_k$ is a unitary matrix.

Answer
We have $V^* = U_k^* \cdots U_1^*$ hence, owing to the unitary character of each matrix U_ℓ,

$$V^*V = U_k^* \cdots U_1^* U_1 \cdots U_k = U_k^* \cdots U_2^* U_2 \cdots U_k = \cdots = I,$$

and

$$VV^* = U_1 \cdots U_k U_k^* \cdots U_1^* = U_1 \cdots U_{k-1} U_{k-1}^* \cdots U_1^* = \cdots = I.$$

Theorem 2.1 – Schur's normal form (1909)

Let $A \in \mathbb{C}^{n\times n}$, there is a unitary matrix U such that

$$U^*AU = \begin{pmatrix} \lambda_1 & \star & \dots & \star \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \star \\ 0 & \dots & 0 & \lambda_n \end{pmatrix}$$

is an upper triangular matrix. This form is called Schur's normal form.

Proof. The characteristic polynomial $\chi_A(\lambda)$ has a root $\lambda_1 \in \mathbb{C}$, which is an eigenvalue of A. Thus, there exists an eigenvector $0 \neq v_1 \in \mathbb{C}^n$ such that $Av_1 = \lambda_1 v_1$. We can assume without loss of generality that $\|v_1\|_2 = 1$. We now construct (using Gram-Schmidt) a matrix $V_1 = (v_1, v_2, \dots, v_n)$ with $v_2, \dots, v_n$ chosen so that $v_1, v_2, \dots, v_n$ form an orthonormal basis (ONB) of $\mathbb{C}^n$. In other words, V_1 is a unitary matrix.

We now consider

$$AV_1 = (Av_1, Av_2, \dots, Av_n) = V_1 \begin{pmatrix} \lambda_1 & \star \\ 0 & \hat{A} \end{pmatrix}$$

We proceed in the same way with the matrix $\hat{A} \in \mathbb{C}^{(n-1)\times(n-1)}$: there exists an eigenvalue $\lambda_2 \in \mathbb{C}$ of $\hat{A}$ and an associated unit eigenvector $\hat{v}_2$, so one can construct an ONB of $\mathbb{C}^{n-1}$ by the Gram-Schmidt procedure. This yields a unitary matrix $V_2 \in \mathbb{C}^{(n-1)\times(n-1)}$ such that

$$\hat{A}V_2 = V_2 \begin{pmatrix} \lambda_2 & \star \\ 0 & \hat{\hat{A}} \end{pmatrix} \iff \hat{A} = V_2 \begin{pmatrix} \lambda_2 & \star \\ 0 & \hat{\hat{A}} \end{pmatrix} V_2^*.$$

Hence,

$$AV_1 = V_1 \begin{pmatrix} \lambda_1 & \star \\ 0 & V_2 \begin{pmatrix} \lambda_2 & \star \\ 0 & \hat{\hat{A}} \end{pmatrix} V_2^* \end{pmatrix} = V_1 \begin{pmatrix} 1 & 0 \\ 0 & V_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & \star & \star \\ 0 & \lambda_2 & \star \\ 0 & 0 & \hat{\hat{A}} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & V_2^* \end{pmatrix}.$$

Finally, since $\begin{pmatrix} 1 & 0 \\ 0 & V_2^* \end{pmatrix}$ is a unitary matrix, we get

$$AV_1 \begin{pmatrix} 1 & 0 \\ 0 & V_2 \end{pmatrix} = V_1 \begin{pmatrix} 1 & 0 \\ 0 & V_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & \star & \star \\ 0 & \lambda_2 & \star \\ 0 & 0 & \hat{\hat{A}} \end{pmatrix}.$$

By repeating this process, we get a matrix $U \in \mathbb{C}^{(n-1)\times(n-1)}$ of the form

$$U = V_1 \begin{pmatrix} 1 & 0 \\ 0 & V_2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & V_3 \end{pmatrix} \cdots \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & \dots & 0 & V_n \end{pmatrix},$$

where each matrix $V_k \in \mathbb{C}^{(n-k+1)\times(n-k+1)}$ is unitary. The matrix U is unitary as a product of unitary matrices, and it satisfies

$$AU = U \begin{pmatrix} \lambda_1 & \star & \dots & \star \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \star \\ 0 & \dots & 0 & \lambda_n \end{pmatrix}$$

The claimed result is obtained after left-multiplying by U^*. □

Review 2.1 – Hermitian and symmetric matrices

A matrix $A \in \mathbb{C}^{n\times n}$ is said to be *Hermitian* if $A^* = A$ and *skew-Hermitian* if $A^* = -A$, where $A^* = \bar{A}^T$. A matrix $A \in \mathbb{R}^{n\times n}$ is said to be *symmetric* if $A^T = A$, and *skew-symmetric* if $A^T = -A$.

Note

In some contexts, an Hermitian/symmetric matrix can also be called self-adjoint *when seen as a linear operator.*

Definition 2.2 – Normal matrix

A matrix A is *normal* if $AA^* = A^*A$ (when $A \in \mathbb{C}^{n\times n}$) or if $AA^T = A^TA$ (when $A \in \mathbb{R}^{n\times n}$).

Lemma 2.1

A normal upper triangular matrix is diagonal.

Proof. Let $R \in \mathbb{C}^{n\times n}$ a normal upper triangular matrix

$$R = \begin{pmatrix} r_{11} & \dots & r_{1,n} \\ 0 & & \\ \vdots & R_1 & \\ 0 & & \end{pmatrix},$$

with $R_1 \in \mathbb{C}^{(n-1)\times(n-1)}$ an upper triangular matrix. Since R is normal, $R^*R = RR^*$ and we get

$$\begin{pmatrix} \overline{r_{1,1}} & 0 \quad \dots \quad 0 \\ \vdots & R_1^* \\ \overline{r_{1,n}} & \end{pmatrix} \begin{pmatrix} r_{1,1} & \dots \quad r_{1,n} \\ 0 & \\ \vdots & R_1 \\ 0 & \end{pmatrix} = \begin{pmatrix} r_{1,1} & \dots \quad r_{1,n} \\ 0 & \\ \vdots & R_1 \\ 0 & \end{pmatrix} \begin{pmatrix} \overline{r_{1,1}} & 0 \quad \dots \quad 0 \\ \vdots & R_1^* \\ \overline{r_{1,n}} & \end{pmatrix},$$

i.e.

$$\begin{pmatrix} |r_{1,1}|^2 & \overline{r_{1,1}}r_{1,2} \quad \dots \quad \overline{r_{1,1}}r_{1,n} \\ \overline{r_{1,2}}r_{1,1} & \\ \vdots & (\overline{r_{1,i+1}}r_{1,j+1})_{i,j=1}^{n-1} + R_1^*R_1 \\ \overline{r_{1,n}}r_{1,1} & \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^n |r_{1,j}|^2 & \begin{pmatrix} r_{1,2} & \dots & r_{1,n} \end{pmatrix} R_1^* \\ R_1 \begin{pmatrix} \overline{r_{1,2}} \\ \vdots \\ \overline{r_{1,n}} \end{pmatrix} & R_1R_1^* \end{pmatrix}$$

The component at index (1, 1) yields $|r_{1,1}|^2 = \sum_{j=1}^{n} |r_{1,j}|^2$, i.e. $r_{1,j} = 0$ for $j = 2, \dots, n$. Hence,

$$\begin{pmatrix} |r_{1,1}|^2 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & R_1^* R_1 & \\ 0 & & & \end{pmatrix} = \begin{pmatrix} |r_{1,1}|^2 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & R_1 R_1^* & \\ 0 & & & \end{pmatrix},$$

i.e. the matrix R_1 is normal. In other words, the first row of a normal upper triangular matrix has all its off-diagonal terms equal to zero. Since R_1 is also a normal upper triangular matrix, by induction we obtain that R is a diagonal matrix. □

Theorem 2.2 – Spectral theorem

A matrix $A \in \mathbb{C}^{n \times n}$ is normal if and only if there is a unitary $U \in \mathbb{C}^{n \times n}$ such that

$$U^* A U = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}.$$

Proof. [⇒] The Schur normal form of A yields a unitary matrix U and an upper triangular matrix R such that $U^* A U =: R$. Since A is normal,

$$R^* R = U^* A^* U U^* A U = U^* A^* A U = U^* A A^* U = U^* A U U^* A^* U = R R^*.$$

The matrix R is normal and upper triangular, hence diagonal.
[⇐] One has $A = U \operatorname{diag}(\lambda_1, \dots, \lambda_n) U^*$, thus

$$A^* A = U \operatorname{diag}(\overline{\lambda_1}, \dots, \overline{\lambda_n}) U^* U \operatorname{diag}(\lambda_1, \dots, \lambda_n) U^* = U \operatorname{diag}(|\lambda_1|^2, \dots, |\lambda_n|^2) U^*,$$

and

$$A A^* = U \operatorname{diag}(\lambda_1, \dots, \lambda_n) U^* U \operatorname{diag}(\overline{\lambda_1}, \dots, \overline{\lambda_n}) U^* = U \operatorname{diag}(|\lambda_1|^2, \dots, |\lambda_n|^2) U^*.$$

The matrix A is then normal. □

Question
Show that the λ_j in the spectral theorem are the eigenvalues of A.

Answer
Let κ an eigenvalue of A, it is a root of the characteristic polynomial χ_A: $\chi_A(\kappa) = 0$. On the one hand

$$\det(U^* A U - \kappa I) = \det(U^* A U - \kappa U^* U) = \det(U^*) \chi_A(\kappa) \det(U) = \chi_A(\kappa),$$

where the last equality is due to U being unitary hence $\det(U^*) \det(U) = \det(U^{-1}) \det(U) = 1$. On the other hand,

$$\det(U^* A U - \kappa I) = \det(\operatorname{diag}\{\lambda_1 - \kappa, \dots, \lambda_n - \kappa\}).$$

Thus, κ is a root of χ_A if and only if $\det(\operatorname{diag}\{\lambda_1 - \kappa, \dots, \lambda_n - \kappa\}) = 0$. The determinant of a diagonal matrix is equal to the product of its diagonal elements, hence it is zero if and only if at least one element is zero, i.e. if there is a j such that $\kappa = \lambda_j$. This shows $\operatorname{sp}(A) \subset \{\lambda_1, \dots, \lambda_n\}$. It is clear that $\{\lambda_1, \dots, \lambda_n\} \subset \operatorname{sp}(A)$, since $\chi_A(\lambda_j) = \det(U^* A U - \lambda_j I) = 0$.

Theorem 2.3 – Jordan normal form

For every $A \in \mathbb{C}^{n\times n}$, there is an invertible matrix T such that

$$T^{-1}AT = J = \begin{pmatrix} J_{n_1}(\lambda_1) & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & J_{n_k}(\lambda_k) \end{pmatrix},$$

with λ_k the eigenvalues of A and $n_k \in \mathbb{N}^*$. The Jordan blocks are given by

$$J_m(\lambda) = \begin{pmatrix} \lambda & 0 & \dots & \dots & 0 \\ 1 & \lambda & \ddots & & \vdots \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & \lambda \end{pmatrix} \in \mathbb{C}^{m\times m}.$$

The number of Jordan blocks associated to a given λ is given by $\dim\ker(A - \lambda I)$, and the sum of the sizes of all Jordan blocks associated to λ is given by its algebraic multiplicity (i.e. the multiplicity of λ as a root of χ_A).

Proof. Linear Algebra II. □

Remark 2.2

On the sub-diagonal of a Jordan block, there can be entries $\varepsilon \neq 0$ instead of ones. Sometimes, the ones of the sub-diagonal are instead on the sup-diagonal.

2.2 Conditioning of the Eigenvalue Problem

Motivation

Let $A = (a_{ij})$ be a given matrix. Due to rounding errors, we can only work with a perturbed matrix $\tilde{A} = (\tilde{a}_{ij})$ such that $\tilde{a}_{ij} = a_{ij}(1 + \varepsilon_{ij})$ with $|\varepsilon_{ij}| \leq \eta$, where $\eta > 0$ is the numerical precision. Thus, we can write:

$$\tilde{A} = A + \eta \cdot C, \quad c_{ij} = a_{ij}\frac{\varepsilon_{ij}}{\eta}, \ |c_{ij}| \leq |a_{ij}|.$$

We consider $A(\eta) = A + \eta C$ for small η. We want to find an estimate for the eigenvalues like:

$$|\lambda(\eta) - \lambda(0)| \leq \text{const.} \cdot \eta.$$

Is this even possible, and if so, with which constant? We will see with the following theorem that this depends on a constant that depends on A.

Lemma 2.2

Let $A \in \mathbb{C}^{n\times n}$ and $\lambda \in \mathbb{C}$ a simple root of χ_A. Let $u, v \in \mathbb{C}^n$ be respectively left and right eigenvectors of A associated to λ. Then

$$u^* v \neq 0.$$

Proof. (Done in Exercise). Since λ is a simple root of χ_A, its algebraic multiplicity is one. With respect to the Jordan form, it means that the sum of the sizes of all Jordan blocks associated to λ is one. Hence, the Jordan form of A is

$$T^{-1}AT = \begin{pmatrix} \lambda & 0 \\ 0 & J' \end{pmatrix},$$

where $J' \in \mathbb{C}^{(n-1)\times(n-1)}$ is a matrix in Jordan form and $T \in \mathbb{C}^{n\times n}$ is an invertible matrix. Since

$$AT = T\begin{pmatrix}\lambda & 0\\ 0 & J'\end{pmatrix} \quad \text{and} \quad T^{-1}A = \begin{pmatrix}\lambda & 0\\ 0 & J'\end{pmatrix}T^{-1},$$

we deduce that the first column of T is a multiple of v, and that the first row of T^{-1} is a multiple of u^*. Thus, there are nonzero complex coefficients α, β such that

$$v = \alpha T_{\cdot,1} \quad \text{and} \quad u = \beta (T^{-*})_{\cdot,1}.$$

We can write $T = (v/\alpha, T_1)$ and $T^{-*} = (u/\beta, Z_1)$ for some matrices $T_1, Z_1 \in \mathbb{C}^{n\times(n-1)}$. By definition,

$$I = T^{-1}T = (T^{-*})^*T = \begin{pmatrix}u^*/\bar\beta\\ Z_1^*\end{pmatrix}\begin{pmatrix}v/\alpha & T_1\end{pmatrix} = \begin{pmatrix}u^*v/(\alpha\bar\beta) & u^*T_1/\bar\beta\\ Z_1^*v/\alpha & Z_1^*T_1\end{pmatrix}.$$

Looking at the component at index (1, 1), we get $u^*v = \alpha\bar\beta \neq 0$. □

Definition 2.3 – Condition number of an eigenvalue

Let $A \in \mathbb{C}^{n\times n}$ and $\lambda \in \mathbb{C}$ an eigenvalue. Let u (resp. v) denote a left (resp. right) eigenvector associated to λ. The quantity $1/|u^*v|$ is called the *condition number of the eigenvalue* λ.

A "well-conditioned" matrix means that small perturbations only result in small changes. For a given matrix $A \in \mathbb{C}^{n\times n}$, if the condition number of an eigenvalue λ is large it means that small perturbations of A may result in large perturbations of λ. The condition number of an eigenvalue can be understood as a measure of the "continuity" of this eigenvalue depending on the size of the perturbation.

Theorem 2.4 – Error estimate for a simple eigenvalue

Let $A \in \mathbb{C}^{n\times n}$ and λ a simple root of the characteristic polynomial χ_A. For small ε, an eigenvalue of $A(\varepsilon) = A + \varepsilon C$ satisfies the following relation:

$$\lambda(\varepsilon) = \lambda + \varepsilon\frac{u^*Cv}{u^*v} + \mathcal{O}(\varepsilon^2)$$

where v is an eigenvector of A associated to λ and u is a left eigenvector of A associated to λ.

Proof. Using Lemma 2.2, we know that $u^*v \neq 0$. We start by showing that there is a differentiable eigenvalue $\lambda(\varepsilon)$ and an associated normalized eigenvector $v(\varepsilon)$ depending on ε such that $\lambda(0) = \lambda$ and $v(0) = v$. This is done using the implicit function theorem.

For that, we consider the function

$$F(\varepsilon, \kappa, w) := \begin{pmatrix}(A(\varepsilon) - \kappa I)w\\ w^*w - 1\end{pmatrix}.$$

It is a C^1 function of the three variables, and $F(0, \lambda, v) = 0$ by definition of λ and v. We compute the Jacobian of F with respect to (κ, w), evaluated at the point $(\varepsilon = 0, \kappa = \lambda, w = v)$:

$$\left(D_{(\kappa,w)}F\right)(0, \lambda, v) = (D_\kappa F, D_{w_1}F, \ldots, D_{w_n}F)(0, \lambda, v) = \begin{pmatrix}-v & A - \lambda I\\ 0 & v^*\end{pmatrix}.$$

For the derivatives with respect to a complex variable $z = x + iy \in \mathbb{C}$, we have used the Wirtinger derivative: $\frac{\partial}{\partial z} = \frac{1}{2}\left(\frac{\partial}{\partial x} - \frac{\partial}{\partial y}\right)$. We now show that this matrix is nonsingular, i.e. that its kernel is the zero element. Assume the matrix is singular, then one can find $(x, y) \in (\mathbb{C}\times\mathbb{C}^n)\setminus\{0\}$ such that

$$\begin{pmatrix}-v & A - \lambda I\\ 0 & v^*\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} = 0 \quad \iff \quad \begin{cases} -vx + (A - \lambda I)y = 0\\ v^*y = 0\end{cases}$$

Left-multiply the first equation by u^* to obtain

$$-xu^*v + u^*(A - \lambda I)y = 0.$$

However, since $u^*(A - \lambda I) = 0$ and $u^*v \neq 0$, we get $x = 0$. The equation $(A - \lambda I)y = 0$ means $y = \ker(A - \lambda I)$. But since λ is an eigenvalue of algebraic multiplicity one, it is also of geometric multiplicity one which means that $\dim\ker(A - \lambda I) = 1$. Since the kernel is a $\mathbb{C}$-linear space and we know $v \in \ker(A - \lambda I)$, we obtain $\ker(A - \lambda I) = \operatorname{span}\{v\} = \{\alpha v : \alpha \in \mathbb{C}\}$. Therefore there is $c \in \mathbb{C}$ such that $y = cv$. The equation $v^*y = 0$ yields $c = 0$ since $v^*v = 1$. Hence, there is no $(x, y) \in (\mathbb{C} \times \mathbb{C}^n) \setminus \{0\}$ such that

$$\begin{pmatrix} -v & A - \lambda I \\ 0 & v^* \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 0.$$

In other words, the matrix $\left(D_{(\kappa,w)}F\right)(0, \lambda, v)$ is invertible, i.e. its determinant is nonzero. The implicit function theorem now applies, and we get the existence of a unique differentiable function $\varphi(\varepsilon) = (\lambda(\varepsilon), v(\varepsilon)) \in \mathbb{C} \times \mathbb{C}^n$ such that $F(\varepsilon, \lambda(\varepsilon), v(\varepsilon)) = 0$ for $|\varepsilon|$ small enough, i.e.

$$(A(\varepsilon) - \lambda(\varepsilon)I)\, v(\varepsilon) = 0,\ \|v(\varepsilon)\|_2 = 1,\ v(0) = v,\ \lambda(0) = \lambda.$$

We have $A(\varepsilon)v(\varepsilon) = \lambda(\varepsilon)v(\varepsilon)$, i.e.

$$(A + \varepsilon C)(v + \varepsilon v'(0) + \mathcal{O}(\varepsilon^2)) = (\lambda + \varepsilon\lambda'(0) + \mathcal{O}(\varepsilon^2))(v + \varepsilon v'(0) + \mathcal{O}(\varepsilon^2)).$$

We multiply this out and compare the powers of ε:

$$\varepsilon^0 : Av = \lambda v, \quad \varepsilon^1 : Cv + Av'(0) = \lambda v'(0) + \lambda'(0)v.$$

We obtain: $(A - \lambda I)v'(0) = -Cv + \lambda'(0)v$. By left multiplying with u^* and using $u^*(A - \lambda I) = 0$, we obtain:

$$0 = -u^*Cv + \lambda' u^*v \Rightarrow \lambda' = \frac{u^*Cv}{u^*v}.$$

□

Remark 2.3

The error estimate for the (right) eigenvectors can be shown to be

$$v_j(\varepsilon) = v_j + \varepsilon \sum_{i=1, i\neq j}^{n} \frac{1}{\lambda_j - \lambda_i} \frac{u_i^* C v_j}{u_i^* v_i} v_i + \mathcal{O}(\varepsilon^2).$$

This is done in an Exercise.

Example 2.2

1. If A is normal, then the left and right eigenvectors are identical, because there is a U unitary (by the spectral theorem) such that:

$$U^*AU = \operatorname{diag}(\lambda_1, \ldots, \lambda_n).$$

After multiplying by U^* on the right, we get $U^*A = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)U^*$, i.e. the left eigenvectors are in the matrix U. If we multiply instead by U on the left, we get $AU = U \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, i.e. the right eigenvectors are in the matrix U. Hence, left and right eigenvectors agree. When the eigenvalues are all simple roots of the characteristic polynomial, we get

$$\frac{1}{|u^*v|} = 1$$

for all eigenvalues, if u and v are normalized. In this case, the problem is well-conditioned.

We could even show here that not only

$$|\lambda(\varepsilon) - \lambda| \leq |\varepsilon| \cdot |v^* C v|$$

holds, but also

$$|\lambda(\varepsilon) - \lambda| \leq |\varepsilon| \|C\|_2.$$

2. If A is not normal, $u^* v$ can be arbitrarily small, e.g.,

$$A = \begin{pmatrix} 1 & \alpha \\ 0 & 2 \end{pmatrix}, \quad \lambda = 1, \quad v = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad u^* = \frac{1}{\sqrt{1+\alpha^2}} \begin{pmatrix} 1 & -\alpha \end{pmatrix}.$$

For a very large α, u is almost the second unit vector, and then:

$$u^* v = \frac{1}{\sqrt{1+\alpha^2}} \to 0.$$

Here, the problem would be very poorly conditioned.

Remark 2.4 – Consideration for multiple eigenvalues

(Details given in the tutorials). We cannot transfer the proof because the conditions for the implicit function theorem are not guaranteed. We consider here as an example the matrix with multiple Jordan blocks:

$$A = \begin{pmatrix} \lambda & 1 & 0 & \dots & 0 \\ 0 & \lambda & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & \ddots & \lambda & 1 \\ 0 & \dots & \dots & 0 & \lambda \end{pmatrix}$$

which represents a Jordan block of size $n \times n$. We consider the characteristic polynomial of $A + \varepsilon C$, where

$$C = \begin{pmatrix} 0 & & & \\ \vdots & & 0 & \\ 0 & & & \\ c & 0 & \dots & 0 \end{pmatrix}.$$

We have

$$\chi_{A+\varepsilon C}(\lambda)(x) = (\lambda - x)^n + \varepsilon(-1)^{n+1} c.$$

If $\lambda(\varepsilon)$ is an eigenvalue of $A + \varepsilon C$, then:

$$(\lambda - \lambda(\varepsilon))^n = \varepsilon(-1)^n c \implies \lambda(\varepsilon) = \lambda + \varepsilon^{\frac{1}{n}} c^{\frac{1}{n}},$$

where the n-th root is considered as a function $\mathbb{C} \to \mathbb{C}$. The problem is therefore not well-conditioned, since the $O(\varepsilon)$ estimate does not hold. Here, it is difficult to compute eigenvalues since they are all close to each other.

Algorithm 2.1 – QR Algorithm, simple version

We want to compute all eigenvalues of a matrix A. For this, we consider the simple iteration:

1. $A_0 := A$
2. We then compute for all $k = 0, 1, 2, \dots$: $A_k = Q_k R_k$ with the QR decomposition and set $A_{k+1} := R_k Q_k$. Essentially, we then have that $A_k \to R$ converges, where R is a right upper

triangular matrix in Schur's normal form, since:

$$A = Q^* R Q \quad \text{with} \quad Q = Q_0 Q_1 Q_2 \dots$$

Remark 2.5

We will see in Page 43 that we can prove convergence.

2.3 Power Method

Algorithm 2.2 – Power method

To compute individual eigenvalues and eigenvectors of a matrix $A \in \mathbb{C}^{n\times n}$, we consider the following procedure: for $y_0 \in \mathbb{C}^n$ arbitrary, set $y_{k+1} := Ay_k$ for $k = 1, 2, \dots$, i.e. $y_k = A^k y_0$.

If one has an approximation to a desired eigenvalue, the inverse power method can be more efficient.

Algorithm 2.3 – Inverse power method (Wielandt iteration)

Let an approximation μ to a desired eigenvalue λ_1 be known, which does not necessarily have to be the largest eigenvalue. Assume

$$|\mu - \lambda_1| \ll |\mu - \lambda_j| \quad \forall j = 2, \dots, n,$$

we have also:

$$\frac{1}{|\mu - \lambda_1|} \gg \frac{1}{|\mu - \lambda_j|}.$$

Since $\frac{1}{\mu - \lambda_j}$ are the eigenvalues of the matrix $(\mu I - A)^{-1}$, we apply the power method to $(\mu I - A)^{-1}$. This can be done without computing the inverse matrix, by only solving the associateed linear system: let y_0 be a starting vector, we solve in the k-th step:

$$(\mu I - A) y_{k+1} = y_k \quad k = 0, 1, 2, \dots$$

We need only one LR decomposition for all iteration steps (since the matrix is the same for each step).

Remark 2.6 – Convergence speed

The convergence speed of the inverse power method may be much better than that of the normal power method. We can, for example, get a rough estimate of the eigenvalue and eigenvector with the normal power method, and then use the inverse power method to obtain precise estimates.

Definition 2.4 – Rayleigh quotient

Let $A \in \mathbb{C}^{n\times n}$ and y_k as in the power method algorithm. The *Rayleigh quotient* is defined by:

$$\frac{y_k^* A y_k}{y_k^* y_k}.$$

Theorem 2.5 – Convergence of the power method

Let $A \in \mathbb{C}^{n\times n}$ be diagonalizable, with $T^{-1}AT = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$ and $T = (v_1 | \dots | v_n)$, where $Av_i = \lambda v_i$. Assume that $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$. If $y_0 = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n$ with $\alpha_1 \neq 0$, then for $y_{k+1} = Ay_k$:

1. We have:
$$y_k = \lambda_1^k \left(\alpha_1 v_1 + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)\right).$$
We note that $\frac{1}{\lambda_1^k} y_k$ converges to an eigenvector $\alpha_1 v_1$ corresponding to the largest eigenvalue.
2. For the Rayleigh quotient, we have
$$\frac{y_k^* A y_k}{y_k^* y_k} = \lambda_1 + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right).$$
If A is normal, then:
$$\frac{y_k^* A y_k}{y_k^* y_k} = \lambda_1 + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right).$$

Proof.

1. Let $y_0 = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$ with $\alpha_1 \neq 0$. We assume without loss of generality that $\|v_j\|_2 = 1$ for all j. Then
$$y_1 = Ay_0 = \alpha_1\lambda_1 v_1 + \alpha_2\lambda_2 v_2 + \ldots + \alpha_n\lambda_n v_n,$$
and iteratively we obtain:
$$\begin{aligned} y_k = A^k y_0 &= \alpha_1\lambda_1^k v_1 + \alpha_2\lambda_2^k v_2 + \ldots + \alpha_n\lambda_n^k v_n \\ &= \lambda_1^k \left(\alpha_1 v_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \ldots + \alpha_n \left(\frac{\lambda_n}{\lambda_1}\right)^k v_n\right). \end{aligned}$$
2. We have:
$$\begin{aligned} y_k^* y_k &= \sum_{i=1}^n \sum_{j=1}^n \overline{\alpha_i}\overline{\lambda_i}^k \alpha_j \lambda_j^k v_i^* v_j = \sum_{i=1}^n |\alpha_i|^2 |\lambda_i|^{2k} \underbrace{v_i^* v_i}_{=1} + \sum_{i=1}^n \sum_{j\neq i}^n \overline{\alpha_i}\overline{\lambda_i}^k \alpha_j \lambda_j^k v_i^* v_j \\ &= |\lambda_1|^{2k} \left(|\alpha_1|^2 + \sum_{i=1}^n |\alpha_i|^2 \left|\frac{\lambda_i}{\lambda_1}\right|^{2k}\right) + |\lambda_1|^{2k} \sum_{i=1}^n \sum_{j\neq i}^n \overline{\alpha_i}\alpha_j \frac{\overline{\lambda_i}^k \lambda_j^k}{|\lambda_1|^{2k}} v_i^* v_j \\ &= |\alpha_1|^2 |\lambda_1|^{2k} \left(1 + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)\right). \end{aligned}$$
Then
$$\begin{aligned} y_k^* A y_k = y_k^* y_{k+1} &= \sum_{i=1}^n |\alpha_i|^2 |\lambda_i|^{2k} \lambda_i v_i^* v_i + \sum_{i=1}^n \sum_{j\neq i}^n \overline{\alpha_i}\overline{\lambda_i}^k \alpha_j \lambda_j^{k+1} v_i^* v_j \\ &= |\alpha_1|^2 |\lambda_1|^{2k} \lambda_1 \left(1 + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)\right). \end{aligned}$$
Then
$$\frac{y_k^* A y_k}{y_k^* y_k} = \frac{|\alpha_1|^2 |\lambda_1|^{2k} \lambda_1 \left(1 + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)\right)}{|\alpha_1|^2 |\lambda_1|^{2k} \left(1 + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)\right)} = \lambda_1 \left(1 + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)\right).$$
For normal matrices, the terms $v_i^* v_j = 0$ vanish for $i \neq j$ since the eigenvectors can be chosen to be orthogonal, and the claim follows for these matrices as well.

□

Example 2.3 – Application of the power method

Let

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix},$$

$\lambda_1 = 2 + \sqrt{2} \approx 3.4142$ is the largest eigenvalue. We get for a starting vector y_0 the following iteration:

$$y_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad y_1 = \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix}, \quad y_2 = \begin{pmatrix} 10 \\ 14 \\ 10 \end{pmatrix}, \ldots$$

and the Rayleigh quotient is then:

$$\frac{y_1^* A y_1}{y_1^* y_1} = \frac{y_1^* y_2}{y_1^* y_1} = \frac{116}{34} \approx 3.4117.$$

Exercise

Show that the rate of convergence for this example is ≈ 0.59.

Example 2.4 – Application of the inverse power method

Let

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

Let $\mu = 3.41$ and we choose $y_0 = \begin{pmatrix} 1 \\ 1.4 \\ 1 \end{pmatrix}$. We obtain:

$$\frac{y_1^* (\mu I - A)^{-1} y_1}{y_1^* y_1} = \frac{y_1^* y_2}{y_1^* y_1} = -237.3288707 \approx \frac{1}{\mu - \lambda_1}$$

with which we then obtain $\lambda_1 \approx 3.414213562$ and all given digits agree with the largest eigenvalue, $2 + \sqrt{2}$.

Remark 2.7

To prevent overflow (the numbers becoming larger than the computer's largest representable number), we can set:

$$z_{k+1} := A y_k, \quad y_{k+1} := \frac{1}{\|z_{k+1}\|_\infty} z_{k+1},$$

where $\|z_{k+1}\|_\infty$ is the largest component of z_{k+1} in terms of magnitude. Another option is to normalize y_k to be of unit norm at every step, or every few steps. In this case,

$$z_{k+1} := A y_k, \quad y_{k+1} := \frac{1}{\|z_{k+1}\|_2} z_{k+1}.$$

Remark 2.8 – Application of the power method (Google)

What makes (or made) Google unique is an algorithm that provides a suitable order of results, the so-called PageRank algorithm, which is about characterizing the importance of web pages. Google determines the rank $r(P)$ of a page P by:

$$r(P) = \sum_{Q \in B_P} \frac{r(Q)}{|Q|}$$

where $B_P = \{\text{all pages that link to } P\}$, and where $|Q|$ is the number of links from Q (to any page!). This is a recursive definition, but one can see that the vector $y = (r(P_1), \dots, r(P_N))$ is an eigenvector of a matrix A associated to eigenvalue 1. The matrix $A = (a_{ij})$ is given by

$$a_{ij} = \begin{cases} \frac{1}{|P_j|} & \text{if } P_j \text{ links to } P_i, \\ 0 & \text{otherwise.} \end{cases}$$

Since the column sum of A is 1, 1 is the largest eigenvalue in magnitude (see the following exercise). Thus, the power method $y_{k+1} = Ay_k$ yields the desired eigenvector y.
For this problem, there is an article "The 25,000,000,000 Dollar Eigenvalue Problem: The Linear Algebra Behind Google" by Bryan and Leise[a] (2005) and Langville and Meyer[b] (2006) in SIAM Review.

[a] https://epubs.siam.org/doi/10.1137/050623280
[b] https://www.jstor.org/stable/j.ctt7t8z9

Exercise
Show that, if a matrix A with non-negative components has a column sum $\leq c$ *(i.e.* $\sum_{i=1}^n A_{i,j} \leq c$, $\forall j$*), then* $\lambda_j \leq c$ *for all* j*. If the column sum is* $= c$*, then* c *is the largest eigenvalue.*

Answer
Note that A *and* A^T *have the same characteristic polynomial, hence the roots of* χ_A *and* χ_{A^T} *are the same and the eigenvalues of* A^T *and* A *are equal. Apply the Gershgorin circle theorem on* A^T*:*

$$\operatorname{sp}(A) = \operatorname{sp}(A^T) \subset \bigcup_{j=1}^{n} \left\{ \mu \in \mathbb{C} : |\mu - a_{j,j}| \leq \sum_{k \neq j} |a_{k,j}| \right\}.$$

Since A *has non-negative components,* $\sum_{k\neq j} |a_{k,j}| = \sum_{k \neq j} a_{k,j}$*, hence* $|\mu - a_{j,j}| \leq c - a_{j,j}$*. Moreover, the reverse triangle inequality yields*

$$|\mu - a_{j,j}| \geq ||\mu| - |a_{j,j}||\,.$$

To show this, apply the triangle inequality to $|\mu| = |\mu \pm a_{j,j}|$ *and* $|a_{j,j}| = |a_{j,j} \pm \mu|$*. Thus,*

$$||\mu| - |a_{j,j}|| \leq c - a_{j,j},$$

and

$$\begin{aligned} |\mu| - a_{j,j} &\leq c - a_{j,j}, \quad \text{if } |\mu| \geq a_{j,j} \\ -|\mu| + a_{j,j} &\leq c - a_{j,j}, \quad \text{if } |\mu| \leq a_{j,j}. \end{aligned}$$

The first line yields $|\mu| \leq c$*, and the second line is for* $|\mu| \leq a_{j,j} \leq c$*. In both cases, we obtain* $|\mu| \leq c$*, i.e. all eigenvalues of* A *are smaller or equal to* c*.*

Now, if the column sum is equal to one we have $\mathbf{1}^T A = c\mathbf{1}^T$*, where* $\mathbf{1} = (1, \dots, 1)$*, which means that* c *is indeed an eigenvalue of* A*, since it is an eigenvalue of* A^T*.*

2.4 Simultaneous Iteration and QR Algorithm

Motivation
In the following, let A *be a real matrix whose eigenvalues satisfy*

$$|\lambda_1| > |\lambda_2| > |\lambda_3| > \ldots > |\lambda_n|$$

i.e. in particular, A *is diagonalizable. We want to compute not only the first but also the second, third, etc., eigenvalues.*

Review – Power method

Let y_0 be arbitrary and $y_{k+1} = Ay_k$. We know that:

$$y_k = \lambda_1^k \left(\alpha_1 \widetilde{v_1} + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)\right) = \lambda_1^k \left(|\alpha_1| \underbrace{\operatorname{sgn}(\alpha_1)\widetilde{v_1}}_{=:v_1} + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)\right),$$

where we assume that $\alpha_1 \neq 0$ and $\widetilde{v_1}$ is an eigenvector with $\|\widetilde{v_1}\|_2 = 1$. The vector v_1 is also an eigenvector of A with $\|v_1\|_2 = 1$. Thus, $\frac{y_k}{\|y_k\|} \to v_1$ if $\lambda_1 > 0$ and $(-1)^k \frac{y_k}{\|y_k\|} \to v_1$ if $\lambda_1 < 0$. In general, $(\operatorname{sgn} \lambda_1)^k \frac{y_k}{\|y_k\|} \to v_1$.

Algorithm 2.4 – Extension of the power method

Let q_0 be arbitrary with $\|q_0\|_2 = 1$. We consider the iteration $Aq_k = \lambda_1^{(k+1)} q_{k+1}$ with $\|q_{k+1}\|_2 = 1$ and $\operatorname{sgn} \lambda_1^{(k+1)} = \operatorname{sgn}(q_k^T A q_k)$. If k is large, this is $\operatorname{sgn} \lambda_1$. We know that $q_k \to v_1$ converges and $\lambda_1^{(k)} \to \lambda_1$ with the convergence rate $\left|\frac{\lambda_2}{\lambda_1}\right|$.

Idea (Computation of the next eigenvalue)
Now let λ_1, v_1 be known and we want to compute λ_2, v_2. We consider the orthogonal complement to $\mathbb{R}v_1$:

$$V = \{u \in \mathbb{R}^n \mid v_1^T u = 0\}.$$

We know $\dim V = n - 1$. *Furthermore, we consider the following mapping:*

$$L_1 := P \circ (A_{|V}) : V \to \mathbb{R}^n \to V$$

where P is an orthogonal projection: $\mathbb{R}^n \to V$. We have $q = \alpha v_1 + u \in \mathbb{R}^n$, where $u \in V$. Then $v_1^T q = \alpha$, with which we obtain α, noting that $\|v_1\| = 1$. Then $Pq = u = q - \alpha v_1 = (I - v_1 v_1^T)q$. Thus, for $u \in V$:

$$L_1(u) = (I - v_1 v_1^T)Au.$$

With the next theorem, we obtain that L_1 has precisely the remaining eigenvalues of A.

Theorem 2.6 – Eigenvalues of L_1

The eigenvalues of L_1 are $\lambda_2, \ldots, \lambda_n$.

Proof. By Schur's normal form, we know that for the matrix A, we have:

$$U^* A U = \begin{pmatrix} \lambda_1 & \star & \star \\ & \ddots & \star \\ 0 & & \lambda_n \end{pmatrix} = R$$

where $U = (u_1 | \ldots | u_n)$ is a unitary matrix with $u_1 = v_1$. The vectors $(u_2, \ldots, u_n)$ form an ONB of V, in particular $v_1^T u_j = 0$ for $j \geq 2$. Then

$$\begin{aligned} L_1(u_i) &= (I - v_1 v_1^T)Au_i = (I - v_1 v_1^T)(AU)_{\cdot,i} = (I - v_1 v_1^T)(UR)_{\cdot,i} \\ &= (I - v_1 v_1^T)(v_1 r_{1,i} + \ldots + u_{i-1} r_{i-1,i} + u_i \lambda_i) \\ &= u_2 r_{2,i} + \ldots + u_{i-1} r_{i-1,i} + u_i \lambda_i. \end{aligned}$$

From this formula, we can deduce the representation matrix of L_1 with respect to the basis $(u_2, \ldots, u_n)$, which looks as follows:

$$\begin{pmatrix} \lambda_2 & \star & \star \\ 0 & \ddots & \star \\ 0 & 0 & \lambda_n \end{pmatrix}$$

And this matrix has precisely the eigenvalues $\lambda_2, \ldots, \lambda_n$. □

Remark 2.9

It may be sometimes convenient to have L_1 in the $(u_1, \ldots, u_n)$ basis. In this case, 0 is another eigenvalue of L_1 and its matrix in the $(u_1, \ldots, u_n)$ basis reads

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \lambda_2 & \star & \star \\ 0 & 0 & \ddots & \star \\ 0 & 0 & 0 & \lambda_n \end{pmatrix}$$

The idea of orthogonalizing Au with respect to the previously obtained eigenvectors is called *deflation*.

Algorithm 2.5 – Computation of the second eigenvalue

To compute λ_2, we apply the power method to L_1: let $p_0 \in V$ be arbitrary with $\|p_0\|_2 = 1$. Then by the power method:

$$L_1(p_k) = (I - v_1 v_1^T)Ap_k = \lambda_2^{(k+1)} p_{k+1} \quad \text{with} \quad \|p_{k+1}\|_2 = 1, \quad \operatorname{sgn} \lambda_2^{(k+1)} = \operatorname{sgn} p_k^T L_1(p_k).$$

Thus, $\lambda_2^{(k)} \to \lambda_2$ and $p_k \to u_2$ etc., and we obtain Schur's normal form.

Algorithm 2.6 – Modification of the deflated power method

We do not compute first λ_1 and v_1, but we use simultaneous iteration for λ_1 and λ_2. This gives us the following algorithm: let q_0, p_0 be arbitrary with $\|q_0\|_2 = \|p_0\|_2 = 1$ and $q_0 \perp p_0$. Then we compute:

$$Aq_k = \lambda_1^{(k+1)} q_{k+1}, \quad \|q_{k+1}\|_2 = 1, \quad \operatorname{sgn} \lambda_1^{(k+1)} = \operatorname{sgn} q_k^T Aq_k,$$

and

$$(I - q_{k+1} q_{k+1}^T)Ap_k = \lambda_2^{(k+1)} p_{k+1}, \quad \|p_{k+1}\|_2 = 1, \quad \operatorname{sgn} \lambda_2^{(k+1)} = \operatorname{sgn} p_k^T Ap_k.$$

The orthogonality $p_{k+1} \perp q_{k+1}$ holds.

Algorithm 2.7 – Alternative notation for simultaneous iteration

We write alternatively, but equivalently to Algorithm 2.6,

$$A(q_k, p_k) = (q_{k+1}, p_{k+1}) \begin{pmatrix} \lambda^{(k+1)} & \alpha_{k+1} \\ 0 & \lambda_2^{(k+1)} \end{pmatrix}, \quad \alpha_{k+1} = q_{k+1}^T Ap_k.$$

The right-hand side can be obtained by applying the QR decomposition to the $n \times 2$ matrix $A(q_k, p_k)$.

Algorithm 2.8 – Generalization of the algorithm

We choose for U_0 an arbitrary orthogonal matrix (for example $U_0 = I$). We consider the iteration:

$$AU_k = U_{k+1} R_{k+1},$$

which is precisely the QR algorithm. Then

$$R_k \to \begin{pmatrix} \lambda_1 & \star & \star \\ 0 & \ddots & \star \\ 0 & 0 & \lambda_n \end{pmatrix}$$

converges and $U_k \to (u_1, \ldots, u_n)$ converges. By doing this, we obtain Schur's normal form.

Algorithm 2.9 – QR algorithm

We set as above: $Q_k := U_{k-1}^T U_k$. Then

$$Q_{k+1}R_{k+1} = U_k^T \underbrace{U_{k+1}R_{k+1}}_{=AU_k} = U_k^T AU_k = U_k^T \underbrace{AU_{k-1}}_{U_k R_k} Q_k = R_k Q_k.$$

Thus, we obtain the following algorithm:

1. $A_0 = A = Q_0 R_0$ (QR decomposition in the last step)
2. $A_1 = R_0 Q_0 = Q_1 R_1$ (QR decomposition)
3. $A_2 = R_1 Q_1 = Q_2 R_2$ (QR decomposition)
4. ...

This justifies Algorithm 2.1.

Since the QR algorithm is a just a power method applied to n orthonormal eigenvectors, with eigenvalues all distinct, it converges.

Remark 2.10 – Historical note on the QR algorithm

The algorithm goes back to Rutishauser, 1958, who, however, had used the LR decomposition. The QR algorithm that we consider here goes back to Francis, 1961, and Kublanovskaya, 1961.

Motivation (Outlook)

1. *The QR decomposition of an arbitrary matrix is performed with $\mathcal{O}(n^3)$ operations, which is too expensive. We therefore first transform A into Hessenberg form, i.e. $Q^T AQ = H$ (which is almost triangular form and has a diagonal below the main diagonal), which is about as expensive as a QR decomposition. If A is symmetric, then we obtain for H a tridiagonal matrix. For the QR decomposition of a Hessenberg matrix, we need only $\mathcal{O}(n^2)$ operations, or $\mathcal{O}(n)$ operations if A is symmetric. Then all A_k are again Hessenberg matrices, as we will see in the next section.*

2. *The convergence speed is very slow, about $\left|\frac{\lambda_2}{\lambda_1}\right|$. We consider the idea of shifting the matrix A, i.e. we consider $A_k - \mu_k I$, where we choose the parameter μ_k such that the convergence is accelerated. We will consider this procedure in the sixth section.*

3. *We have not yet captured complex eigenvalues. We cannot converge a real matrix A_k to a real upper triangular matrix if there are complex eigenvalues. In the case of a complex eigenvalue, A converges to a matrix in (almost) Schur's normal form with a 2×2 block on the main diagonal:*

$$\begin{pmatrix} a_{r,r}^{(k)} & a_{r,r+1}^{(k)} \\ a_{r+1,r}^{(k)} & a_{r+1,r+1}^{(k)} \end{pmatrix}$$

This converges to complex conjugate eigenvalues $\alpha \pm i\beta$, as we will see in Section 2.7.

2.5 Transformation to Hessenberg Form

Idea

We will use Householder transformations, defined for a given $v \in \mathbb{R}^m$ by $I - 2vv^T$. They can be understood as the linear operator performing a mirror symmetry with respect to the hyperplane orthogonal to v. In other words, if $x = \alpha v + \beta w$ with $w \perp v$, then $(I - 2vv^T)x = -\alpha v + \beta w$.

Lemma 2.3 – Householder transformations

Let $x \in \mathbb{R}^m$, there is $v \in \mathbb{R}^m$ with $\|v\|_2 = 1$ such that

$$(I - 2vv^*)x = (\star, 0, \dots, 0).$$

Proof. If $x_1 = 0$, define $v = (\|x\|_2, x_2, \dots, x_m)$. Then

$$2\frac{vv^*}{\|v\|_2^2}x = \frac{2}{\|v\|_2^2}\begin{pmatrix}\|x\|_2\\ x_2\\ \vdots\\ x_m\end{pmatrix}\sum_{j=2}^m |x_j|^2 = \frac{2}{2\sum_{j=2}^m |x_j|^2}\begin{pmatrix}\|x\|_2\\ x_2\\ \vdots\\ x_m\end{pmatrix}\sum_{j=2}^m |x_j|^2 = \begin{pmatrix}\|x\|_2\\ x_2\\ \vdots\\ x_m\end{pmatrix}.$$

If $x_1 \neq 0$, define $v = (\|x\|_2 + \sigma x_1, \sigma x_2, \dots, \sigma x_m)$, where $\sigma = \frac{\overline{x_1}}{|x_1|}$ ($\iff x_1 = |x_1|\overline{\sigma}$). Then

$$\begin{aligned}2\frac{vv^*}{\|v\|_2^2}x &= \frac{2}{\|v\|_2^2}\begin{pmatrix}\|x\|_2 + \sigma x_1\\ \sigma x_2\\ \vdots\\ \sigma x_m\end{pmatrix}\left(\|x\|_2 x_1 + \overline{\sigma}\sum_{j=1}^m |x_j|^2\right)\\ &= \frac{2}{2\|x\|_2^2 + 2\|x\|_2|x_1|}\begin{pmatrix}\|x\|_2 + \sigma x_1\\ \sigma x_2\\ \vdots\\ \sigma x_m\end{pmatrix}\overline{\sigma}\left(\|x\|_2|x_1| + \|x\|_2^2\right) = \begin{pmatrix}\overline{\sigma}\|x\|_2 + x_1\\ x_2\\ \vdots\\ x_m\end{pmatrix}.\end{aligned}$$

In both cases, we obtain

$$(I - 2vv^*)x = (\star, 0, \dots, 0). \tag{2.1}$$

□

Theorem 2.7 – Transformation to Hessenberg form

Let $A \in \mathbb{R}^{n\times n}$. This matrix can be transformed to Hessenberg form by $(n-2)$ Householder transformations:

$$Q^T A Q = H = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ 0 & \ddots & \ddots & \vdots \\ \vdots & & & \\ 0 & \cdots\ 0 & * & * \end{pmatrix},$$

where $Q := Q_{n-2} \cdot \ldots \cdot Q_1$, and $Q_i := \begin{pmatrix} I_i & 0 \\ 0 & I_{n-i} - 2w_i w_i^T\end{pmatrix}$ is an Householder transformation. The matrix H is an upper triangular matrix with entries on the sub-diagonal. If A is symmetric, H is tridiagonal.

Proof.

1. We choose $\widetilde{Q}_1 = I - 2w_1 w_1^T$ (with $w_1^T w_1 = 1$) as an $(n-1)\times(n-1)$ Householder matrix, such that

$$\widetilde{Q}_1 \begin{pmatrix} a_{2,1}\\ \vdots\\ a_{n,1}\end{pmatrix} = \begin{pmatrix}\star\\ 0\\ \vdots\\ 0\end{pmatrix}$$

Note that $\widetilde{Q}_1$ is symmetric. We obtain:

$$Q_1 = \begin{pmatrix} 1 & 0 \\ 0 & \widetilde{Q}_1 \end{pmatrix} \quad \text{and} \quad A^{(1)} = Q_1 A_1 Q_1^T = \left(\begin{array}{c|c} a_{1,1} & \\ \star & \\ 0 & \star \\ \vdots & \\ 0 & \end{array}\right) \begin{pmatrix} 1 & 0 \\ 0 & \widetilde{Q}_1 \end{pmatrix} = \left(\begin{array}{c|c} a_{1,1} & \\ \star & \\ 0 & \star \\ \vdots & \\ 0 & \end{array}\right).$$

2. We choose $\widetilde{Q}_2 := I - 2w_2w_2^T$ an $(n-2)\times(n-2)$ Householder matrix such that

$$\widetilde{Q}_2 \begin{pmatrix} a_{32}^{(1)} \\ \vdots \\ a_{n2}^{(1)} \end{pmatrix} = \begin{pmatrix} \star \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \text{and let } Q_2 = \begin{pmatrix} I_2 & 0 \\ 0 & \widetilde{Q}_2 \end{pmatrix} \Rightarrow A^{(2)} = Q_2 A^{(1)} Q_2^T = \left(\begin{array}{cc|c} \star & \star & \\ \star & \star & \\ 0 & \star & \\ \vdots & 0 & \star \\ \vdots & \vdots & \\ 0 & 0 & \end{array}\right).$$

3. We proceed inductively in this manner and obtain:

$$Q^T A Q = H$$

with $Q := Q_{n-2} \cdot \ldots \cdot Q_2 \cdot Q_1$ and H an upper triangular matrix with a nonzero subdiagonal.

4. For symmetric matrices, we have

$$H^T = Q^T A^T Q = Q^T A Q = H$$

Due to H having the all diagonals below the subdiagonal equal to zero, all diagonals of $H^T = H$ above the supdiagonal are zero. Therefore, H is tridiagonal.

□

Remark 2.11 – Consideration of computational effort

It requires about $\frac{5}{3}n^3$ operations for a general A and for symmetric matrices about $\frac{2}{3}n^3$ operations.

Theorem 2.8 – Inheritance of Hessenberg form

Let $H = QR$ be a QR decomposition. We set $\tilde{H} := RQ$. If H is a Hessenberg matrix, then $\tilde{H}$ is also a Hessenberg matrix. If H is tridiagonal and symmetric, then $\tilde{H}$ is also tridiagonal and symmetric.

Proof. By a Householder transformation, we obtain (for the QR decomposition):

$$Q_1 H = \left(\begin{array}{c|c} \star & \\ 0 & \\ \vdots & \star \\ 0 & \end{array}\right) \quad \text{and} \quad Q_1 := I - 2w_1w_1^T \quad \text{with} \quad w_1 = \begin{pmatrix} \star \\ \star \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

The fact that H is an Hessenberg matrix is used to have only two nonzero components in w_1, since $w_1 = (\star, \sigma H_{2,1}, \ldots, \sigma H_{n,1})$ for some $\sigma \in \mathbb{C}$, $|\sigma| = 1$. See the proof of Lemma 2.3 for more details. Then

$$RQ_1 = \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \end{pmatrix} \begin{pmatrix} * & * & 0 \\ * & * & 0 \\ 0 & 0 & I_{n-2} \end{pmatrix} = \begin{pmatrix} * & * & * \\ * & * & * \\ 0 & 0 & * \end{pmatrix}$$

i.e. we obtain an upper triangular matrix with an entry on the subdiagonal. More generally, one has $\widetilde{Q}_k = I_{n-k+1} - 2w_k w_k^T$ where $w_k = (\star, \star, 0, \dots, 0) \in \mathbb{R}^{n-k+1}$, and define

$$Q_k = \begin{pmatrix} I_{k-1} & 0 \\ 0 & \widetilde{Q}_k \end{pmatrix} = \begin{pmatrix} I_{k-1} & 0 & 0 & 0 \\ 0 & \star & \star & 0 \\ 0 & \star & \star & 0 \\ 0 & 0 & 0 & I_{n-k-1} \end{pmatrix}.$$

Letting $Q = Q_1 \dots Q_n$, one obtains the claim. When H (not necessarily Hessenberg) is tridiagonal, one can show that Q is in Hessenberg form, and so is RQ. This means that $\tilde{H} = RQ$ is in Hessenberg form. Note that $\tilde{H} = RQ = Q^T QRQ = Q^T HQ$, and the symmetry of H yields the symmetry of $\tilde{H}$. A symmetric Hessenberg matrix is tridiagonal, so finally $\tilde{H}$ is symmetric and tridiagonal. □

Algorithm 2.10 – Modification of the QR algorithm

1. We perform a pre-transformation, i.e. we bring $Q^T AQ = H_0$ to Hessenberg form. This requires $\mathcal{O}(n^3)$ steps.
2. Apply the classical QR algorithm to H_0, i.e. $H_k = Q_k R_k$. This requires generally $\mathcal{O}(n^2)$ operations, and only $\mathcal{O}(n)$ operations in the symmetric case.
3. $H_{k+1} := R_k Q_k$, which also requires generally $\mathcal{O}(n^2)$ operations, and only $\mathcal{O}(n)$ operations in the symmetric case.

2.6 QR Algorithm with Shift

Motivation
We will now always assume that A has only real eigenvalues that are pairwise distinct, i.e. $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$. Moreover, we work on the Hessenberg matrix $H = Q^T AQ$. We want to improve the convergence speed with a shift idea.

Idea (Shift)
We expect after the fourth section that we have the convergence rate

$$\left| h_{i+1,i}^{(k)} \right| = \mathcal{O}\left(\left| \frac{\lambda_{i+1}}{\lambda_i} \right|^k \right).$$

This can be very slow under certain circumstances. We consider the shifted matrix $\tilde{H} = H - \mu I$, which has the eigenvalues $\lambda_i - \mu$. Applying the QR algorithm here, we obtain:

$$\left| \tilde{h}_{i+1,i}^{(k)} \right| = \mathcal{O}\left(\left| \frac{\lambda_{i+1} - \mu}{\lambda_i - \mu} \right|^k \right).$$

This converges very quickly if $\mu \approx \lambda_{i+1}$.

Convention
We can assume without loss of generality that H is a non-reduced Hessenberg matrix, i.e. for all i, $h_{i+1,i} \neq 0$. If an element were zero, then the matrix would have the form:

$$H = \begin{pmatrix} H_1 & \star \\ 0 & H_2 \end{pmatrix}$$

and then the eigenvalues of H are the eigenvalues of H_1 and H_2, so the QR algorithm would be used separately on H_1 and H_2.

Theorem 2.9

Let H be a non-reduced Hessenberg matrix and μ an eigenvalue of H. Let $H - \mu I = QR$ be the QR decomposition. We consider $\tilde{H} := RQ + \mu I$. Then $\tilde{h}_{n,n} = \mu$ and $\tilde{h}_{n,n-1} = 0$, i.e. the matrix decomposes after one QR step.

Proof. We consider $H - \mu I$ with $h_{i+1,i} \neq 0$ for all i. This implies that the first $n-1$ columns of this matrix are linearly independent (because column j has a component at index $j+1$ which does not appear in all previous columns). We can write:

$$Q^T(H - \mu I) = \begin{pmatrix} R_{n-1} & * \\ 0 & r_{n,n} \end{pmatrix} =: R$$

This is the QR decomposition, which always exists, and R_{n-1} must be invertible because it is a square matrix of linearly independent columns. Because $H - \mu I$ is singular, we must have $r_{n,n} = 0$, so the last row of RQ must also be zero. Thus, we have:

$$\tilde{H} = RQ + \mu I = \begin{pmatrix} * & * & * & * \\ \ddots & * & * & * \\ & \ddots & * & * \\ & & 0 & \mu \end{pmatrix}$$

□

If one does not know the eigenvalue μ, one can use the iterative values $h_{n,n}^{(k)}$ as approximations.

Algorithm 2.11 – QR algorithm with shift (for real eigenvalues)

Without loss of generality, let H_0 be a non-reduced Hessenberg matrix. We consider $H_k - h_{n,n}^{(k)} I = Q_k R_k$ and $H_{k+1} := R_k Q_k + h_{n,n}^{(k)} I$ for $k = 1, 2, \ldots$ until

$$\left| h_{n,n-1}^{(k)} \right| \leq \texttt{eps} \left(\left| h_{n,n}^{(k)} \right| + \left| h_{n-1,n-1}^{(k)} \right| \right)$$

where `eps` is the machine precision. Then we accept $h_{n,n}^{(k)}$ as an eigenvalue. We start again with the submatrix $(h_{ij}^{(k)})_{i,j=1}^{n-1}$ until we finally reach a 1×1 matrix.

Remark 2.12 – Convergence speed

Let

$$H - h_{n,n} I = \begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ & \ddots & * & * & * \\ & & \ddots & * & b \\ & & & \varepsilon & 0 \end{pmatrix} \Rightarrow Q_{n-2} \cdot Q_{n-3} \cdot \ldots \cdot Q_1 \cdot (H - h_{n,n} I) = \begin{pmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ & \ddots & * & * & * \\ & & 0 & a & b \\ & & & \varepsilon & 0 \end{pmatrix}$$

where the Q_i are defined as in Theorem 2.8, i.e. we have an identity matrix and once a 2×2 block on the main diagonal. Moreover, ε and b remain unchanged: the matrix Q_i is constructed to only act on the i-th column of $H - h_{n,n} I$. In the symmetric case, $b = \varepsilon$. We still need to compute the QR decomposition of $\begin{pmatrix} a & b \\ \varepsilon & 0 \end{pmatrix}$. It is:

$$\begin{pmatrix} a & b \\ \varepsilon & 0 \end{pmatrix} = \underbrace{\frac{1}{\sqrt{a^2 + \varepsilon^2}} \begin{pmatrix} a & -\varepsilon \\ \varepsilon & a \end{pmatrix}}_{=:\tilde{Q}} \underbrace{\begin{pmatrix} \sqrt{a^2 + \varepsilon^2} & \frac{ba}{\sqrt{a^2+\varepsilon^2}} \\ 0 & -\frac{b\varepsilon}{\sqrt{a^2+\varepsilon^2}} \end{pmatrix}}_{=:\tilde{R}}.$$

Then

$$\tilde{R}\tilde{Q} = \begin{pmatrix} * & * \\ -\frac{b\varepsilon^2}{a^2+\varepsilon^2} & * \end{pmatrix}$$

Let $Q_{n-1} := \begin{pmatrix} I_{n-2} & 0 \\ 0 & \tilde{Q} \end{pmatrix}$ and $R = \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ & \ddots & * & * \\ & & 0 & \tilde{R} \\ & & 0 & \end{pmatrix}$, we investigate:

$$H - h_{n,n}I = \underbrace{Q_1 \cdot \ldots \cdot Q_{n-1}}_{=:Q} R,$$

and it is

$$\tilde{H} = RQ + h_{n,n}I = \begin{pmatrix} * & * & * & * & * \\ \ddots & * & * & * & * \\ & \ddots & * & * & * \\ & & \ddots & * & * \\ & & & -\frac{b\varepsilon^2}{a^2+\varepsilon^2} & * \end{pmatrix}.$$

We see that the left element becomes smaller at every iteration, and as soon as this is below the machine precision, we accept $h_{n,n}$ as an eigenvalue. If $\varepsilon^2 \ll a^2$, we obtain quadratic convergence, i.e. from $h_{n,n-1} = \mathcal{O}(\varepsilon)$ it follows that $\tilde{h}_{n,n-1} = \mathcal{O}(\varepsilon^2)$. For symmetric matrices, we even obtain cubic convergence (because $b = \varepsilon$), i.e. from $h_{n,n-1} = \mathcal{O}(\varepsilon)$ it follows that $\tilde{h}_{n,n-1} = \mathcal{O}(\varepsilon^3)$.

Remark 2.13 – Stability consideration of the QR algorithm

We finally obtain that $\hat{Q}^T A \hat{Q} = \hat{R}$ is Schur's normal form. The QR algorithm is stable in the sense of backward analysis, i.e. if $\hat{R} = \hat{Q}^T \hat{A} \hat{Q}$ is Schur's normal form of a perturbed matrix, then:

$$\|A - \hat{A}\|_2 \le C \cdot \texttt{eps} \cdot \|A\|_2, \quad \hat{Q}^T\hat{Q} = I + F \quad \text{with} \quad \|F\|_2 \le c \cdot \texttt{eps}.$$

Proof. Wilkinson, The Algebraic Eigenvalue Problem, 1965. □

2.7 Computation of Complex Eigenvalues

Motivation

We now investigate complex, non-real eigenvalues of real matrices $A \in \mathbb{R}^{n\times n}$, which occur in pairs of conjugate complex eigenvalues. Here, we will iteratively compute the real Schur normal form.

Theorem 2.10 – Real Schur normal form

For $A \in \mathbb{R}^{n\times n}$, there exists an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ such that

$$Q^T A Q = \begin{pmatrix} R_{11} & R_{12} & \dots & R_{1m} \\ & R_{22} & & \vdots \\ & & \ddots & \vdots \\ & & & R_{mm} \end{pmatrix}$$

where each R_{ii} is either a real number or a real 2×2 matrix with conjugate complex eigenvalues.

Proof. We proceed by induction on the number of pairs of complex conjugate eigenvalues, k. If $k = 0$, then A has only real eigenvalues, and we can choose $Q = U$ real in Schur's normal form, as in the earlier proof. For the induction step: A has a pair of complex conjugate eigenvalues, i.e. we have an eigenvalue $\lambda = \alpha + i\beta$ and $\bar{\lambda} = \alpha - i\beta$ with $\beta \neq 0$. Then there are linearly independent eigenvectors $v = x + iy$ and $\bar{v} = x - iy$ (since from $Av = \lambda v$ it follows that $\bar{A}\bar{v} = \bar{\lambda}\bar{v}$). It is $Av = \lambda v = (\alpha + i\beta)(x + iy) = (\alpha x - \beta y) + i(\alpha y + \beta x)$. On the other hand, $Av = A(x + iy)$. By coefficient comparison, we obtain:

$$Ax = (\alpha x - \beta y), \quad Ay = \alpha y + \beta x, \quad \text{i.e.} \quad A(x, y) = (x, y) \cdot \begin{pmatrix} \alpha & \beta \\ -\beta & \alpha \end{pmatrix}.$$

Since $v, \bar{v}$ are linearly independent, x, y must also be linearly independent, because:

$$(v, \bar{v}) = (x, y) \cdot \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}, \quad \det \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix} = -2i \neq 0.$$

The vectors x and y span a two-dimensional subspace of $\mathbb{R}^n$, which A maps into itself. Let (u_1, u_2) be an ONB of this subspace, which we extend to an ONB $(u_1, \ldots, u_n)$ of $\mathbb{R}^n$. We set $U := (u_1, \ldots, u_n) \in \mathbb{R}^{n \times n}$, which is an orthogonal matrix. It holds that:

$$AU = U \begin{pmatrix} R_{11} & \star \\ 0 & \tilde{A} \end{pmatrix}$$

where R_{11} is a 2×2 matrix with a pair of complex conjugate eigenvalues. By the induction hypothesis, there exists $\tilde{Q}$ such that $\tilde{Q}^T \tilde{A} \tilde{Q}$ has the block triangular form, from which the claim follows immediately. We finally set:

$$Q := \begin{pmatrix} I_2 & 0 \\ 0 & \tilde{Q} \end{pmatrix} U.$$

□

Idea (QR algorithm for complex eigenvalues)
We apply the shifted QR algorithm, with the additional knowledge that there are two complex conjugate eigenvalues. We consider the following iteration for a complex μ_k:

1. $H_k - \mu_k I = Q_k R_k$
2. $H_{k+1} := R_k Q_k + \mu_k I$
3. $H_{k+1} - \bar{\mu}_k I = Q_{k+1} R_{k+1}$
4. $H_{k+2} := R_{k+1} Q_{k+1} + \bar{\mu}_k$

Here, Q_k is of course not orthogonal but unitary.

Theorem 2.11

If H_k is real, then we can choose the QR decomposition such that H_{k+2} is real again.

Proof. We know that:

$$H_{k+1} = R_k Q_k + \mu_k I = Q_k^*(H_k - \mu_k I) Q_k + \mu_k I = Q_k^* H_k Q_k$$

and analogously we obtain:

$$H_{k+2} = Q_{k+1}^* H_{k+1} Q_{k+1} = (Q_k Q_{k+1})^* H_k (Q_k Q_{k+1})$$

It suffices to show that $Q_k Q_{k+1}$ is real. We compute with the help of the above calculations:

$$\begin{aligned} Q_k Q_{k+1} R_{k+1} R_k &= Q_k (H_{k+1} - \bar{\mu}_k I) R_k = Q_k (R_k Q_k + \mu_k I - \bar{\mu}_k I) R_k \\ &= (Q_k R_k)^2 + (\mu_k - \bar{\mu}_k) Q_k R_k \\ &= (H_k - \mu_k I)^2 + (\mu_k - \bar{\mu}_k)(H_k - \mu_k I) \\ &= (H_k - \mu_k I)(H_k - \mu_k I + \mu_k I - \bar{\mu}_k I) \\ &= H_k^2 - 2\mathrm{Re}(\mu_k) H_k + |\mu_k|^2 I =: M_k \end{aligned}$$

and we see that M_k is real. Since $R_{k+1}R_k$ is a triangular matrix and Q_kQ_{k+1} is a unitary matrix, we have computed the QR decomposition of M_k. We can choose the QR decomposition such that the diagonal elements of R_k, R_{k+1} are real, and the upper triangular property of $R_{k+1}R_k$ shows that Q_kQ_{k+1} is also real. □

Remark 2.14 – Uniqueness of the QR decomposition

The QR decomposition is unique up to multiplication by a diagonal matrix.

Remark 2.15 – Computational effort

We want to compute H_{k+2} from H_k only with real operations: The computation of the matrix M_k requires $\mathcal{O}(n^3)$ operations. We show now that we can compute H_{k+2} from H_k in $\mathcal{O}(n^2)$ real operations.

Theorem 2.12

Let $A \in \mathbb{R}^{n\times n}$ and $H = Q^TAQ$ be a Hessenberg matrix with $h_{i+1,i} \neq 0$ for $i = 1, \dots, n-1$. Then Q and H can be determined from the first column of Q.

Proof. Let $Q = (q_1, \dots, q_n)$. We have $AQ = QH$, i.e. $Aq_i = \sum_{j=1}^{i+1} q_j h_{ji}$. On the other hand, $Q^TAQ = H$, i.e. $q_j^TAq_i = h_{ji}$. We take q_1 as given. Then we know that $h_{11} = q_1^TAq_1$. Then q_2 is a multiple of $Aq_1 - h_{11}q_1$. Thus, we can determine q_2 up to the sign uniquely (because $\|q_2\|_2 = 1$). Thus, we also obtain h_{12}, h_{21}, h_{22}. By induction, the claim follows. □

Algorithm 2.12 – Francis' QR step

1. We compute the first column of M_k: $M_ke_1 = H_k(H_ke_1) - 2\mathrm{Re}(\mu_k)(H_ke_1) + |\mu_k|^2e_1$, which takes $\mathcal{O}(n^2)$ operations.
2. We compute the Householder matrix Q_0 with $Q_0(M_ke_1) = \alpha e_1$, which is a reflection, and it is:
$$Q_0 = \begin{pmatrix} \star & \star & \star & \\ \star & \star & \star & 0 \\ \star & \star & \star & \\ & 0 & & I \end{pmatrix}.$$
The 3×3 block comes from $H_ke_1 = \alpha_1e_1 + \alpha_2e_2$ (because H is Hessenberg), and thus $H_k(H_ke_1) = \alpha_1H_ke_1 + \alpha_2H_ke_2 = \beta_1e_1 + \beta_2e_2 + \beta_3e_3$. So the Householder vector for M_ke_1 only involves three nonzero components.
3. We transform $Q_0^TH_kQ_0$ into Hessenberg form $\tilde{H}$ in $\mathcal{O}(n^2)$ operations with Householder matrices $\tilde{Q}_1, \dots, \tilde{Q}_{n-3}$. Then we compute $Q^TH_kQ = \tilde{H}$ where $Q := Q_0\tilde{Q}_1 \cdot \ldots \cdot \tilde{Q}_{n-3}$. It holds $\tilde{H} = H_{k+2}$.

Proof. When computing the Hessenberg matrix $\tilde{H}$ of H_k, we know that the matrices $\tilde{H}$ and Q are fully determined from the first column of Q. We focus now on obtaining that first column. Write $H_k = (h_1|\dots|h_n)$, we have:

$$Q_0^TH_kQ_0 = \begin{pmatrix} * & * & * & \\ * & * & * & 0 \\ * & * & * & \\ & 0 & & I \end{pmatrix} \begin{pmatrix} h_1 & h_2 & h_3 & \dots & h_n \end{pmatrix} \begin{pmatrix} * & * & * & \\ * & * & * & 0 \\ * & * & * & \\ & 0 & & I \end{pmatrix}$$
$$= \begin{pmatrix} * & * & * & \\ * & * & * & 0 \\ * & * & * & \\ & 0 & & I \end{pmatrix} \begin{pmatrix} LC(h_1,h_2,h_3) & LC(h_1,h_2,h_3) & LC(h_1,h_2,h_3) & h_4 & \dots & h_n \end{pmatrix},$$

where $LC(v_1, \ldots, v_k)$ denotes a linear combination of vectors $v_1, \ldots, v_k$, and the coefficients of the linear combination may be different between all occurences of the notations "LC". Thus,

$$Q_0^T H_k Q_0 = \begin{pmatrix} * & * & * & \\ * & * & * & 0 \\ * & * & * & \\ & 0 & & I \end{pmatrix} \begin{pmatrix} * & * & * & & & & \\ * & * & * & & & & \\ * & * & * & & & * & \\ * & * & * & \ddots & & & \\ 0 & 0 & 0 & \ddots & \ddots & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\ 0 & 0 & 0 & \ldots & 0 & * & * \end{pmatrix} = \begin{pmatrix} * & * & * & & & & \\ * & * & * & & & & \\ \star & * & * & & & * & \\ \star & \star & * & \ddots & & & \\ 0 & 0 & 0 & \ddots & \ddots & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \\ 0 & 0 & 0 & \ldots & 0 & * & * \end{pmatrix}$$

We want to eliminate the $\star$-entries in the first column, which we can do using Householder matrices of the form

$$\tilde{Q}_j = \begin{pmatrix} I_j & 0 & 0 \\ & \star \;\; \star \;\; \star & \\ 0 & \star \;\; \star \;\; \star & 0 \\ & \star \;\; \star \;\; \star & \\ 0 & 0 & I_{n-j-3} \end{pmatrix}, \quad j = 1, \ldots, n-3.$$

Then $\tilde{Q}_i e_1 = e_1$ for $i = 1, \ldots, n-3$, so $Qe_1 = Q_0 e_1$. Since $Q_0 (M_k e_1) = \alpha e_1$ and $Q_0^{-1} = Q_0^T = Q_0$, $Q_0 e_1$ is a multiple of $M_k e_1$. On the other hand, we knew that $M_k e_1 = (Q_k Q_{k+1})(R_k R_{k+1}) e_1 = (Q_k Q_{k+1})\beta e_1$. Then $Q_k Q_{k+1} e_1$ is also a multiple of $M_k e_1$. We deduce that $Q_k Q_{k+1} e_1$ is a multiple of $Q_0 e_1 = Q e_1$, and since the columns of $Q_k Q_{k+1}$ and Q are all normalized we get $Q_k Q_{k+1} e_1 = \pm Q_0 e_1 = Q e_1$. With a suitable choice of signs, $Q_k Q_{k+1} e_1 = Q e_1$.

Now, we have

$$\tilde{H} = Q^T H_k Q \text{ and } H_{k+2} = (Q_k Q_{k+1})^T H_k (Q_k Q_{k+1}).$$

We know that H_k is a Hessenberg matrix, so H_{k+1} and *a fortiori* H_{k+2} as well. Hence, the two equations above are two reductions of H_k to Hessenberg form, and the first column of Q and $Q_k Q_{k+1}$ agree. Because the Hessenberg reduction is completely determined by the first column of the orthogonal matrix, both reductions are the same, hence

$$Q = Q_k Q_{k+1}, \quad \tilde{H} = Q^T H_k Q = H_{k+2}.$$

□

Remark 2.16 – Termination of iteration

We terminate the iteration if for $\ell = n$ (real eigenvalues) or $\ell = n-1$ (complex eigenvalues) it holds:

$$\left|h_{\ell,\ell-1}^{(k)}\right| \leq \text{eps}\left(\left|h_{\ell-1,\ell-1}^{(k)}\right| + \left|h_{\ell,\ell}^{(k)}\right|\right)$$

1. If $\ell = n$, we accept $h_{n,n}^{(k)}$ as the eigenvalue and restart with $\left(h_{ij}^{(k)}\right)_{i,j=1}^{n-1}$.
2. If $\ell = n-1$, we accept the eigenvalues of the lower right 2×2 block of H_k as eigenvalues of A and restart with $\left(h_{ij}^{(k)}\right)_{i,j=1}^{n-2}$.

2.8 Computation of Singular Values

Theorem 2.13 – About singular values

For $A \in \mathbb{R}^{m \times n}$, there exist orthogonal matrices $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ such that

$$A = U \Sigma V^T$$

with $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m \times n}$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$, where $p = \min\{m, n\}$. The σ_j are called singular values of A, which are uniquely determined.

Proof. Let $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ with $\|x\|_2 = \|y\|_2 = 1$. Moreover, let $Ax = \sigma y$ with

$$\sigma = \|A\|_2 = \max_{\|v\|=1} \|Av\|_2.$$

Thus, x can be chosen such that $\|Ax\| = \|A\|$. Let V_1 be an orthogonal $n \times n$ matrix with x in the first column (i.e. we extend x to an ONB of $\mathbb{R}^n$) and U_1 an orthogonal $m \times m$ matrix with y in the first column. Then

$$A_1 := U_1^T A V_1 = \left(\begin{array}{c|c} \sigma & \\ 0 & \\ \vdots & \star \\ 0 & \end{array} \right) = \begin{pmatrix} \sigma & w^T \\ 0 & \tilde{A}_1 \end{pmatrix},$$

where $w \in \mathbb{R}^{n-1}$ is a suitable vector and $\tilde{A}_1 \in \mathbb{R}^{(m-1)\times(n-1)}$ is a suitable matrix. We consider

$$\left\| A_1 \begin{pmatrix} \sigma \\ w \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} \sigma^2 + w^T w \\ \star \end{pmatrix} \right\|_2 \geq \sqrt{\sigma^2 + w^T w}.$$

On the other hand,

$$\|A_1\|_2 = \max_{\|v\|_2=1} \|A_1 v\|_2 = \max_{\|v\|_2=1} \|U_1^T A V_1 v\|_2 = \max_{\|u\|_2=1} \|U_1^T A u\|_2 = \max_{\|z\|_2=1} \|Az\|_2 = \|A\|_2 = \sigma.$$

We deduce $w = 0$, and thus $A_1 = \begin{pmatrix} \sigma & 0 \\ 0 & \tilde{A}_1 \end{pmatrix}$. By induction on $\tilde{A}_1$, the claim follows. □

Remark 2.17

Assume the same notation as in the above theorem and let $U = (u_1, \dots, u_m)$ and $V = (v_1, \dots, v_n)$. Furthermore, let $\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_p = 0$. Then

1. $\operatorname{rank} A = r$
2. $\operatorname{Ker} A = \operatorname{Span}\{v_{r+1}, \dots, v_n\}$ (and $v_{r+1}, \dots, v_n$ is an ONB of $\operatorname{Ker} A$)
3. $\operatorname{Image} A = \operatorname{Span}\{u_1, \dots, u_r\}$ (and $u_1, \dots, u_r$ is an ONB of $\operatorname{Image} A$)
4. $\|A\|_2 = \sigma_1$
5. For the Frobenius norm $\|\cdot\|_F$, we have:

$$\|A\|_F^2 := \sum_{i,j} a_{ij}^2 = \sum_{k=1}^r \sigma_k^2.$$

Remark 2.18

It holds that

$$A = U\Sigma V^T = \sum_{i=1}^r \sigma_i \underbrace{u_i v_i^T}_{\text{rank}=1}.$$

If the rank of A is small, then much of the information about it can be stored in the vectors u_i and v_i. The best approximation of A of rank $k \leq r$ is:

$$A \approx \sum_{i=1}^k \sigma_i u_i v_i^T.$$

This is called the "low-rank approximation of A", which is useful to approximate, e.g., large $N \times N$ matrices by means of r vectors of size N. This information is needed, for example, in data compression, physics, natural sciences (e.g. geoscience, astrophysics)...

Remark 2.19 – Further application

In search engines, one often uses the method of "latent semantic indexing," which is stored in a so-called term-document matrix, where for each search term and each document, it is recorded how often the term appears there. Thus, one replaces the term-document matrix with a low-rank approximation.

Remark 2.20 – Observation on the computation of singular values

One could apply the QR algorithm directly to A^TA (for $n \le m$) or to AA^T (for $m \le n$). The product formation is computationally expensive and rounding errors occur. Therefore, we now consider a better algorithm: If P and Q are orthogonal, then A and PAQ have the same singular values, since:

$$A = U\Sigma V^T \iff PAQ = \underbrace{PU}_{\tilde{P}}\,\Sigma\,\underbrace{V^TQ}_{\tilde{Q}}.$$

Note that $\tilde{P}$ and $\tilde{Q}$ are orthogonal matrices as products of orthogonal matrices. The following auxiliary theorem shows that we can transform any matrix A with such transformations P and Q into bidiagonal form, and thus the problem reduces to computing the singular values of a bidiagonal matrix.

Theorem 2.14

For $A \in \mathbb{R}^{m\times n}$ (with $m \ge n$ without loss of generality), there exist orthogonal matrices P, Q with

$$PAQ = \begin{pmatrix} B \\ 0 \end{pmatrix}, \quad B = \begin{pmatrix} \ddots & \ddots & 0 \\ 0 & \ddots & \ddots \\ 0 & 0 & \ddots \end{pmatrix}$$

i.e. B is an $n \times n$ bidiagonal matrix.

Proof. We use P_i and Q_i as Householder transformations. By multiplying from the left with P_1, we obtain:

$$P_1A = \left(\begin{array}{c|c} \star & \\ 0 & \\ \vdots & \star \\ 0 & \end{array}\right)$$

Multiplying from the right with $Q_1 = \begin{pmatrix} 1 & 0 \\ 0 & \tilde{Q}_1 \end{pmatrix}$, where $\tilde{Q}_1 \in \mathbb{R}^{(n-1)\times(n-1)}$ is a Householder matrix, gives:

$$P_1AQ_1 = \left(\begin{array}{c|cccc} \star & \star & 0 & \dots & 0 \\ \hline 0 & & & & \\ \vdots & & & \hat{A} & \\ 0 & & & & \end{array}\right)$$

By induction on a smaller matrix $\hat{A}$, the claim follows. □

Remark 2.21

We chase the "bad" elements on and below the subdiagonal until they "fall" out. This method is called "chasing" in the literature.

Remark 2.22 – Computation of Singular Values of a Tridiagonal Matrix

We have a matrix $B^TB =: H$ which is tridiagonal with real nonnegative eigenvalues, and consider a step of the QR algorithm for $\mu := h_{n,n}$:

$$M := H - \mu I = QR.$$

Then $\tilde{H} = RQ + \mu I = Q^T(H - \mu I)Q + \mu I = Q^T HQ$. We want to compute the eigenvalues of H without explicitly computing the matrix H or the matrix M. According to Theorem 2.12, $\tilde{H}$ and Q are uniquely determined by the first column of Q. Then $M = QR$ and $Q^T = Q_{n-1} \cdot \ldots \cdot Q_2 \cdot Q_1$ as a product of Householder transformations with

$$Q_j = \begin{pmatrix} I_{j-1} & & & \\ & * & * & \\ & * & * & \\ & & & I_{n-j-1} \end{pmatrix}, \quad j = 1, \ldots, n-1.$$

The first column of Q is $Qe_1 = Q_1e_1$, hence $Me_1 = QRe_1 = \alpha Qe_1 = \alpha Q_1 e_1$. This means that Q_1e_1 is a multiple of

$$Me_1 = \begin{pmatrix} h_{1,1} - \mu \\ h_{21} \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Then

$$Q_1 e_1 = \pm \frac{Me_1}{\|Me_1\|_2} = \begin{pmatrix} c \\ s \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{with} \quad c^2 + s^2 = 1, \quad \text{thus:} \quad Q_1 = \left(\begin{array}{cc|c} c & s & 0 \\ s & -c & \\ \hline 0 & & I_{n-2} \end{array}\right).$$

We now transform $Q_1^T HQ_1$ to tridiagonal form using Theorem 2.12: since H is symmetric, $Q_1^T HQ_1$ is also symmetric, and we are looking for $\tilde{Q}$ such that

$$\tilde{H} = \tilde{Q}^T(Q_1^T HQ_1)\tilde{Q} = (Q_1\tilde{Q})^T H (Q_1\tilde{Q}), \quad \tilde{Q} = \begin{pmatrix} I_2 & 0 \\ 0 & \check{Q} \end{pmatrix}, \quad \check{Q}^T\check{Q} = \check{Q}\check{Q}^T = I,$$

where $\tilde{H}$ is Hessenberg symmetric, hence tridiagonal. We have $Q_1\tilde{Q}e_1 = Q_1e_1 = Qe_1$. Thus, $Q_1\tilde{Q} = Q$, since the first column matches. Since $H = B^TB$, $\tilde{H} = Q^TB^TBQ = Q^TB^TP^TPBQ$ is tridiagonal if PBQ is bidiagonal. We transform BQ_1 to bidiagonal form using Theorem 2.14.

Algorithm 2.13 – Golub, Kahan, 1965

Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$. Then we perform the following algorithm:

1. We transform A with Householder matrices to bidiagonal form:

$$PAQ = \begin{pmatrix} B \\ 0 \end{pmatrix}, \quad B = \begin{pmatrix} \ddots & \ddots & 0 \\ 0 & \ddots & \ddots \\ 0 & 0 & \ddots \end{pmatrix}, \quad \beta = \|B\|_F := \sqrt{\sum_{i,j=1}^{n} b_{ij}^2}.$$

2. We use Remark 2.22 to find the singular values of B, which are the square root of the

eigenvalues of $H = B^T B$. We compute

$$Q_1 = \begin{pmatrix} \star & \star & 0 \\ \star & \star & \\ & 0 & I \end{pmatrix}$$

as in Remark 2.22. We then transform BQ_1 by "chasing" to bidiagonal form $\tilde{B}$. We repeat (with $\tilde{B}$ instead of B) until $|b_{n-1,n}| \leq \beta \cdot \text{eps}$, .i.e. until the off-diagonal term on row $n-1$ is "small enough to be considered zero". When this is done, $h_{n-1,n} \approx 0$, hence $h_{n,n}$ is a good approximation to the eigenvalue of H, which is the square of a singular value of B (which is itself a singular value of A).

3. We reduce the dimension by one and repeat with $(b_{ij})_{i,j=1}^{n-1}$, until we finally reach (almost-) diagonal form.

Remark 2.23

Due to equivalence to the QR algorithm for $H = B^T B$, we obtain cubic convergence, since this matrix is symmetric and tridiagonal.