University of Trieste

International School for Advanced Studies

The Abdus Salam International Centre for Theoretical Physics

Numerical Analysis

Lectures Notes

Author:
Marco Sciorilli

Gennaio 2021

Abstract

This document contains my notes on the course of Numerical Analysis held by Prof. Luca Heltai and Prof. Gianluigi Rozza for the Master Degree in Data Science and Scientific Computing at SISSA in the year 2020/2021. As they are a work in progress, every correction and suggestion is welcomed. Please, write me at: marco.sciorilli@gmail.com. A special thanks to Gabriele Sarti, as the basis for this work comes from his notes on the same course.

Contents

Chapter 1

Rounding/truncation error, conditional error

1.1 Floating-point representation

Real number representation in $\mathbb{F}$ (the set of the floating-point numbers):

$$x = (-1)^s \cdot (0.a_1 a_2 ... a_t) \cdot \beta^e = (-1)^s \cdot m \cdot \beta^{e-t} \quad with\ a \neq 0 \tag{1.1}$$

- s is a sign bit (1 or 0)
- β is the basis adopted by computer (usually it is 2)
- m is the mantissa of length t made by digits a, with $0 \leq a_i \leq \beta - 1$
- e is the exponent

The set of number representable by a machine is characterized by β, t and the range (L, U) of the exponent. It is commonly denoted as $\mathbb{F}(\beta, t, L, U)$.
The roundoff error occur when we replace $x \neq 0$ with its $\mathbb{F}$ representation, $\hat{x}$, and is defined as

$$\frac{|x - \hat{x}|}{|x|} \leq \frac{1}{2}\epsilon_M \quad with\ \epsilon_M = \beta^{1-t} \tag{1.2}$$

Where ϵ_M is the machine epsilon: the minimal variation representable by a machine

$$\epsilon_M \text{ the largest number } | \quad fl(1 + \epsilon_M) = fl(1) \tag{1.3}$$

Where $fl()$ is the floating-point representation of a number. $\frac{1}{2}\epsilon_M$ is the roundoff unit, $|x - \hat{x}|$ is the absolute error, and $\frac{|x-\hat{x}|}{|x|}$ is the relative error of the approximation operated. The relative error accounts for the order of magnitude of x.
0 is not part of $\mathbb{F}$ and is therefore handled separately. A number exceeding the lower bound is treated as 0 while numbers exceeding the upper bound is treated as inf. $\mathbb{F}$ is not homogeneously dense, but it is denser near 0, and less dense near infinity.
In $\mathbb{F}$ associativity and distributivity are not always respected, as for the case of the loss of significant digits. Indeterminate forms as $\frac{0}{0}$ and $\frac{inf}{inf}$ produces error flagged as NaN.

1.2 Complex numbers

The classic representation of complex number is

$$z = x + iy = \varphi e^{i\theta} = \varphi(cos\theta + isen\theta) \tag{1.4}$$

Where $i = \sqrt{-1}$, $x = Re(z)$, $y(Im)$ and $\varphi = \sqrt{x^2 + y^2}$.
z is a complex number ($\in \mathbb{C}$ with a real part x and an imaginary part y, both represented by two floating-point numbers. Its modulus is φ, and its complex conjugate is

$$\overline{z} = x - iy = \varphi e^{-i\theta} = \varphi(cos\theta - isen\theta) \tag{1.5}$$

The complex conjugate is used in the conjugate transposition of matrices

$$(A_{ij})^* = \overline{A_{ij}} \tag{1.6}$$

1.3 Matrices

Some properties of matrices are

- $A + B = (a_{ij}) + (b_{ij}) = (a_{ij} + b_{ij})$
- $\lambda A = (\lambda a_{ij})$
- $C_{m\times n} = A_{m\times p} B_{p\times n} = (c_{ij}) = \sum_{k=1}^{p} a_{ik} b_{kj}$

If a matrix is diagonal, its determinant is the product of diagonal elements. A matrix is lower/upper triangular if all the elements above/under the main diagonal are zero.
If $A \in \mathbb{R}^{m\times n}$ and its transpose $A^t \in \mathbb{R}^{n\times m}$, A is symmetrical if $A = A^t$. If $A = A^H = \overline{A}^t$, A is hermitian.

1.4 Vectors

A set of vectors $y_1, ..., y_m$ is linearly independent if

$$a_1 y_1 + ... + a_m y_m = 0 \Leftrightarrow a_1, ..., a_m = 0 \tag{1.7}$$

B is a basis for $\mathbb{R}^n$ or $\mathbb{C}^n$ if $B = y_1, ..., y_n$ and $y_1, ..., y_n$ are all independent vectors. Any vector w in $\mathbb{R}^n$ can then be written as

$$w = \sum_{k=1}^{n} a_k y_k \tag{1.8}$$

a_k are unique components of w in relation to B.
THe scalar dot product of v and w is defined as

$$(v, w) = w^t v = \sum_{k=1}^{n} a_k b_k \tag{1.9}$$

with a and b respectively components of v and w.
The modulus of a vector v is given by the euclidean norm formula

$$\|v\| = \sqrt{(v,v)} = \sqrt{\sum_{k=1}^{n} v_k^2} \tag{1.10}$$

The vector product (cross product) of $v, w \in \mathbb{R}^3$ is the vector u orthogonal to v and w, with modulus $|u| = |v||w|sin\alpha$.
$v \in \mathbb{C}^n$ is an eigenvector of $A \in \mathbb{C}^{n\times m}$ associated with eigenvalue λ if

$$Av = \lambda v \tag{1.11}$$

The eigenvalues of diagonal and triangular matrices are the elements on the diagonal.
A matrix is said to be positive definite if

$$z^t Az \geq 0 \ \forall \ z \in \mathbb{R}^n \tag{1.12}$$

1.5 Real Functions

If $f(\alpha) = 0$, α is a zero r root of f. It is called simple if $f'(\alpha) \neq 0$, multiple otherwise.
The space $\mathbb{P}_n$ of polynomials of degree $\leq n$ is defined as

$$p_n(x) = \sum_{k=0}^{n} a_k x^k \tag{1.13}$$

with a_k given coefficients.
The number of zeros cannot usually be estimated a priori (except for polynomials, where its $= n$). The value for p_n zeros cannot be computed with an explicit formula for $n \geq 5$.
Foundamental theorem of integration, for f continuous in $[a, b]$

$$F(x) = \int_a^x f(x)dt \ \forall x \in [a,b) \ \Rightarrow \ F'(x) = f(x) \ \forall \ x \in [a,b) \tag{1.14}$$

First mean-value theorem for integrals, for f continuous in $[a, b)$ and $x_1, x_2 \in [a, b)$ with $x_1 < x_2$

$$\exists \xi \in (x_1, x_2) \ s.t. \ f(\xi) = \frac{1}{x_2 - x_1} \int_{x_1}^{x_2} f(t)dt \tag{1.15}$$

$f \in [a, b]$ is differentiable in $x \in (a, b)$ if

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} \tag{1.16}$$

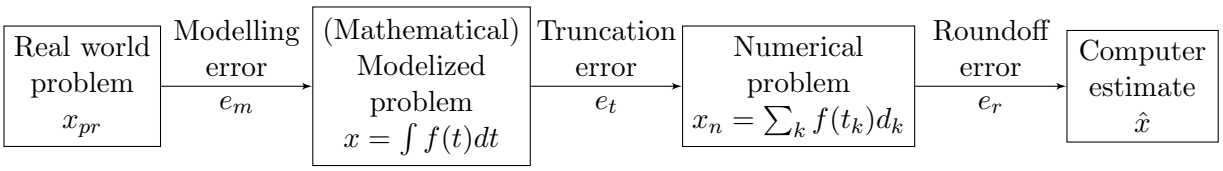
exist and is finite.
Mean value theorem: if $f \in C^\circ([a, b])$ and is differentiable in (a, b)

$$\exists \xi \in (a,b) \ s.t. \ f'(\xi) = \frac{f(b) - f(a)}{b - a} \tag{1.17}$$

Taylor expansion of p_n: if $f \in C^\circ([x_0 - c, x_0 + c])$ (a neighbovarphiod of x_0), f can be approximated in that interval as

$$T_n(x) = f(x_0) + (x - x_0)f'(x_0) + ... + \frac{1}{n!}(x - x_0)^n f^{(n)}(x_0) = \sum_{k=0}^{n} \frac{(x - x_0)^k}{K!} f^{(k)}(x_0) \tag{1.18}$$

1.6 Estimating errors



The sum of truncation error derived from reducing a problem to a finite set of operations and roundoff error coming from a machine representation is called computational error e_c

$$e_c^{abs} = |x - \hat{x}| \quad e_c^{rel} = \frac{|x - \hat{x}|}{|x|} \tag{1.19}$$

To convert a mathematical problem in numerical form we use discretization parameter h, positive.
if $(num) \to (mat)$ as $h \to 0$ the numericl process is said to be convergent.
If we can bound e_c as $e_c \leq Ch^p$ we say that the method is convergent of order p. if a lower bound $C'h^p \leq e_c$ also exists, we can approximate the final error.
Logarithmic scale is effective for numerical methods since lines slopes represent the order of convergence for each method. The semi-logarithmic scale is also used to visualize functions that span many orders of magnitude in y in a short x interval.
The computational cost is O (ops, operations) and can be constant, linear, polynomial, exponential, factorial, ecc.
Numerical approximation can be performed exclusively on well-posed problems, thats to say problems for which the solution:

- Exists
- Is unique
- Depends continuously on data

The total error is:

$$f(x) - \hat{f}(\hat{x}) = \underbrace{\hat{f}(\hat{x}) - f(\hat{x})}_{\substack{\text{computation} \\ \text{error} \\ (e_c = e_t + e_r)}} + \underbrace{f(\hat{x}) - f(x)}_{\substack{\text{propagated data} \\ \text{error} \\ \text{(independent from f)}}} \tag{1.20}$$

ex. finite differences approximation ($f'(x) = \lim_{h\to 0} \frac{f(x-h)-f(x)}{h}$)

- Truncation error (obtained through Taylors) $\sim \frac{1}{2}|(f''(x)|h + O(h^2)$
- Rounding error $\sim \frac{2\epsilon}{h}$ with ϵ =machine precision

The optimal h is therefore $h = 2\sqrt{\frac{\epsilon}{|f''(x)|}}$

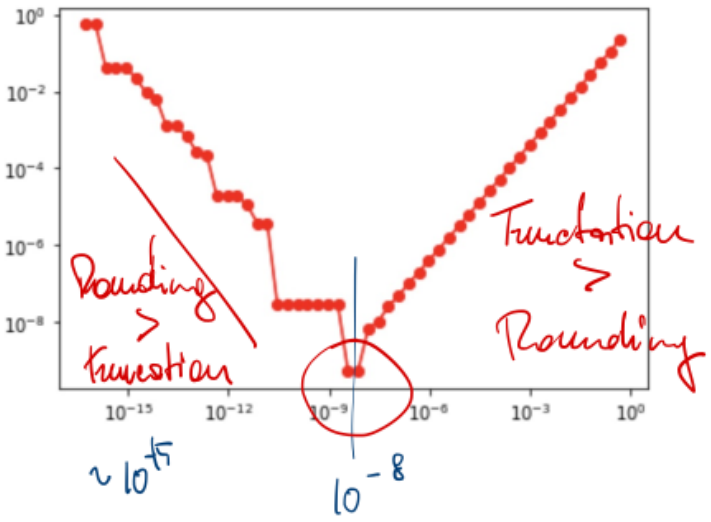


Figure 1.1: Plot of total error vs h. The optimal value of h which minimize the total error is reached when $h \sim \sqrt{\epsilon}$

Problem stability: small changes in input data produce small variation on the output. It is a synonymous of well-posedness.
Given δ a perturbation in data s.t. $d + \delta d \in D$, and $x + \delta x$ the perturbed solution, then

$$\forall d \in D \ \exists \eta(d) \text{ and } K \ s.t. \ \|\delta d\|_d < \eta \in D \Rightarrow \|\delta x\|_x < K\|\delta d\|_d \tag{1.21}$$

Condition numbers: it can be either relative or absolute and measure problem sensitiveness with regards to input data.
If we define $\Delta y = f(x) - f(\hat{x})$ and $\Delta x = x - \hat{x}$, we have that the relative condition numbers is

$$K_{rel} = \frac{\frac{\Delta y}{y}}{\frac{\Delta x}{x}} \approx |f'(x)| \frac{|x|}{|f(x)|} \tag{1.22}$$

And the absolute condition number is:

$$K_{abs} = \frac{\Delta y}{\Delta x} \quad (\text{if } f(x) \text{ or } x = 0\ ) \approx |f'(x)| \tag{1.23}$$

If $K \gg 1$ the problem is ill-posed (sensitive, unstable) and is thus not approximable through numerical methods.
A numerical approximation can be seen as a sequence of simpler approximating problems that converge to the original one

$$\lim_{n\to\infty} \|y_n - y\| = \lim_{n\to\infty} \|x_n - x\| = 0 \text{ is as } \lim_{n\to\infty} f_n(x) = f(x) \tag{1.24}$$

1.7 Banach Spaces

Given $\vec{v}$ over $\mathbb{R}$ or $\mathbb{C}$, a seminorm is a function $|\cdot| : V \to \mathbb{C}$ which satisfy:

- $|cf| = |c||f|\ \forall c \in \mathbb{C}$ (homogeneity)
- $|f + g| \leq |f| + |g|$ (triangular inequality)

$\vec{v}$ is a vector space and the norm is a linear mapping.
If $|f| = 0$ iff $f = 0$ (positive definite) is also verified, we have a norm. A vector space is said to be complete if every Cauchy sequence in that space converges to one of the space's elements.
A complete vector space with a norm is called Banach Space.
The scalar product is a mapping $V \times V \to \mathbb{C}$ which is:

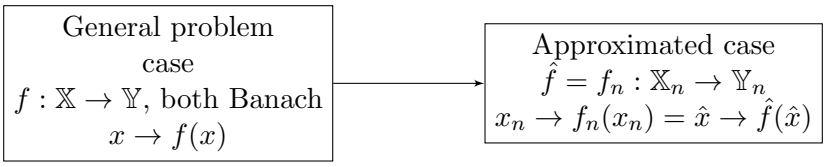
- Linear: $(\alpha_1 v_1 + \alpha_2 v_2, w) = \alpha_1(v_1, w) + \alpha_2(v_2, w)$
- Symmetric:$(v, w) = (\overline{w, v})$
- Positive definite: $(v_1, v_2) \geq 0 \ \ \forall v_1, v_2$ and $(v_1, v_2) = 0$ iff $v_1, v_2 = 0$

A Banach space with scalar product and a norm $\|f\| = (f, f)$ induced by the product is called Hilbert space.
Examples of norms in the Banach spaces:

- $\|x\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$
- $\|x\|_1 = \sum_{i=1}^{n} |x_i|$
- $\|x\|_2 = \sqrt{x_1^2 + x_2^2}$ (euclidean norm)
- $\|x\|_\infty = \sup_{1 \leq i \leq n}(x_i)$

In a finite-dimensional vector space (dimension is given by the number of vectors in the basis), all norms are equivalents:

$$\forall \|\cdot\|_a, \|\cdot\|_b \ \exists 0 \leq c_1 \leq c_2 \ s.t. \ c_1\|x\|_b \leq \|x\|_a \leq c_2\|x\|_b \tag{1.25}$$



1.8 Converge, consistency and Lax-Richtmyer

A numerical method is convergent if the approximation $\hat{f}_n$ of a problem f satisfies:

- $\lim_{n\to\infty} \|x_n - x\|_{\mathbb{X}} = 0$
- $\lim_{n\to\infty} \left\| \hat{f}_n(\underbrace{x_n}_{\text{approx.}}) - f(x) \right\|_{\mathbb{X}} = 0$

A numerical problem is consistent when, if $x \in \mathbb{X}_n \forall n$, we have that

$$\lim_{n\to\infty} \left\| f_n(\underbrace{x}_{\text{exact}}) - f(x) \right\| = 0 \tag{1.26}$$

Example 1. *Sum of two numbers*

- $\mathbb{X} : \mathbb{R}^2, \|x\|_{\mathbb{X}} = |x_1| + |x_2| = \|x\|_{l^1(\mathbb{R}^2)}$
- $\mathbb{R}, \|y\|_{\mathbb{Y}} = |y| = \|y\|_{l^1(\mathbb{R}^1}$

$$K_{rel} = \frac{|\Delta y|}{|\Delta x|} \cdot \frac{|x|}{|y|} \Rightarrow K_{rel} \leq \frac{|x_1| + |x_2|}{|x_1 + x_2|} \tag{1.27}$$

Result: *Unstable in* $\mathbb{F}$ *when* $x_1 \cong -x_2 \Rightarrow K_{rel} \to \infty$

- A convergent approximation is always stable.
- Finite differences are unstable since they are a sum of two numbers with close absolute value and opposite sign.
- For integration, $K_{\text{rel}} = \frac{\int |x|}{|\int x|}$, so it is ill-posed when $x \sim 0$.
- The condition number of a matrix A is $K_{\text{rel}} = \left\|A^{-1}\right\| \|A\|$
  This usually corresponds to

$$K(A) = \frac{\left|\lambda_{\text{MAX}(A)}\right|}{|\lambda_{\text{MAX}}(A)|} \tag{1.28}$$

The **Lax-Richtmyer theorem** says that if a problem is consistent, then stability and convergence are equivalent.

- Stability controls perturbation in data and their impact.
- Consistency controls bad approximation of a problem.
- Convergence controls bad discretizations of the problem space (and includes stability).

A method is consistent if the residual (error produced by plugging the exact solution in the scheme) goes to 0 as $h \to 0$

Chapter 2

Nonlinear equation

We may want to find the roots of the non linear functions ($\alpha \in \mathbb{R}$ *s.t.* $f(\alpha) = 0$) in a computational way. Most common approaches are iterative, sinche there is no explicit solving formula for $p \in \mathbb{R}^n$, with $n \geq 5$ (**Abel's theorem**).

2.1 Bisection method (linear convergence)

It is used to compute the root of a function f on interval $[a, b]$.
Constrains for convergence:

- f should be continuous on $[a, b]$.
- Interval end points should have different sign ($f(a)f(b) < 0$) to have at least 1 solution (**theorem of zeros for continuous functions**)

We generate a sequence of intervals whose length is halved at each step, with $x^{(k)}$ being the midpoint at step k.
The error of estimation at step k is:

$$\left|e^{(k)}\right| = \left|x^{(k)} - \alpha\right| < \frac{1}{2}\left|I^{(k)}\right| = \left(\frac{1}{2}\right)^{k+1}(b-a) \tag{2.1}$$

In order to ensure that the error $\left|e^{(k)} < \epsilon\right|$, we carry out K_{mm} iterations at least:

$$K_{mm} > \log_2\left(\frac{b-a}{\epsilon}\right) - 1 \tag{2.2}$$

The error does not decrease monotonically. The only possible stopping criterion is controlling the size of $I^{(k)}$.

2.2 Newton's method (Quadratic or linear convergence)

H is used to compute the root of a function f by using the values of f and f' (more efficient than bisection).
Constrains for convergence:

- $f : \mathbb{R} \to \mathbb{R}$ should be differentiable.
- x_0 is sufficiently close to α given f (estimate through graph and bisection).

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})^2},\ k = 0, 1, ... \tag{2.3}$$

If $f \in \mathcal{C}^2$, we have that $\lim_{k\to\infty} \frac{x^{(k+1)}}{(x^{(k)}-\alpha)^2} = \frac{f''(\alpha)}{2f'(\alpha)}$ than we have quadratic convergence.
If f has zeros with multiplicity $m > 1$, if $f'(x) \neq 0\ \forall x \in I(\alpha)$, the method converges linearly. To restore quadratic convergence, one can use the **modified Newton method**, or **adaptive Newton methods** if m is unknown.

$$x^{(k+1)} = x^{(k)} - m\frac{f(x^(k))}{f'(x^{(k)})}, \quad k = 0, 1, ... \tag{2.4}$$

(α of f has multiplicity m iff $f(\alpha) = ... = f^{(m-1)}(\alpha) = 0$ and $f^{(m)}(\alpha) \neq 0$).
Stopping criterion: Control of the movement

$$\left|x^{(k+1)} - x^{(k)}\right| < \epsilon \tag{2.5}$$

We can also perform a test on the residual which is valid only if $|f'(x)| \simeq 1\ \forall x \in I(\alpha)$, else it produces an over or underestimation of error

$$\left|r^{(k_{\min})}\right| = \left|f(x^{(k_{\min})})\right| < \epsilon \tag{2.6}$$

2.3 Secant method (sublinear convergence)

In case $f'(x)$ is not available, we can replace its value with an incremental ration based on previous values

$$x^{(k+1)} = x^{(k)} - \Big(\frac{f(x^{(k)}) - f(x^{k-1})}{x^{(k)} - x^{(k-1)}}\Big)^{-1} f(x^{(k)}) \tag{2.7}$$

Constrains for convergence:

- α has $m = 1$ (for superlinear).
- $x^{(0)}$ is selected in $I(\alpha)$ suitable.
- $f'(x) \neq 0\ \forall x \in I(\alpha)$

If $m = 1$ and $f \in \mathcal{C}^2(I(\alpha))$, $\exists c > 0$ s.t.

$$\left|x^{(k+1)} - \alpha\right| \leq c\left|x^{(k)} - \alpha\right|^P \quad \text{with } p \approx 1.618 \tag{2.8}$$

Else the method converges linearly.

2.4 Systems of nonlinear equations

Given $f_1, ..., f_n$ nonlinear functions in $x_1, ..., x_n$, we can set $f = (f_1, ..., f_n)^T$ and $\overline{x} = (x_1, ..., x_n)^T$ to write a system

$$\overline{f}(\overline{x}) = 0 \tag{2.9}$$

We can extend the Newton method to that system by replacing the f' with the Jacobian Matrix $J_{\overline{f}}$, as

$$(J_{\overline{f}})_{ij} = \frac{\partial f_i}{\partial x_j} \quad i, j = 1, ..., n \tag{2.10}$$

The secant method can also be adopted by recursively defining matrices B_k which are suitable approximation of $J_{\overline{f}}(x^0)$ (**Broyden Method**). This belongs to the family of quasi-newton methods.

2.5 Fixed point iterations

Given a function $\phi : [a, b] \to \mathbb{R}$, we want to find an α so that

$$\phi(\alpha) = \alpha \tag{2.11}$$

If α exists, it is called a **fixed point** of ϕ and it could be computed as follows:

$$x^{(k+1)} = \phi(x^{(k)}),\ k \geq 0 \quad \text{with } x^{(0)} \text{ initial guess} \tag{2.12}$$

ϕ is called the iteration function. The Newton method is a special case of fixed point iteration where

$$\phi_N(x) = x - \frac{f(x)}{f'(x)} \tag{2.13}$$

2.6 Global convergence

1. Iff $\phi(x)$ is continuous in $[a, b]$ and $\phi(x) \in [a, b]$ $\forall x \in [a, b]$ then there exists at least one $\alpha \in [a, b]$.
2. Moreover, if $\exists L < 1$ (**Asymptotic convergence factor**) s.t.

$$|\phi(x_1) - \phi(x_2)| \leq L|x_1 - x_2| \ \forall x_1, x_2 \in [a, b] \tag{2.14}$$

   then $\alpha \in [a, b]$ is unique and the iteration converges to α $\forall x^{(0)} \in [a, b]$

Proof. 1. From our assumptions we have that $g(x) = \phi(x) - x$ is continuous in $[a, b]$, with:

$$g(a) = \phi(a) - a \geq 0 \quad \text{and} \quad g(b) = \phi(b) - b \leq 0 \tag{2.15}$$

For theorem of zeroes for c functions, we know that g has at least 1 zeros, and thus $\exists \alpha$ for ϕ in $[a, b]$.

2. If two fixed points existed, we would have

$$|\alpha_1 - \alpha_2| = |\phi(\alpha_1) - \phi(\alpha_2)| \leq L|\alpha_1 - \alpha_2| < |\alpha_1 - \alpha_2| \tag{2.16}$$

which is absurd for $L < 1$. For x^0 in $[a, b]$ and $x^{(k+1)} = \phi(x^{(k)})$, we have

$$0 \leq \left|x^{(k+1)} - \alpha\right| = \left|\phi(x^{(k)}) - \phi(\alpha)\right| \leq L\left|x^{(k)} - \alpha\right| \Rightarrow \frac{\left|x^{(k)} - \alpha\right|}{x^{(0)} - \alpha} \leq L^k \tag{2.17}$$

The smaller the L, the faster the convergence.
Since $L < 1$, for $k \to \infty$, we notice

$$\lim_{k\to\infty} \left|x^{(k)} - \alpha\right| \leq \lim_{k\to\infty} L^k = 0 \tag{2.18}$$

Which is convergence.

□

2.7 Local convergence (Ostrowski's theorem)

If ϕ is a continuous and differentiable function on $[a, b]$, with fixed point α, and $|\phi'(\alpha)| < 1$ then $\exists\delta > 0$ s.t.

$$\left|x^{(0)} - \alpha\right| \leq \delta \quad \forall x^{(0)} \text{ in } [a, b] \tag{2.19}$$

for which the $x^{(k)}$ converges to α. It holds

$$\lim_{k\to\infty} \frac{x^{(k+1)} - \alpha}{x^{(k)} - \alpha} = \phi'(\alpha) \tag{2.20}$$

$\forall c$ s.t. $0 < |\phi'(\alpha)| < c < 1$, for large k: $\left|x^{(k+1)} - \alpha\right| \leq c\left|x^{(k)} - \alpha\right|$
If $|\phi'(\alpha) > 1|$, the method diverges. If $|\phi'(\alpha)| = 1$, it depends on the function.

2.8 Quadratic convergence

If $\phi \in C^2([a, b])$ and α is fixed point of ϕ, with ϕ having local convergence. Then, if $\phi'(\alpha) = 0$ and $\phi''(\alpha) \neq 0$, fixed point converges with order 2 and

$$\lim_{k\to\infty} \frac{x^{(k+1)} - \alpha}{(x^{(k)} - \alpha)^2} = \frac{1}{2}\phi''(\alpha) \tag{2.21}$$

2.9 Stopping criteria

Error estimator at step k is

$$\alpha - x^{(k)} = e^{(k)} \approx \frac{1}{(1 - \phi'(\alpha))}(x^{(k+1)} - x^{(k)}) \tag{2.22}$$

Satisfactory when we have quadratic convergence (since $\phi'(\alpha) = 0$) or when $-1 < \phi'(\alpha) < 0$, problems when $\phi'(\alpha) \simeq 1$.
In that case we can use the central of the residual as described for newton method.

2.10 Aitken method

If ϕ converges linearly to α, there must be a X s.t. $\phi(x^{(k)}) - \alpha = X(x^{(k)} - \alpha)$. This allows us to obtain a better estimate of $x^{(k+1)}$ than $\phi(x^{(k)})$ (the **Aktken's extrapolation formula, Stefferson's method**)

$$\alpha = x^{(k)} + \frac{(\phi(x^{(k)} - x^{(k)})}{(1-\lambda)} \quad \text{with}$$
$$\lambda^{(k)} = \frac{\phi(\phi(x^{(k)})) - \phi(x^{(k)})}{\phi(x^{(k)}) - x^{(k)}} \quad \text{given}$$
$$\lim_{k\to\infty} \lambda^{(k)} = \phi'(\alpha)$$
$$\Rightarrow x^{(k+1)} = x^{(k)} - \frac{(\phi(x^{(k)}) - x^{(k)})^2}{\phi(\phi(x^{(k)})) - 2\phi(x^{(k)}) + x^{(k)}} \quad k \geq 0$$

The derived function $\phi_\Delta(x)$ has the same α as $\phi(x)$, but converges faster:

- Linear $\phi \to$ Quadratic ϕ_Δ
- $p \geq 2$ $\phi \to 2p - 1$ ϕ_Δ
- Linearly with $m \geq 2$ $\phi \to$ Linearly with $L = 1 - \frac{1}{m}\phi_\Delta$

H may converge even if normal FPI diverges.

2.11 Rope method

Obtained by modifying the Newton method, replacing $f'(x)$ with a fixed q

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)}}{q} \quad k = 0, 1, ... \tag{2.23}$$

e.g. $q = \frac{f(b)-f(a)}{b-a}$ in $[a, b]$.
Since it is a FPI with $\phi(x) = x - \frac{1}{q}f(x)$, we have convergence when

$$\phi'(x) = \left|1 - \frac{1}{q}f'(\alpha)\right| < 1 \tag{2.24}$$

Chapter 3

Interpolation

3.1 Approximation

Approximating a set of data or a function in $[a, b]$ consists in finding a suitable function $\tilde{f}$ that represents them with enough accuracy.
We can use Taylor polynomials to approximate complex functions but require many computations and have unpredictable behaviors on the sides of the domain.
If X is a Banach space and $M \subseteq X$, $\tilde{f} \in M$ is the best approximation of a function $f \in X$ when

$$\left\| f - \tilde{f} \right\| = E(f) = \inf_{\tilde{f} \in M} \left\| f - \tilde{f} \right\| \tag{3.1}$$

- $\tilde{f} \in M$ is the best approximation of the f in M.
- If M is a finite-dimensional subspace of X, then $\exists\ \tilde{f}$ B.A. of f in M (existence theorem).
- If X is strictly convex (any x, y on the unit sphere ∂B are joined by a segment that touches ∂B only in x, y), than $\tilde{f}$ is unique (uniqueness theorem).

3.2 Interpolation

Given $n + 1$ points $\{q_i = (x_i, y_i)\}_{i=0}^{n}$ on an interval, we want to find the function φ s.t. $\varphi(x_i) = y_i \forall i$.
We call this function φ **interpolant**, and the point nodes. We say that φ interpolates y_i in nodes q_i.
Interpolation is a form of approximation that could be used both to simplify a complex function in order to make it easier to derive or to understand data distributions. The iterpolants can be polynomial, trigonometric, rational, ecc.

3.3 Lagrange interpolation (φ is polynomial)

Given $n+1$ couples $\{x_i, y_i\}$, $i = 0, ..., n$ with x_i as nodes, we want to find a polynomial of degree $\leq n$ ($\pi_n \in \mathbb{P}^n$) s.t.

$$\pi_n(x_i) = y_i \quad \forall i \tag{3.2}$$

If y_i represent the values of a continuous f, π_n is the interpolant of f, denoted $\pi_n f$
In this setting $\exists!\ \pi_n \in \mathbb{P}^n$ s.t. $\pi_n(x_i) = y_i\ \forall i$
In order to obtain an expression for π_n, we study a special case in which $y_i = 0\ \forall j$ except when $y_{j=k} = 1$.

$$p_k \in \mathbb{P}_n, \quad p_k(x_j) = \underbrace{\delta_{jk}}_{\text{Kronecker symbol}} = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

φ_k is called **Lagrange basis** since it is a basis for $\mathbb{P}^n$. It has the following expression (also called Lagrange characteristic polynomials)

$$\varphi_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \frac{x - x_j}{x_k - x_j} \tag{3.4}$$

We define the Lagrange interpolant of f on nodes $x_0, ..., x_n$ the following linear combination of degree n

$$\mathcal{L}^n f = \sum_{k=0}^{n} f(x_k)\varphi_k(x) \tag{3.5}$$

The Lagrange basis is especially fit for good approximation since other polynomial sets may be ill-conditioned for the approximation task.
The example P^n generate the **Vandermonde matrix**, which is ill conditioned since for n large, rightmost columns will be very similar, making the matrix hardly invertible

$$B = (B_{ij}) = \begin{bmatrix} 1 & q_0 & q_0^2 & ... & q_0^n \\ 1 & q_1 & q_1^2 & ... & q_1^n \\ ... & ... & ... & ... & ... \\ 1 & q_n & q_n^2 & ... & q_n^n \end{bmatrix} = (q_i^s) \tag{3.6}$$

3.4 Interpolation error

$\{x_i,\ i = 0, ..., n\}$ are $(n+1)$ nodes on a bounded interval I. Given $f \in C^{n+1}(I)$. Then, $\forall x \in I \quad \exists \xi_x \in I$ s.t.

$$E_n f(x) = f(x) - \mathcal{L}^n f(x) = \frac{f^{(n+1)}(\xi_x) w(x)}{(n+1)!}, \quad \text{with } w(x) \prod_{i=0}^{n} x - x_i \tag{3.7}$$

Since $\|\cdot\|_\infty$ represent the highest value (sup) of a function, we can bound the error as

$$\|f(x) - \mathcal{L}^n f(x)\|_\infty \leq \frac{\left\|f^{(n+1)}(\xi)\right\|_\infty \|w(x)\|_\infty}{(n+1)!} \tag{3.8}$$

If f is analytically extendable in an oval $O(a,b,R)$ with $R>0$

$$\Rightarrow \left\| f^{(n+1)} \right\|_{\infty,\overline{O(a,b,R)}} \leq \frac{(n+1)!}{R^{n+1}} \|f\|_{\infty,\overline{O(a,b,R)}} \tag{3.9}$$

Thus, we can control the $(n+1)$ derivative directly with f.
Also

$$\|f - \mathcal{L}^n f\|_{\infty,[a,b]} \leq \frac{(n+1)!}{R^{n+1}} \|f\|_{\infty,O(a,b,R)} \left(\frac{|b-a|}{R}\right)^{n+1} \tag{3.10}$$

Then increasing the degree n of the interpolator does not guarantee a better approximation of n. Indeed, we may have that

$$\lim_{n\to\infty} \|f - \mathcal{L}^n f\|_\infty = \infty \tag{3.11}$$

3.5 Runge counterexample

$f(x) = \frac{1}{(1+x^2)}$ is interpolated at equispaced nodes in $I=[-s,s]$. The error$\|f - \mathcal{L}^n f\|_\infty$ tends to infinity when $n \to \infty$.
The presence of severe oscillations of $\mathcal{L}f$ w.r.t. f, especially near the endpoints, indicates lack of convergence. This is also called **Runge's phenomenon**.

3.6 Stability of interpolation

We want to estimate the impact of perturbed values $\hat{f}(x)$ on the interpolator $\mathcal{L}^n f$. We have that

$$\left\| \mathcal{L}^n f - \mathcal{L}^n \hat{f} \right\|_\infty = \Lambda_n(\{x_i\}_{i=0}^n) \left\| f - \hat{f} \right\| \tag{3.12}$$

where

$$\Lambda_n(\{x_i\}_{i=0}^n) = \|\sigma_{i=0}^n |\varphi_i(x)|\|_\infty \tag{3.13}$$

is the **Labesgue's constant** depending on interpolation nodes.
From the fisrt formula, we have that small variations in f yield small changes in $\mathcal{L}^n f$ if Λ is small. Therefore, Λ can be regarded as a condition number for interpolation.
It can be proved that $\|\mathcal{L}^n f\| \leq \Lambda_n(x)$.
For Lagrange interpolation at equispaced nodes

$$\Lambda_n(x) \simeq \frac{2^{n+1}}{e \cdot n(\log n + \gamma)}, \quad \text{with } e \approx 2.718 \text{ and } \gamma \approx 0.547 \tag{3.14}$$

For large values of n, this becomes unstable.

3.7 Distance from B.A.

If $p = \inf_{x \in \mathbb{P}^n} \|f - x\|_\infty$ with $\{x_i\}_{i=0}^n$ $(n+1)$ nodes, then

$$\begin{aligned}\|f - \mathcal{L}^n f\| =& \|f - p - \mathcal{L}^n(f-p)\| \\ \leq& \|f-p\|_\infty + \|\mathcal{L}^n(f-p)\|_\infty \\ \leq& \|I - \mathcal{L}^n\|_{\mathcal{L}} \|f-p\|_\infty \\ \leq& (\|I\|_{\mathcal{L}} - \|\mathcal{L}^n\|_{\mathcal{L}}) \|f-p\|_\infty \\ \leq& (1 + \|\mathcal{L}^n\|_{\mathcal{L}}) \|f-p\|_\infty\end{aligned}$$

where I is an operator for $f - p$ and $\|\mathcal{L}^n\|_{\mathcal{L}} = \sup_{n \neq 0} \frac{\|\mathcal{L}^n n\|_\infty}{\|n\|_\infty}$.
p is the best approximation of f on nodes $\{x_i\}_{i=0}^n$.

3.8 Chebyshev nodes

In order to minimize Λ and thus avoid Runge's phenomenon, we can use Chebyshev nodes on interval $[a, b]$ defined as

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} \hat{x}_i \tag{3.15}$$

where $\hat{x}_i = -\cos\left(\frac{\pi i}{n}\right),\ i = 0, ..., n$.
The nodes are equispaced on the semicirconference of diameter $[a, b]$ and are clustered towards the endpoints of the interval.
For Chebyshev nodes if f is continuously differentiabe in $[a, b]$, $\mathcal{L}^n f$ converges to f as $n \to \infty$ $\forall x \in [a, b]$

3.9 Erdos theorem

$\forall X$, where X is an infinite triangular matrix of interpolation points

$$X = \begin{bmatrix} 0 & 0 & 0 & ... & 0 \\ x_0 & 0 & 0 & ... & 0 \\ x_0 & x_1 & 0 & ... & 0 \\ ... & ... & ... & ... & ... \\ x_0 & x_1 & x_2 & ... & x_n \end{bmatrix} \tag{3.16}$$

we have

$$\Lambda_n(X) \geq \frac{2}{\pi} \log(n+1) - c \tag{3.17}$$

For Chebyshev nodes we have $\Lambda_n(X) \leq \frac{2}{\pi \log(n+1)+1}$.
For equispaced nodes we have $\Lambda_n(X) \leq \frac{2^{n+1}}{c(n \log n)}$.

3.10 Faber theorem

$$\forall x\ \exists f \in C^0([a, b])\ s.t. \|f - \mathcal{L}^n f\|_\infty \nrightarrow 0 \tag{3.18}$$

The Faber theorem proves that even on Chebyshev nodes not all continuous function will converge when used for interpolation.

3.11 Weierstrass approximation theorem

Suppose $f \in C^0([a,b])$. Then,

$$\forall \epsilon > 0 \ \exists p \in \mathbb{P}^n \ s.t. \ \|f - p\| < \epsilon, \ \forall x \in [a,b] \tag{3.19}$$

It shows that polynomial functions are dense in $C^0([a,b])$ and each polynomial can be uniformly approximated by one with rational coefficients.

3.12 Bernstein coefficients

The $n+1$ Bernstein basis for $\mathbb{P}^n$ are defined as

$$b_{n,i}(x) = \binom{n}{i} x^i (1-x)^{n-i}, \ i = 0, ..., n \tag{3.20}$$

$b_{n,i}$ is the $i-th$ polynomial in the Bernstein basis of degree n.
A linear combination of $b_{n,i}$ is called a Bernestein polynomial of degree n based on function f

$$B_n f(x) = \sum_{k=0}^{n} f(\frac{k}{n}) b_{n,k}(x) \tag{3.21}$$

$\forall x$ this is a weighted average of the $n+1$ values $f(\frac{k}{n})$ called bernstein coefficients.
Properties:

- $\sum_{i=0}^{n} b_{i,n} = (1 - x + x)^n = 1^n = 1 \quad \forall x \in [0,1]$
- $b_{i,n} \geq 0 \quad \forall x \in [0,1]$
- B_n is a linear positive operator: $B_n f \geq 0$ if $f \geq 0$
- $B_n f\left(\frac{i}{n}\right) \neq f\left(\frac{i}{n}\right)$ and if $f \in C^0([a,b])$ we have that $B_n f(x) \to f(x)$ as $n \to \infty$

The convergence is not pointwise as in interpolation, but uniform

$$\lim_{n \to \infty} \|f(x) - B_n f(x)\|_\infty = 0 \quad \text{with} \quad 0 \leq x \leq 1 \tag{3.22}$$

3.13 Qualitative proof of Weierstrass theorem

$\forall f \in C(I)$ and $\forall x_0 \in I$, we can find a quadratic function q s.t. $q > f \ \forall x$, but $q(x_0)$ is close to $f(x_0)$. The same can be done with $q < f$.

$$q = f(x) \pm \left(\frac{\epsilon}{2} + \frac{\|f\|_\infty}{2\sigma}(x - x_0)^2\right)$$

with

$$|x_1 - x_2| \leq \sigma \to |f(x_1) - f(x_2)| \leq \frac{\epsilon}{2}$$

$$q = ax^2 + bx + c$$

$$M = \max_{x_0 \in [a,b]} (a(x_0), b(x_0), c(x_0))$$

M depends exclusively on $\|f\|$, ϵ and σ but not on x_0.
By choosing a large n we have $\|f_i - B_n\| \leq \frac{\epsilon}{M}$. Using the triangle inequality we get $\|q - B_n q\| \leq 3\epsilon$. We have then

$$B_n f(x_0) \leq B_n q(x_0) \leq q(x_0) + 3\epsilon = f(x_0) + 4\epsilon \tag{3.23}$$

Same can be done below, having that

$$\forall x_0 \ f(x_0) - \epsilon \leq B_n f(x_0) \leq f(x_0) + \epsilon \tag{3.24}$$

So

$$\rightarrow \|B_n f - f\|_\infty \leq \epsilon \tag{3.25}$$

3.14 More on interpolation

- We can build a piecewise linear interpolant of f to avoid Burge effect when the number of nodes increases. f is a piecewise line or continuous function also called **finite element interpolant**.
- We can perform interpolation bu cubic splines, which are piecewise cubic function $f \in C^2$
- While the Minmax approximation we used so far is based on $\|\cdot\|_\infty$, the least squares approximation uses the Euclidean norm $\|\cdot\|_2$ to minimize $MSE = \sum_{i=0}^{n}(y_i - \tilde{f}(x_i))^2$
- Piecewise linear and splines are well suited to approximate data and functions in several dimensions.
- Trigonometric interpolation is well suited to approximate periodic functions. $\tilde{f}$ is a linear combination of sin and cos functions. FFT and IFFT allow for efficient computation of Fourier coefficients for a trigonometric interpolant from node values.

Chapter 4

Best Approximation in Hilbert spaces

$\mathcal{L}^2$ is an Hilbert space where the norm induced by the scalar product between vectors is $\|x\|_{\mathcal{L}^2} = (x_1^2 + x_2^2 + ... + x_n^2)^{\frac{1}{2}} = \sqrt{(x,x)}$

$$(a,b),\ a,b \in \mathcal{L}^2([0,1]) = \int_0^1 a \cdot b,\ \|a\| = \sqrt{\int_0^1 a^2} \tag{4.1}$$

4.1 Best approximation theorem in $\mathcal{L}^2$

Given a function $f \in \mathcal{L}^2(\mathbb{P}^n)$, p is B.A. of f in $\mathbb{P}^n$ iff

$$(f-p,q) = 0 \quad \forall q \in \mathbb{P}^n,\ \forall f \in \mathcal{L}^2(\mathbb{P}^n) \tag{4.2}$$

(recall: $\|p-f\| \leq \|q-f\|\ \forall q \in \mathbb{P}^n$ if p is B.A. of f w.r.t. chosen norm.)

Proof. Knowing that p is B.A.

$$\|q-f\|^2 = \|q-p+p-f\|^2 = \|q-p\|^2 + \|p-f\|^2 + \underbrace{2(q-p,p-f)}_{0 \text{ since } p-q=q\in\mathbb{P}^k} \text{ and } (0,n) = 0$$

$$\Rightarrow \|p-f\|^2 \leq \|q-f\|^2 \forall q \in \mathbb{P}^n$$

□

Or alternatively, we can

Proof. Knowing that $(f-p,q) = 0 \Rightarrow \|p-f\|^2 \leq \|p-f+tq\|$ with $t \geq 0$ perturbation,

$q \in \mathbb{P}^n$

$$\begin{aligned}\left\| \underbrace{p - f + \frac{tg}{2}}_{A} \underbrace{- \frac{tg}{2}}_{-B} \right\|^2 \leq & \left\| \underbrace{p - f + \frac{tg}{2}}_{A} + \underbrace{\frac{tg}{2}}_{+B} \right\|^2 \\ 0 \leq & 4\Big(p - f + \frac{tg}{2}, \frac{tg}{2}\Big) \\ 0 \leq & t^2\|q\|^2 + 2t(p - f, q) \\ \Rightarrow (p - f, q) \geq & -\frac{t}{2}\|q\|^2\end{aligned}$$

By choosing $-q$ instead, we get $(p - f, q) \leq \frac{t}{2}\|q\|^2$.
Thus, it is valid $\forall t, \forall q$ that

$$-\frac{t}{2}\|q\|^2 \leq (p - f, q) \leq \frac{t}{2}\|q\|^2 \tag{4.3}$$

which implies that $(p - f, q) = 0$ since a t can be chosen to bound it on both sides.
Since $(p - f, q) = 0 \ \forall q \Leftrightarrow (p - f, v_i) = 0 \ \forall i = 0, 1, ..., n$ with $\mathbb{P} = span\{v_i\}$

$$\Rightarrow (p, v_i) = (f, v_i) \Rightarrow (\sum_{j=0}^{n} p_j v_j, v_i) = (f, v_i) \tag{4.4}$$

□

Computing integrals is easier than performing interpolation, and it yields better results.

4.2 Matrix formulation

We can rewrite $(\sum p_j v_j, v_i) = (f, v_i)$ as a matricial relation

$$Mp = F \text{ where } M_{ij} = (v_J, v_i) = \int_0^1 v_j v_i \text{ and } F_i = (f, v_i) = \int_0^1 f v_i \tag{4.5}$$

If we set $v_i = x^{(i)}$, we obtain the Hilbert matrix

$$M_{ij} = \int_0^1 x^{(J)} x^{(i)} = \frac{1}{i + j + 1} \tag{4.6}$$

The conditional number of the Hilbert matrix is

$$K(M) = O\Big(\frac{(1 + \sqrt{2})}{\sqrt{n}}\Big)^{4n} \tag{4.7}$$

When n increases K explodes, which is very bad. M is difficult to invert and very ill-conditioned because of collinear lines. We would like $M_{ij} = I$, so we use the Legendre basis function to make it orthonormal w.r.t. $\mathcal{L}^2$.

We want $v_i \in \mathcal{P}^n$ s.t. $M_{ij} = (v_i, v_j) = \delta_{ij}$. To build it, we use the Graham-Schmidt method

$$\begin{cases} v_0 = 1, \quad f \ s.t. \ \int_0^1 f = 1 \\ f^{i+1} = x^{i+1} - \sum_{j=0}(x^{(i+1)}, v_J)v_J \\ v_{i+1} = \frac{f^{i+1}}{\|f^{i+1}\|} \end{cases} \tag{4.8}$$

The first line is set of additive basis having unity as first element. This ensures orthogonality between basis function.
As i (the degree) increases, $v_{i+1} = \frac{f^{i+1}}{\|f^{i+1}\|} \to \infty$ since $x^{i+1} \to \infty$
We can avoid instability by using $v_{i+1} = \frac{f_{i+1}}{f_{i+1}(0)}$ instead.
The points created with Graham-Schmidt represent the Legendre Basis. They make p (best approximation) easy to compute, since M becomes easy to invert and we have a diagonal matrix formed by orthogonal basis

$$p = M^{-1}F \tag{4.9}$$

Chapter 5

Integration

Integration is an operation $f[a,b] \to \mathbb{R}$, defined as

$$I(f) = \int_a^b f(x)dx \tag{5.1}$$

Integration is very expensive from a numeric point of view if f is complicated. Our purpose is to make it simpler, given $f \in C^0([a,b])$.
Many possible approaches called quadratures

- Midpoint formula (degree 1)

$$I_{mp}(f) = (b-a)f\Big(\frac{a+b}{2}\Big) \tag{5.2}$$

- Trapezoidal formula (degree 1)

$$I_t(f) = \frac{(b-a)}{2}(f(a)+f(b)) \tag{5.3}$$

- Simpson formula (degree 3), using $\mathcal{L}^2 f$

$$I_s(f) = \frac{(b-a)}{6}\Big(f(a) + 4f\Big(\frac{a+b}{2}\Big) + f(b)\Big) \tag{5.4}$$

and their composite variations, using M intervals:

- $I^c_{mp} = H\sum_{k=1}^{M} f(\overline{x}_k)$
- $I^c_t(f) = \frac{H}{2}\sum_{k=1}^{M-1} f(x_k) + \frac{H}{2}(f(a)+f(b))$
- $I^c_s(f) = \frac{H}{6}\sum_{k=1}^{M}(f(x_{k-1}) + 4f(\overline{x}_k) + f(x_k))$
  with $\overline{x}_k = \frac{(x_{k-1}+x_k}{2}$ and $H = \frac{(b-a)}{M}$

Those are all specific cases of a more general quadrature formula

$$I_n(f) = \sum_{i=0}^{n} \alpha_i f(y_i) \tag{5.5}$$

- $\{y_i\}$ are quadrature nodes.
- α_i are the quadrature weights.

We can use $\mathcal{L}^n f \in \mathbb{P}^n$ at nodes y_i as approximation function, to get the interpolatory quadrature formula

$$f_n(x) = \mathcal{L}^n f(x) : I_n(f) = \int_a^b f_n(x)dx = \int_a^b \sum_{i=0}^n \varphi_i(x) f(y_i)dx$$
$$= \sum_{i=0}^n f(y_i) \int_a^b \varphi_i(x)dx \Rightarrow \sum_{i=0}^n \alpha_i f(y_i)$$

with α_i being $\int_a^b \varphi_i(x)dx$.
The degree of accuracy/exactness of a quadrature is the integer r s.t. quadrature using $\mathbb{P}^n$ doesn't produce errors on $I(f)$

$$\max_{M \in \mathbb{N}} I_n(f) = \int f \ \forall f \in \mathbb{P}^r \tag{5.6}$$

Midpoint rule: for f linear function $\in [a, b]$, we can choose y_i as $\alpha_i = \frac{1}{2}b + \frac{1}{2}a$ to cancel out positive and negative approximation. So we approximate exactly $f \in \mathbb{P}^1$ with a constant function ($\mathbb{P}^0$). We have that

$$\left| \int_a^b f(x)dx - f\Big(\frac{b-a}{2}\Big) \right| \leq \frac{\|f''\|_\infty}{3} \Big(\frac{b-a}{2}\Big)^3 = \frac{\|f''\|_\infty}{24} \quad \text{for } [a, b] = [0, 1] \tag{5.7}$$

Since we can see the error depends on the size of the interval, we usually prefer composite quadrature, pasting together intervals through continuity conditions to keep them small.

5.1 Legendre polynomials and max accuracy

Given $m \in \mathbb{N} > 0$, a quadrature formula $\sum_{i=0}^n \overline{\alpha}_i f(\overline{y}_i)$ has degree of accuracy $n + m$ iff it makes use of interpolation and the nodal polynomial $w_{n+1} = \prod_{i=0}^n (x - \overline{y}_i)$ associated to nodes $\{\overline{y}_i\}$ is s.t.

$$\int_a^b w_{n+1}(x)p(x)dx = 0 \quad \forall p \in \mathbb{P}_{m-1} \tag{5.8}$$

The maximum value for m is $n + 1$, achieved when w_{n+1} is proportional to $L_{n+1}(x)$, the Legendre polynomial of degree $n + 1$. Legendre polynomials can be computed recursively as

$$L_0(x) = 1, \quad L_1(x) = x, \quad L_{k+1}(x) = \frac{2k+1}{k+1} x L_k(x) - \frac{k}{k+1} L_{k-1}(x) \tag{5.9}$$

Since L_{n+1} is orthogonal to $\forall L_{\{0,1,...,n\}}$ ($\int_a^b L_{n+1}(x)L_j(x)dx = 0 \ \forall j < n + 1$), we can see why m is bounded at $n + 1$. Thus, the highest degree of accuracy is $2n + 1$, obtained using the **Gauss-Legendre formula**

$$I_{GL} = \begin{cases} \overline{y}_i = \text{roots of } L_{n+1}(x) \\ \overline{\alpha}_i = \frac{2}{(1-y_i^2)(L'_{n+1}(y_i))^2} \end{cases} \qquad i = 0, ..., n \tag{5.10}$$

The related Gauss-Legendre-Lobatto formula includes interval bounds among quadrature points, and has a D.O.A. of $2n-1$.
The interval used for I_{GL} is $\{-1,1\}$, thus the $\overline{y}_i, \overline{\alpha}_i$ yo reconvert to original values for (a,b), use Chebyshev formula.

$$y_i = \frac{a+b}{2} + \frac{b-a}{2}\overline{y}_i, \quad \alpha_i = \frac{b-a}{2}\overline{\alpha}_i \tag{5.11}$$

Proof. Knowing $f \in \mathbb{P}^{n+m}$, we apply quotient theorem for $\mathbb{P}$

$$f(x) = \underbrace{w_{n+1}(x)}_{\in\mathbb{P}^{n+1}}\underbrace{p(x)}_{\in\mathbb{P}^{m+1}} + \underbrace{q(x)}_{\in\mathbb{P}^n}$$
$$\int_a^b f(x) = \underbrace{\int_a^b w_{n+1}(x)p(x)dx}_{(*)} + \int_a^b q(x)dx$$

Assuming that $(*)=0$, we get $\int_a^b f(x) = \int_a^b q(x) = I_n(q)$ (quadrature for $q \in \mathbb{P}^n$ is exact since we took $n+1$ nodes).
Knowing $(*)=0$, we want to prove that if D.O.A. is $n+m$, then $(*)=0$

$$I_n(f) = \int f \ \forall f \in \mathbb{P}^{n+m} \to \int_a^b \underbrace{w_{n+1}(x)p(x)}_{\in\mathbb{P}^{n+m}}dx = I_n(w_{n+1}(x)p(x)) \tag{5.12}$$

Since $I_n(w_{n+1}(x)p(x)) = 0$ because $w_{n+1}(y_i) = 0 \ \forall i$, we proved it.
To prove that m is bound at $n+1$, we could replace $p \in \mathbb{P}^{m-1}$, with $w_{n+1}(x)$, obtaining

$$\int_a^b w_{n+1}(x)w_{n+1}(x)dx = 0 \quad \text{for } m \geq n+2$$
$$\Rightarrow w_{n+1}(x) = 0$$

Which is false, because based on false assumption. □

5.2 Peano integration kernel theorem

The **Peano kernel** represents the error we make when integrating a function $g(x) = (x-\theta)_+^k$ for a given θ.

$$K(\theta) = E_x((x-\theta)_+^k) = \int_a^b (x-\theta)_+^k dx - I_n((x-\theta)_+^k) \tag{5.13}$$

with

$$(x-\theta)_+^k = \begin{cases} (x-\theta)^k & \text{for } x > 0 \\ 0 & \text{for } x < 0 \end{cases} \tag{5.14}$$

Since

$$\int_a^b (x-\theta)_+^k dx = \frac{(x-\theta)^{k+1}}{k+1}\Big|_{x=b} - \underbrace{\frac{(x-\theta)^{k+1}}{k+1}\Big|_{x=a}}_{0 \text{ since } a \leq \theta} \tag{5.15}$$

we have that it doesn't depend on a.
The Peano kernel theorem says that given a quadrature formula of degree α and $f \in C^{k+1}([a,b])$, with $0 \le k \le \alpha$ then

$$|E(f)| \le \frac{1}{k!}\|k\|_2 \left\| f^{(k+1)} \right\|_2 \tag{5.16}$$

(where other norms combination can be $1-\infty$ and $\infty - 1$)

Proof.

$$f(x) = \underbrace{\sum_{i=0}^{k} \frac{f^i(a)}{i!}(x-a)^i}_{\substack{p(x) \text{ Taylor exp. of } f \\ \text{of order } k \text{ around } a}} + \underbrace{\frac{1}{k!}\int_a^b f^{(k+1)(\theta} K(\theta) d\theta}_{r(x) \text{ from P.K.T.}} = p(x) + r(x)$$

$$E(f) = \int f - I_n(f) = \underbrace{\int p - I_n(p)}_{\substack{\text{cancel out because} \\ p \in \mathbb{P}^d, d<k}} + \int r - I_n(r) = \int r - I_n(r)$$

$$\int_a^b r = \int_a^b \frac{1}{k!}\Big(\int_a^b f^{(k+1)}(\theta)(x-\theta)_+^k d\theta\Big)dx = \int_a^b f^{(k+1)}(\theta)\Big(\int_a^b \frac{(x-\theta)_+^k}{k!}dx\Big)d\theta$$

$$I_n(r) = \int_a^b I(f^{(k+1)}(\theta)\frac{(x-\theta)_+^k}{k!}d\theta = \int_a^b f^{(k+1)}(\theta) I\Big(\frac{(x-\theta)_+^k}{k!}\Big)d\theta$$

$$\Rightarrow E(f) = \int_a^b f^{(k+1)}(\theta) E_x((x-\theta)_+^k) \cdot \frac{1}{k!} = \frac{1}{k!}\int_a^b f^{(k+1)}(\theta)K(\theta)d\theta$$

□

5.3 More on numerical integration

- Simpson adaptive formula uses different steplenghts to compute the composite interpolant on the integral reducing the nodes needed.
- Monte Carlo methods approximate the integral of f as a function statistical mean. They usually lead to poor results.

Chapter 6

Linear Systems

A linear system of order n, $n > 0$, is constituted by a given matrix $A_{n\times n} = (a_{ij})$, a given vector $b = (b_j)$ and an unknown vector $x = (x_j)$ that should be found by solving the system.

$$Ax = b \quad \Rightarrow \sum_{j=1}^{n} a_{ij}x_j = b_i, \quad i = 0, ..., n \tag{6.1}$$

The solution exists and is unique iff A is non-singular ($\det(A) \neq 0$) for any vector b.
In principle, we can compute the solution using the Cramer rule, where A_i is the matrix obtained by replacing the $i - th$ column of A by b, by applying **Laplace extension**

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad i = 1, ..., n \tag{6.2}$$

However, this is computationally infeasible since it requires $\approx 3(n + 1)!$ operations. We can reduce the computational cost by applying a method from one of the approaches:

- Direct methods: yield system solution in finite steps.
- Iterative methods: require a (theoretically) infinity of steps.

A full matrix linear system cannot be solved in principle under n^2 operations, one for each element of the matrix.

6.1 Direct methods

Let's define

$$U = (u_{ij}) \Rightarrow u_{ij} = 0 \ \forall i, j \ s.t. \ 1 \leq j < i \leq n, \ U \text{ is upper triangular}$$
$$L = (l_{ij}) \Rightarrow l_{ij} = 0 \ \forall i, j \ s.t. \ 1 \leq i < j \leq n, \ L \text{ is lower triangular}$$

If A is non-singular and triangular, we have that

$$\det(A) = \prod_{i=1}^{n} \lambda_i(A) = \prod_{i=1}^{n} a_{ii} \quad \Rightarrow a_{ii} \neq 0 \ \forall i \tag{6.3}$$

6.1.1 LU factorisation

Let $A \in \mathbb{R}^{n\times n}$, and L, U respectively lower and upper triangular s.t.

$$A = LU \quad \text{LU decomposition/factorisation of } A \tag{6.4}$$

Instead of solving a full linear system, e can solve two triangular systems

$$Ax = b \;\rightarrow\; \begin{cases} Ly = b \\ Ux = y \end{cases} \tag{6.5}$$

Since the two systems are triangular, they can be solved applying respectively a forward substitutions algorithm to get x from U.
Both require n^2 operations to complete.

FORWARD	BACKWARD
$y_1 = \frac{1}{l11} b_1$	$x_1 = \frac{1}{u_{nn}} y_n$
$y_i = \frac{1}{l_{ii}}(b_i - \sum_{j=1}^{i-1} l_{ij} y_j),\ \forall i = 2, ..., n$	$x_i = \frac{1}{u_{ii}}(y_i - \sum_{j=i+1}^{n} u_{ij} x_j),\ \forall i = n-1, ..., 1$

Finding the matrices L, U required for this task takes around $\frac{2n^3}{3}$ operations, and is done as follows

1. The elements of L and U satisfy the nonlinear system

$$\sum_{r=1}^{\min(i,j)} l_{ir} u_{rj} = a_{ij}, \quad i, j = 1, ..., n \tag{6.6}$$

2. The system is undetermined having n^2 equations and $n^2 + n$ unknowns. Consequently, LU factorization is not unique.
3. By forcing $l_{ii} = 1$ (all diagonal elements of $L = 1$), we eliminate n unknowns, obtaining a determined system that can be solved using Gauss elimination method.

6.1.2 Gauss elimination method (GEM)

The GEM transforms a system $Ax = b$ with $A \in \mathbb{R}^{n\times n}$ in a equivalent system $Ux = \hat{b}$, where U is an upper triangular matrix, and $\hat{b}$ is a properly transformed b which can be solved by backward substitution.
To perform the transformation, we exploit the fact that adding to an equation a linear combination of other equations will not change the solution.

$$l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \tag{6.7}$$

We want to find coefficient that will yield 0 when combined.

$$\rightarrow a_{ij}^{(k+1)} = a_{ij}^{(k)} - \underbrace{l_{ik}}_{\substack{\text{Will set to 0 all} \\ a_{ij} \text{ below pivot}}} a_{kj}^{(k)}$$

$$\rightarrow b_i^{(k+1)} = \underbrace{b_i^{(k)}}_{\substack{\text{Adopt } b \text{ to} \\ \text{changes accordingly}}} - l_{ik} b_k^{(k)}$$

$$\forall k = 1, ..., n-1 \quad \forall i = k+1, ..., n \quad \forall j = k+1, ..., n$$

The elements on the main diagonal ($a_{kk}^{(k)}$) are called **pivots** and have to be non-zero. Updating a takes $2(n-k)^2$ operations, updating b takes $2(n-k)$ operation and l takes $(n-k)$. The total is $\frac{3n^2-n}{2}$, plus n^2 to solve $Ux = \hat{b} \approx \frac{2}{3}n^3$ operations.
Gauss method is equivalent to LU factorization, but the latter proves to be very effective when we are trying to solve many systems having different b's but same A (reducing operations from $\frac{2}{3}Mn^3$ of Gauss to the simple solving of LU, $2Mn^2$, where M is the number of systems).
Same matrices to which GEM can be applied (pivots $\neq 0$):

- (Strictly) diagonal dominant by row: $|a_{ii}| \geq \sum_{\substack{j=1,..,n \\ j \neq i}} |a_{ij}|$ with $i = 1, ..., n$
- (Strictly) Diagonally dominant by column: $\left| a_{jj} \geq \sum_{i=1,...,n i \neq j} \right|$ with $j = 1, ..., n$
- Symmetric positive definite: $\lambda_i(A) > 0$ with $i = 0, ..., n$

These matrices have all in common that all their principal submatrices A_i of order $i = 1, ..., n-1$ are non-singular.

- If $A \in \mathbb{R}^{n \times n}$ is symmetric positive definite, $\exists! R$ s.t.

$$A = R^T R \tag{6.8}$$

- This procedure is known as **Cholesky factorization** and requires about $\frac{n^3}{3}$ operations (against $\frac{2n^3}{3}$ of LU)
- Since L, U matrices are triangular, $l_{ii} = 1$, we can calculate the determinant of $A = LU$ as $O(n^3)$

$$\det(A) = \det(L)\det(U) = 1 \cdot \det U = \prod_{k=1}^{n} u_{kk} \tag{6.9}$$

- We can model matrix inversion of A as a linear system where $x^{(k)}$ corresponds to the $k-th$ column of A^{-1} and $i^{(k)}$ to the $k-th$ column of $I \in \mathbb{R}^{n \times n}$. We solve

$$Ax^{(k)} = i^{(k)} \tag{6.10}$$

obtaining the inverse matrix A^{-1} in $2n^3$ operations.

6.1.3 Memory-space limitations

A square matrix of order n is called **sparse** if the number of nonzero entries is of order n (on n^2 total entries).
The pattern is the $2D$ representation of nonzero entries positions.

- Lower band p_1: $a_{ij} = 0$ when $i > j + p_1$
- Upper band p_2: $a_{ij} = 0$ when $j > i + p_2$

The maximum between p_1, p_2 is called **matrix bandwidth**.
The **fill-in phenomenon** occurs when after an LU decomposition, L and U present less sparsity than the original A, leading to a bigger memory usage. To reduce the phenomenon, we can apply row and column permutations to reorder A before performing the factorisation (pivoting).

6.1.4 Pivoting

If a pivot in A becomes 0, GEM fails. To avoid that, we can reorder the rows (and columns) in a way that no pivot is zero. This technique is called pivoting

$$PA = LU \tag{6.11}$$

P is a permutation matrix initially set equal to I, changed accordingly to permutations made on A.
It is advised to perform pivoting at each step of LU factorisation to use always the pivot with maximum modulus int he $A^{(k)}$ submatrix, using both rows and columns permutation (P and Q).

$$PAQ = LU \tag{6.12}$$

That's **total pivoting** requires $\frac{2n^3}{3}$ operations.
Alternatively we can search the maximum modulus pivot in the same row or column of the current one: **partial pivoting** require n^2 operations.
By applying partial pivoting to LU factorization, we have to solve

$$Ax = b \;\Rightarrow PAx = Pb \;\Rightarrow \begin{cases} Ly = Pb \\ Ux = y \end{cases} \tag{6.13}$$

While for complete pivoting we have:

$$Ax = b \;\Rightarrow \underbrace{PAQ}_{LU}\underbrace{Q^{-1}x}_{x^*} = Pb \;\Rightarrow \begin{cases} Ly = Pb \\ Ux^* = y \end{cases} \quad \Rightarrow x = Qx^* \tag{6.14}$$

6.1.5 Precision of direct methods

Total pivoting is more stable than partial pivoting.
When a linear system is solved numerically, we are looking for the exact solution $\hat{x}$ of a perturbed system

$$(A + \delta A)\hat{x} = b + \delta b \tag{6.15}$$

where δA and δb depend on the method used to approximate the results.
We call conditioning of a matrix M (symmetric positive definite) the constant

$$K(M) = \frac{\lambda_{\max}(M)}{\lambda_{\min}(M)} \tag{6.16}$$

also called the spectral condition number of M. From the perturbed system formula, we get $x - \hat{x} = -A^{-1}\delta b$, and thus

$$\|x - \hat{x}\| = \left\|A^{-1}\delta b\right\| \tag{6.17}$$

We can set a bound for the relative error, given previous relations, as

$$\frac{\|x - \hat{x}\|}{\|x\|} \leq K(A)\frac{\|\delta b\|}{\|b\|} = K(A)\frac{\|r\|}{\|b\|} \tag{6.18}$$

with residual $r = b - A\hat{x}$.
The bigger the K, the worse the solution provided by a direct method. If $K \approx 1$, the matrix is well conditioned. r is an estimator of the error $x - \hat{x}$. If $K(A)$ is small, then error is small when $\|r\|$ is small. Vice versa, if $K(A)$ is large, we can't use r as measure for the error.

6.1.6 Other direct methods

- **Thomas algorithm** is used to perform an optimised LU-factorization of a triangular matrix in n operations.
- The solution of an overdetermined system $Ax = b$, $A \in \mathbb{R}^{m\times n}$ with $m > n$ can be computed using QR factorization or singular value decomposition.

6.2 Iterative methods

Solving $Ax = b$ iteratively implies building a series of vector $x^{(k)\in\mathbb{R}^n}$ s.t.

$$\lim_{k\to\infty} x^{(k)} = x \quad \forall x^{(0)} \in \mathbb{R} \tag{6.19}$$

This can be achieved through recursion, as

$$x^{(k+1)} = Bx^{(k)} + g, \ s.t. \ k \geq 0 \tag{6.20}$$

B well chosen depending on A, g vector satisfying $x = Bx + g$.
B is the **iteration matrix**, which helps defining error at step k as

$$e^{(k)} = x - x^{(k)} = B^k e^{(k-1)} \quad \text{with} \lim_{k\to\infty} e^{(k)} = 0 \ \forall e^{(0)} \tag{6.21}$$

The error goes to 0 iff $\rho(B) < 1 = \max|\lambda_1(B)|$. ρ is called the spectral radius of B, the max modulus of its eigenvalues.
$\rho(B) < 1$ is necessary for convergence. The smaller $\rho(B)$, the less iteration are needed to reduce $e^{(0)}$ under a threshold e. We require at least $k_{\min}$ iteration, where

$$\min(k_{\min}) \ s.t. \ \rho(B)^{k_{\min}} \leq \epsilon \tag{6.22}$$

6.2.1 Constructing an iterative method

We usually split A s.t. $A = P - (P - A)$ where P is an invertible matrix called **preconditioner** of A (preconditioning makes convergence faster and smoother)

$$\begin{aligned} &Ax = b \ \Rightarrow Px = (P_A)x + b \\ &\Rightarrow B = P^{-1}(P - A) = I - P^{-1}A \ \Rightarrow g = P^{-1}b \end{aligned}$$

We can thus define the **Richardson method** as:

$$P(x^{(k+1)} - x^{(k)}) = r^{(k)} = b - Ax^{(k)} \tag{6.23}$$

where $r^{(k)}$ is the residual at iteration k.
It can also be generalize by adding a parameter α_k before the $r^{(k)}$ which is used to improve the convergence of series $x^{(k)}$. This is equal to solve the linear system

$$Pz^{(k)} = r^{(k)}, \text{ with } x^{(k+1)} = x^{(k)} + \alpha_k z^{(k)} \quad \Big(P \underbrace{\Big(\frac{x^{(k+1)} - x^{(k)}}{\alpha_k}\Big)}_{z^{(k)}} = r^{(k)}\Big) \tag{6.24}$$

where $z^{(k)}$ is called the preconditioned residual of step k. P should be either diagonal triangular or tridiagonal to reduce the number of operations required to compute $z^{(k)}$.

6.2.2 Jacobi method

If, given $A \in \mathbb{R}^{n\times n} = (a_{ij})$, we have that $a_{ii} \neq 0 \forall i, 0 \leq i \leq n$, we can set

$$P = D = diag(a_{11}, a_{22}, ..., a_{nn}) \ \text{ and } \ \alpha_k = 1 \ \forall k \tag{6.25}$$

It can be written under the form $x^{(k+1)} = Bx^{(k)} + g$, with

$$B_J = D^{-1}(D - A) = I - D^{-1}A \tag{6.26}$$

If the matrix $A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant by row, than the Jacobi method always converges.

6.2.3 Gauss-Seidel method

In order to obtain a faster convergence, we can include the newly computed components of vector $x_j^{(k+1)}$, $j = 1, ..., i-1$ to the previous $x_j^{(k)}, j \geq i$, to compute $x_i^{(k+1)}$.
In this case the update is sequential rather than simultaneous, but leads to faster convergence. It corresponds to

$$P = D_E \text{ and } \alpha_k = 1, \quad \text{with } E \begin{cases} E_{ij} = -a_{ij} & \text{if } \ i > j \\ E_{ij} = 0 & \text{if } \ i \leq j \end{cases} \tag{6.27}$$

Then we have

$$B_{GS} = (DE)^{-1}(D - E - A) \tag{6.28}$$

If A is strictly diagonally dominant by row, Gauss-Seidel converges. If A is symmetric positive definite, then Gauss-Seidel converges.
If A is triangular whose diagonal are non null and invertible, then Jacobi and Gauss-Seibel are either both divergent or both convergent. If they converge, we have that $\rho(B_{GS}) = \rho(B_J)^2$

6.2.4 Richardson method

If $\alpha_k = \alpha \ \forall k$ the method is called stationary, else it is called dynamic.
If A and P are s.p.d., there are two optional criteria to choose α:

- Stationary case: $\alpha_k = \alpha_{opt} = \frac{2}{\lambda_{min}+\lambda_{max}},\ k \geq 0,\ \underset{max}{\lambda_{min}}$ eig. of $P^{-1}A$
  If $P = I$, we get the stationary Richardson method:

$$B = I - \alpha A,\ g = b \ \text{ with } \ \alpha = \frac{2}{\lambda_{min}(A) + \lambda_{max}(A)} \tag{6.29}$$

- Dynamic case: $\alpha_k = \frac{(z(k)^T r^{(k)}}{(z^{(k)})^T A z^{(k)}},\ k \geq 0$ where $z^{(k)=P^{-1}r^{(k)}}$ so
  $r^{(0)} = b - Ax \to r^{(k+1)} = r^{(k)} - \alpha_k A z^{(k)}$
  If $P = I$, we get the gradient method

$$B = I\alpha_k A,\ g = b \ \text{ with } \ \alpha_k = \frac{(r^{(k)})^T r^{(k)}}{(r^{(k)})^T A r^{(k)}},\ k \geq 0 \tag{6.30}$$

In both cases, the convergence is s.t.

$$\left\|x^{(k)} - x\right\|_A \leq \left(\frac{K(P^{-1}A) - 1}{K(P^{-1}) + 1}\right)^k \left\|x^{(0)} - x\right\|_A,\ k \geq 0 \ \text{ where } \ \underbrace{\|v\|_A}_{\substack{\text{energy norm} \\ \text{of } A}} = \sqrt{v^T A v} \ \text{ and } \ \underbrace{K(P^{-1}A)}_{\text{condition number}} \tag{6.31}$$

The gradient method converges faster, followed by GS and J. If a is a generic matrix, keeping low both K and the number of operations is hard.
If A is s.p.d. instead, we have that for gradient method the optimal α_k is

$$\alpha_k = \frac{(r^{(k),z^{(k)}})}{(Az^{(k)}, z^{(k)}}, \quad \text{with } z^{(k)} = P^{-1}r^{(k)} \tag{6.32}$$

and

$$K(P^{-1}A) = \frac{\lambda_{max}(P^{-1}A)}{\lambda_{min}(P^{-1}A)} \tag{6.33}$$

6.2.5 Conjugate gradient method

When A and P are both s.p.d., we can apply the conjugate gradient method, which converges even faster then the gradient (at most n steps) $\to$ direct method.
Given $\underline{x}^{(0)}$, $\underline{r}^{(0)} = Ax^{(0)} + b, \underline{z}^{(0)} = P^{-1}r^{(0)}, \underline{p}^{(0)} = z^{(0)}$, we have

$$\begin{aligned}
\alpha_k =& \frac{p^{(k)T} r^{(k)}}{p^{(k)} A p^{(k)}} \Rightarrow x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)} \\
\Rightarrow r^{(k+1)} =& r^{(k)} - \alpha_k A p^{(k)} \Rightarrow P z^{(k+1)} = r^{(k+1)} \\
\Rightarrow \beta_k =& \frac{(Ap^{(k)})^T z^{(k+1)}}{(Ap^{(k)})^T p^k} \Rightarrow p^{k+1} = z^{(k+1)} \beta_k p^{(k)}
\end{aligned}$$

The error estimate then becomes:

$$\left\|e^{(k)}\right\|_A = \left\|x^{(k)} - x\right\|_A \leq \frac{2e^k}{1 + e^{2k}} \left\|x^{(0)} - x\right\|_A, \quad \text{where } c = \frac{\sqrt{K(P^{-1}A) - 1}}{\sqrt{K(P^{-1}A) + 1}} \tag{6.34}$$

6.2.6 Convergence criteria

As for direct methods, we have that

$$\frac{\left\|x^{(k)} - x\right\|}{\|x\|} \leq K(A) \frac{\left\|r^(k)\right\|}{\|b\|} \tag{6.35}$$

or, if A is preconditioned

$$\frac{\left\|x^{(k)} - x\right\|}{\|x\|} \leq K(P^{-1}A) \frac{\left\|P^{-1}r^{(k)}\right\|}{\|P^{-1}b\|} \tag{6.36}$$

6.2.7 Stopping conditions

- $\left\|r^{(K_{min})}\right\| \leq \epsilon\|b\| \Rightarrow \frac{\left\|e^{(K_{min})}\right\|}{\|x\|} \leq \epsilon K(A)$, which is meaningful only if $K(A)$ is reasonably small.
- $\delta^{(k)} = x^{(k+1)} - x^{(k)} \Rightarrow \left\|\delta^{(K_{min})}\right\| \leq \epsilon$, which is better if $P(B) \gg 1$

6.2.8 Choosing the method

The choice of the method is particularly important for large A, and depends largely on context (A properties resources). Direct methods are usually more effective in absence of a good P, but more sensitive to ill-conditioning and require large storage.

Chapter 7

Least squares

Having $n+1$ points $x_0, ..., x_n$ and $n+1$ values, $y_0, ..., y_n$, the interpolating polynomial may show large oscillations for large values of n.
We can instead define a polynomial $\tilde{f}_m(x)$ of degree $m < n$ that approximates the data "at best":

$$\sum_{i=0}^{n} \left| y_i - \tilde{f}_m(x_i) \right|^2 \leq \sum_{i=0}^{n} |y_i - p_m(x_i)|^2 \quad \forall p_m(x) \in \mathbb{P} \tag{7.1}$$

If the values of y_i were those of a function f, then $\tilde{f}_m$ is called the least squares approximation of f.
We can determine the coefficients of $\tilde{f}_m$ as:

$$\frac{\partial \phi}{\partial a_k} = 0, \ k = 0, ..., m \quad \text{with } \tilde{f}_m = a_0 + a_1 x_i + ... a_m x_i^m \ \text{ and } \phi = \sum_{i=0}^{n} \left| y_i - \tilde{f}_m \right| \tag{7.2}$$

While $\tilde{f}_m$ is a polynomial, we can generalize the formula for functions of a space V_m obtained by linearly combining $m+1$ independent functions ($\{\psi_i, j = 0, 1, ..., m\}$).
The choice of ψ is dictated by the conjectured behaviour of the function underlying the current data distribution

$$\tilde{f}(x) = \sum_{j=0}^{m} a_j \psi_j(x) \xrightarrow[\substack{a \text{ can be} \\ \text{obtained} \\ \text{by solving}}]{} B^T B a = B^T y \tag{7.3}$$

where $B = b_{ij} = \psi_j(x_i)$, a are the unknown coefficients and y are the data.

Chapter 8

Eigenvalues and eigenvectors

Given $A \in \mathbb{C}^{n\times n}$, the eigenvalue problem consists in finding a scalar λ and a non-null vector x s.t.

$$Ax = \lambda x \tag{8.1}$$

Any such λ is called eigenvalue of A, while x is the associated eigenvector. All multiples $\alpha x, \alpha \neq 0$, are also eigenvectors of λ.
If x is known, we can recover λ using the Rayleigh quotient

$$\frac{\overline{x}^T Ax}{\|x\|^2} \tag{8.2}$$

The eigenvalues of A are the roots of the characteristic polynomial of A

$$p_A(\lambda) = \det(A - \lambda I) \tag{8.3}$$

A $n\times n$ matrix has exactly n eigenvalues (with or without multiplicity). A is diagonalizable if $\exists U \in \mathbb{C}^{n\times n}$ s.t. $\det(U) \neq 0$ and

$$U^{-1}AU = \Lambda = diag(\lambda_1, ..., \lambda_n) \tag{8.4}$$

The columns of U are the eigenvectors of A.
It A is diagonal or triangular, λ's are its diagonal entries. Otherwise, if A is a general large matrix, seeking the zeros of p_A is hard.

8.1 Power method

If $A \in \mathbb{R}^{n\times n}$ and its eigenvalues are ordered as

$$|\lambda_1| > |\lambda_2| > |\lambda_3| > ... > |\lambda_n| \tag{8.5}$$

then we can compute λ, and x, iteratively using the **power method**. Given an arbitrary $x^{(0)} \in \mathbb{C}^n$ and setting $y^{(0)} = \frac{x^{(0)}}{\|x^{(0)}\|}$, we can compute for $k = 1, 2, ...$

$$x^{(k)} = Ay^{(k-1)}, \quad y^{(k)} = \frac{x^{(k)}}{\|x^{(k)}\|}, \quad \lambda^{(k)} = (y^{(k)})^H Ay^{(k)} \tag{8.6}$$

$y^{(k)}$ can also be expressed as a power thus the name

$$y^{(k)} = \beta^{(k)} A^k x^{(0)} \quad \beta^{(k)} = (\prod_{i=0}^{k} \left\| x^{(i)} \right\|)^{-1} \tag{8.7}$$

This method generate a sequence of unitary $\{y^{(k)}\}$ s.t. for $k \to \infty$, they align in the direction of eigenvector x_1 in all cases, we have $\lambda^{(k) \to \lambda_1}$ for $k \to \infty$.
The stopping condition is $\left|\lambda^{(k} - \lambda^{(k-1)}\right| < \epsilon \left|\lambda^{(k)}\right|$.

8.2 Convergence of power method

Since $x_1, ..., x_n$ are assumed to be linearly independent, they are a basis of $\mathbb{C}^n$. We can thus expand them as

$$x^{(0)} = \sum_{i=1}^{n} \alpha_i x_i, \quad y^{(0)} = \beta^{(0)} \sum_{i=1}^{n} \alpha_i x_i, \quad \text{with } \beta^{(0)} = \frac{1}{\left\| x^{(0)} \right\|} \text{ and } \alpha_i \in \mathbb{C} \tag{8.8}$$

At step k we have

$$y^{(k)} = \beta^{(k)} \sum_{i=1}^{n} \alpha_i \underbrace{\lambda_i^k}_{\substack{\text{because} \\ x^{(1)} = A y^{(0)}}} x_i, \qquad \beta^{(k)} = \frac{1}{\prod_{i=0}^{k} \left\| x^{(i)} \right\|} \tag{8.9}$$

therefore

$$y^{(k)} = \lambda_1^k \beta^{(k)} \Big(\alpha_1 x_1 + \sum_{i=2}^{n} \alpha_i \frac{\lambda_i^k}{\lambda_1^k} x_i \Big) \tag{8.10}$$

We see that $y^{(k)}$ tends to align to x, since $\frac{\lambda_i}{\lambda_1} < 1 \ \forall i \geq 2$.

8.3 Inverse power method

As the previous one, but if A is nonsingular we can use A^{-1} whose eigenvalues are reciprocal of those of A, to obtain the eigenvalue of A with minimum modulus

$$x^{(k)} = A^{-1} y^{(k-1)}, \quad y^{(k)} = \frac{x^{(k)}}{\left\| x^{(k)} \right\|}, \quad \mu^{(k)} = (y^{(k)})^H A^{-1} y^{(k)} \tag{8.11}$$

$$\Rightarrow \lim_{k \to \infty} \mu^{(k)} = \frac{1}{\lambda_n} \tag{8.12}$$

We can use LU or Colesky factorization to compute $x^{(k)}$ in $A x^{(k)} = y^{(k-1)}$.

8.4 Power method with shift

If we use $A_\mu = A - \underbrace{\mu}_{\text{shift}} I$ whose eigenvalues are $\lambda(A_\mu) = \lambda(A) - \mu$

$$x^{(k)} = A \mu^{-1} y^{(k-1)}, \quad y^{(k)} = \frac{x^{(k)}}{\left\| x^{(k)} \right\|}, \quad \lambda_\mu^{(k)} = \frac{1}{(y^{(k)})^H A^{-1} y^{(k)}} \tag{8.13}$$

The searched eigenvalue is approximately $\underbrace{\lambda(A)}_{\substack{\text{The } \lambda \\ \text{closest to} \\ \mu}} = \lambda_\mu + \mu$

8.5 Gershgorin circles-computing the shift

LeT $A \in \mathbb{C}^{n \times n}$, the Gerhgorin circles are $c_i^{(n)}$ (row circle), $c_i^{(c)}$(column circle), associated with $i - th$ row and column such that

$$c_i^{(r)} = \{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|\} \tag{8.14}$$

$$c_i^{(c)} = \{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ji}|\} \tag{8.15}$$

All eigenvalues of A belong to the region of $\mathbb{C}^n$ defined by the intersection of $c_i^{(r)}$ and $c_i^{(c)}$ $\forall i$ (the union of all row circles and column circles). There is no guarantee that a circle contains eigenvalues, unless if it's isolated. The circle provide a guess for the shift. All the eigenvalues of a strictly diagonally dominant matrix are non-null.

8.6 QR method

If A and B are similar ($P^{-1}AP = B$), then $\lambda_A = \lambda_B$

$$BP^{-1}x = P^{-1}Ax = \lambda P^{-1}x \tag{8.16}$$

A method to compute all the eigenvalues of A is transforming it in a similar diagonal/-triangular matrix.
The QR method uses repeatecly QR factorization to compute

$$Q^{(k+1)}R^{(k+1)} = A^{(k)} \Rightarrow A^{(k+1)} = R^{(k+1)}Q^{(k+1)} \tag{8.17}$$

$A^{(k)}$ and $A^{(k+1)}$ are similar and the rate of decay to zero f lower triangular coefficients in $A^{(k)}$ depends on $\max_i \left|\frac{\lambda_{i+1}}{\lambda_i}\right| \forall i$. If A is symmetric, $A^{(k)}$ for $k \to \infty$ is diagonal.

Chapter 9

Ordinary differential equations

A differential equation involves one or more derivatives of an unknown function. If those derivatives are taken w.r.t. a single variable, it is called **ordinary differential equation**, whereas it is a **partial differential equation** if partial derivatives are present. The ODE or PDE has order p, where p is the maximum order of differentiation.
Any equation of order $p > 1$ can always be reduced to a system of p equation of order 1.
An ODE admits infinite solution. We formulate a **Cauchy problem** by adding a boundary condition on initial data to the ODE, ensuring the unicity of the solution.
We want to find $y : I \subset \mathbb{R} \to \mathbb{R}$ s.t.

$$\begin{cases} y'(t) = f(t, y(y)) \ \forall t \in I & \text{(ODE)} \\ y(t_0) = y_0 & \text{(Boundary c.)} \end{cases} \tag{9.1}$$

A function is said to be **Lipschitz-continuous** w.r.t. x is $\exists L > 0$ s.t.

$$|f(x_1) - f(x_2)| \leq L|x_1 - x_2| \quad \forall x_1, x_2 \in \mathbb{R} \tag{9.2}$$

Uniformly Lipschitz-continuous means "on the whole interval". Lipschitz continuity gives more regularity than normal continuity because incremental quotients are bounded (a.k.a. f cannot peak anywhere)

9.1 Existence and unicity (Cauchy-Lipschitz theorem

If $f(t, y)$ is continuous w.r.t. t and y, and uniformly Lipschitz continuous w.r.t. y, then the solution of the Cauchy problem exists, is unique and belong to $C^1(I)$.
Solution of the Cauchy problem are seldom explicit and often cannot be represented even in a an implicit form. Numerical methods allow for the approximation of every ODE family for which solutions exist.
The common approach it to divide $I = [t_0, T]$ into N_h intervals of length $h = \frac{(T-t_0)}{N_h}$. h is called the discretization step. Each $t_n = t_0 + n \cdot h$ is a node on which we compute $u_n \approx y_n = y(t_n)$. Lastly, $\{u_0 = y_0, u_1, ..., u_{N_h}\}$ is the numerical solution of the Cauchy problem.

9.2 Numerical differentiation

We aim to approximate a given function $f = [a, b] \to \mathbb{R}$ continuously differentiable on $[a, b]$, its derivative at a generic $\overline{x} \in [a, b]$.
(in case of ODE, we call $f \to y$ and $\overline{x} \to t_n$).
The derivative $y'(t_n)$ is given by

$$y'(t_n) = \begin{cases} = \lim_{h\to 0^+} \frac{y(t_n+h)-y(t_n)}{h} \\ = \lim_{h\to 0^+} \frac{y(t_n)-y(t_n-h)}{h} \\ = \lim_{h\to 0} \frac{y(t_n+h)-u(t_n-h)}{2h} \end{cases} \tag{9.3}$$

If Dy_n is an approximation of $y'(t_n)$, we then have three possible approaches:

- Forward finite difference

$$Dy_n^F = \frac{y(t_n + h) - y(t_n)}{h} \tag{9.4}$$

- Backward finite difference

$$Dy_n^B = \frac{y(t_n) - y(t_n - h)}{h} \tag{9.5}$$

- Centered finite difference:

$$Dy_n^C = \frac{y(t_n + h) - u(t_n - h)}{2h} \tag{9.6}$$

all for $n = 1, ..., N_h - 1$, $h = t_{n+1} - t_n = t_n - t_{n-1}$.
For both FFD and BFD we have that approximation error is

$$\tau_n = \left| y'(t_n) - Dy^{F/B} \right| \leq Ch, \quad \text{where } C = \frac{1}{2} \max_{t\in[t_n,t_{n+1} \text{ (or } t_{n-1})]} \left| y''(t) \right| \tag{9.7}$$

while for CFD it is

$$\tau_n = \left| y'(t_n) - Dy_n^C \right| \leq Ch^2, \quad \text{where } C = \frac{1}{6} \max_{t\in[t_n,t_{n+1}]} \left| y'''(t) \right| \tag{9.8}$$

We call τ_n the truncation error in t_n. τ_n is of order $p > 0$ if

$$\tau_n(h) \leq Ch^P \quad \text{for } C \geq 0 \tag{9.9}$$

a.k.a τ_n has order 1 for FFD and BFD, and order 2 for CFD.

9.3 Finite difference method for ODEs

In the Cauchy problem we can approximate the derivative $y'(t_n)$ in t_n using finite differences, obtaining $u_n \approx y(t_n)$

- Forward Euler(FE): explicit method

$$\begin{cases} \frac{u_{n+1}-u_n}{h} = f(t_n, u_n), \quad n = 0, ..., N_h - 1 \\ u_0 = y_0 \end{cases} \tag{9.10}$$

- Backward Euler(BE): implicit method

$$\begin{cases} \frac{u_{n+1}-u_n}{h} = f(t_{n+1}, u_{n+1}), \quad n = 0, ..., N_h - 1 \\ u_0 = y_0 \end{cases} \tag{9.11}$$

- Centered Euler(CE)

$$\begin{cases} \frac{u_{n+1}-u_n}{2h} = f(t_n, u_n), \quad n = 0, ..., N_h - 1 \\ u_0 = y_0, \ u_1(t.b.d) \end{cases} \tag{9.12}$$

FE is explicit since u_{n+1} depends explicitly on u_n ($u_{n+1} = u_n + hf(t_n, u_n)$) while BE is implicit since u_{n+1} is implicitly defined in terms of u_n ($u_{n+1} = u_n + hf(t_{n+1}, u_{n+1})$).
FE formula is a simple computation, while BE is a nonlinear (use Newton or F.P.I) problem. However BE is generally more stable. Since CE require u_1 to be applied, it generally preceded by a single pass of FE or BE.

9.4 Stability (on unbounded intervals)

The choice of h is not arbitrary. If h is not small enough, stability problems may arise. Given the model problem

$$\begin{cases} y'(t) = \lambda y(t) & t \in (0, \infty) \\ y(0) = 1 & \text{where } \lambda < 0 \in \mathbb{R} \end{cases} \tag{9.13}$$

the exact solution is $y(t) = e^{\lambda t}$ with $y(t) \to 0$ as $t \to \infty$.
The property that $\lim_{n\to\infty} u_n = 0$ is called **absolute stability**.
If we apply FE, we obtain $u_{n+1} = (1+\lambda h)u_n = (1+\lambda h)^{n+1}$. If $1+\lambda h < -1$, then $|u_n| \to \infty$ as $n \to \infty$, so FE is unstable. We have thus to limit h by imposing $|1 + \lambda h| < 1 (h < 2|h|)$. This condition is required on unbounded intervals since N_h (the number of t_n) may $\to \infty$ even if $h \nrightarrow 0$, in order to ensure stability.
If we apply BE to model, we get $u_{n+1} = \left(\frac{1}{1-\lambda h}\right) u_n = \left(\frac{1}{1-\lambda h}\right)^{n+1}$. Since $\lim_{n\to\infty} u_n = 0 \ \forall h$, we say that BE is unconditionally stable.

9.5 Absolute stability in perturbation control

Given a generalized model problem

$$\begin{cases} y'(t) = \lambda(t)y(t) + r(t) \\ y(0) = 1 \end{cases} \tag{9.14}$$

on an unbounded interval, with λ and r two continuous functions.
If λ and r are constant, we get $y(t) = \left(1 + \frac{r}{\lambda}\right)e^{\lambda t} - \frac{r}{\lambda}$ which tends to $\frac{-r}{\lambda}$ as $t \to \infty$, thus a method would not be absolutely stable on it. However it is possible to prove that a method which is absolutely stable on the original mode problem keeps perturbations under control even when applied to the generalized problem as $t \to \infty$.
If we introduce a method to compute z_n, which is perturbed by p_k at each time step k, representing truncation and numerical errors, we can compute $e_n = |z_n - u_n|$. We find that e_n is bounded by

$$|e_n| \leq \varphi(\lambda)|p| \quad \text{where } \varphi(\lambda) = 1 + \left|\frac{2}{\lambda}\right| \tag{9.15}$$

We also have $\lim_{n\to\infty} |e_n| = \frac{|p|}{|\lambda|}$, so the error caused by perturbation doesn't depend neither on n nor h.
e_n is called the perturbation error at step n.
In cases where $\lambda_{mm} > 0$ and $\lambda_{max} < \infty$, we can extend the control of perturbation of model problem to normal Cauchy problems if

$$-\lambda_{max} < \frac{\partial f}{\partial y(t,y)} < -\lambda_{min} \quad \forall t \geq 0, \forall y \in Dy \tag{9.16}$$

In this case, the steplength h should be chosen as function of $\frac{\partial f}{\partial y}$, depending on the case

- If h is constant
$$0 < h < 2 \max_{t\in[t_0,T]} \frac{\partial f}{\partial y}(t, y(t)) \tag{9.17}$$
- if h depends on the step
$$0 < h_n < 2\frac{\alpha}{|f_y(t_n, u_n)|} \quad \text{for } \alpha < 1 \tag{9.18}$$

9.6 Convergence of forward Euler

A numerical method is convergent if

$$\forall n = 0, .., N_h \quad |y_n - u_n| \leq C(n) \tag{9.19}$$

where $C(h) \to 0$ when $h \to 0$.
Moreover, if $\exists p > 0$ s.t. $C(h) = O(h) = O(h^P)$ ($\exists c > 0$ s.t. $C(h) \leq ch^P$ for $\max p$), then the method converges with order p.
In the case of FE, we have that if $y \in C^2([0,T])$ and f uniformly Lipschitz continuous on y, then

$$|y(t_n) - u_n| \leq c(t_n)h, \quad \forall n \geq 0 \tag{9.20}$$

where

$$c(t_n) = \frac{e^{Lt_n} - 1}{2L} \max_{t\in[0,T]} \left|y''(t)\right| \tag{9.21}$$

with L Lipschitz constant.
The method converges with order $p = 1$.

The local truncation error of the method represents the error that would be generated by forcing the exact solution to satisfy that specific numerical scheme. For FE we have

$$T_{n+1}(h) = \frac{y(t_{n+1}) - y(t_n)}{h} - y'(t_n) \tag{9.22}$$

The global truncation error is $T(h) = \max_n |T_n(h)|$. For FE this corresponds to

$$T(h) = \frac{1}{2} \max_{t \in [t_0, T]} \left| y''(t) \right| h \tag{9.23}$$

The same results can be applied to BE. If f also satisfies $\frac{\partial f}{\partial y}(t, y) \leq 0 \ \forall t \in [0, T]$, $\forall y \in (-\infty, \infty)$, we have the more precise estimate

$$|y(t_n) - u_n| \leq h t_n \frac{1}{2} \max_{t \in [0, T]} \left| y''(t) \right| \tag{9.24}$$

9.7 Consistency

Consistency is necessary in order to achieve convergence, since it fulfills the basic assumption that e_n in infinitesimal w.r.t. h. If violated, it would inhibit the global error $\to 0$ when $h \to 0$.
The error follows $O\left(\frac{1}{h}\right)$ when h approaches 0, so it can blow up due to round-off errors if h is too small.

9.8 Crank-Nicholson method (Trapezoidal method)

CN belongs to the family of Runge-Kutta methods, which use a single step h but evaluate $f(t, y)$ several times per interval $[t_n, t_{n+1}]$. The number of evaluation at each step is called the order w.r.t. h.
CN is of order 2, obtained by applying the fundamental theorem of integration to the Cauchy problem on $[t_n, t_{n+1}]$

$$\int_{t_n}^{t_{n+1}} y'(t) dt = \int_{t_n}^{t_{n+1}} f(t, y(t) = dt \to y_{n+1} - y_n = \int_{t_n}^{t_{n+1}} f(t, y(t)) dt \tag{9.25}$$

Then, we use the trapezoidal method to approximate the integral

$$u_{n+1} - u_n = \frac{h}{2}(f(t_n, u_n) + f(t_{n+1}, u_{n+1})) \quad \forall n \geq 0 \tag{9.26}$$

The method is unconditionally stable when applied to the model problem, and is implicit. Its explicit variant is called Heun's method, still of order 2

$$u_{n+1} - u_n = \frac{h}{2}(f(t_n, u_n) + f(t_{n+1}, u_n + h f(t_n, u_n))) \tag{9.27}$$

9.9 Improved Euler method (midpoint method)

9.10 Runge-Kutta of order 4 (Simpson method)

9.11 Systems of ODEs

9.12 Other notions of ODEs

Chapter 10

Finite elements and boundary-value problems