Numerical Mathematics

What is numerical mathematics?

Numerical mathematics is a branch of applied mathematics and generally refers to the mathematical study of computational procedures, algorithms or methods for the approximate calculation of certain quantities (on the computer).

Those quantities can be:

- evaluation of functions, e.g. $\sin(1)$; $e^{1,7}$
- solution of linear equations / systems, e.g.
  $Ax=b$ with $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, $x \in \mathbb{R}^n$,
  $n > 1.000.000$
- solution of nonlinear equations, e.g $xe^x=3$
- computer-aided simulation of complex problems as e.g.
  - weather forecast
  - fluid mechanics; building cars, ships
  - medicin, e.g. simulation of bone healing

Numerical Mathematics is about

- construction of suitable solution methods which are
  - fast ("efficient")
  - reliable, $\|x_{true} - x_{numerical}\| \leq tol$
  - robust against e.g. measurement errors
- mathematical analysis of those methods (convergence, speed, effort)
- efficient realization on a computer

Numerical Mathematics has many subareas

- Numerical analysis
- Numerical linear algebra
- Numerical optimization
- Numerical methods for ODE's, PDE's
- Numerical Finance
- Computational Physics, Biology
- ... many more...

Section 1: Solving nonlinear equations

Given a nonlinear equation

$$\tilde{f}(x) = \tilde{b}$$

with $\tilde{f}: [a,b] \subset \mathbb{R} \to \mathbb{R}$ continuous, $\tilde{b} \in \mathbb{R}$. For solving this nonlinear equation, we reformulate it as

$$f(x) := \tilde{f}(x) - \tilde{b} = 0$$

and find the zeros of f. The problem we consider is: given $f: [a,b] \subset \mathbb{R} \to \mathbb{R}$, find $x^* \in [a,b]$ such that $f(x^*) = 0$.

In special cases, it is possible to determine the solutions by hand, e.g.

$$f(x) = x^2 - 3x + 2 = 0$$

has zeros $x_1 = 1$, $x_2 = 2$. But even for polynomials of order greater than 4, there exists no explicit solution formula $\Rightarrow$ need numerical approximations.

Methods for approximating the zeros of f are usually iterative. The aim is to generate a sequence of values x_k such that

$$\lim_{k \to \infty} x_k = x^*$$

In contrast to linear equations, methods for root finding of nonlinear equations usually

depends on the choice of inital guess x_0.

Definition 1.1 (global and local convergence)

1) Numerical methods for which convergence to x^* holds for any choice of $x_0 \in [a,b]$ are said to be globally convergent to x^*.

2) Numerical methods that only converge if x_0 belongs to a suitable neighborhood of root x^* are called locally convergent to x^*.

Definition 1.2: (Convergence of order p)

A sequence (x_k) generated by a numerical method is said to converge to x^* with order $p \geq 1$ if $\exists C > 0$:

$$|x_{k+1} - x^*| \leq C |x_k - x^*|^p \quad \forall k \geq k_0$$

The numerical method is said to be of order p and C is called the convergence factor

Remark 1.3:

1) If $p=1$ in Def. 1.2. in order for x_k to converge to x^* it is necessary that $C<1$. We speak of linear convergence.

2) If $p=2$: quadratic convergence; $p=3$: cubic convergence, ...

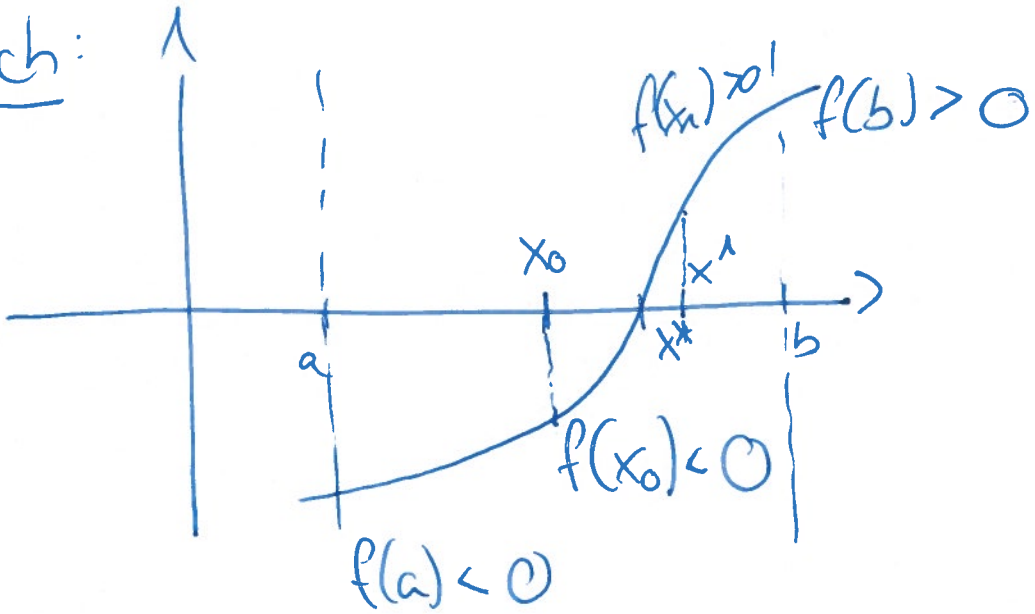
Section 1.1 The Bisection Method

Recall from Analysis:

Theorem 1.4 (Intermediate Value Theorem (IVT))
Suppose that $f:[a,b]\to\mathbb{R}$ is continuous and y is a real number between $f(a)$ and $f(b)$. Then there is a $c\in[a,b]$ such that $f(c)=y$.

Idea: Set $y=0$; then our zero $x^*=c$.
$\Rightarrow$ From IVT follows that either $f(c)=0$ or $f(a)\cdot f(b)<0$. Why?

Sketch:



To find the zero, we proceed as follows:

$k=0$: Set $a_k\leftarrow a$, $b_k\leftarrow b$.
while $|b_k-a_k|<tol$
1) compute midpoint of $[a_k,b_k]$: $x_k:=\frac{1}{2}(a_k+b_k)$
2) if $f(x_k)=0$, stop
3) if $sign(f(x_k))=sign((f(a_k))$: $a_{k+1}\leftarrow x_k$ $b_{k+1}\leftarrow b_k$

4) Else:

$b_{k+1} \leftarrow x_k$; $a_{k+1} \leftarrow a_k$

5) $k \leftarrow k+1$

For the above algorithm, we have

$$|b_k - a_k| = |b_0 - a_0|/2^k \quad \text{and}$$

$a_0 \leq a_k \leq a_{k+1} < b_{k+1} \leq b_k \leq b_0$ for $k = 0, 1, 2, \ldots$

Thus, we have 2 monotone bounded sequences with $\lim_{k\to\infty} a_k = \hat{a}$, $\lim_{k\to\infty} b_k = \hat{b}$.

It follows

$$|\hat{b} - \hat{a}| = \lim_{k\to\infty} |b_k - a_k| = \lim_{k\to\infty} |b_0 - a_0|/2^k = 0$$

$\Rightarrow \hat{a} = \hat{b}$.

Also $\operatorname{sign} f(a_k) = \operatorname{sign} f(a_0) = -\operatorname{sign} f(b_0) = -\operatorname{sign} f(b_k)$ for $k = 0, 1, 2, \ldots$ so wlog $f(a_0) > 0$, then $f(b_0) < 0$ so $f(a_k) > 0 > f(b_k)$.

For the limit $k \to \infty$, we get due to continuity $f(\hat{a}) = f(\hat{b})$ and $x^* = \hat{a} = \hat{b}$ is the zero we are looking for.

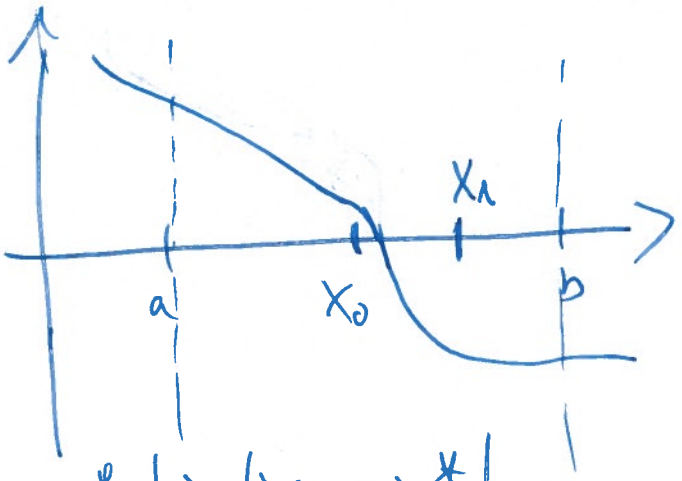
Remark 1.5. (Convergence of bisection method)

1) As we only used continuity in the above argument, the bisection method is globally convergent.

2) The convergence of the bisection method is of no order, as we cannot ensure that

$$|x_{k+1} - x^*| \leq C|x_k - x^*| \text{ with } C < 1,$$

see e.g.



where $|x_1 - x^*| > |x_0 - x^*|$.

Section 1.2. Fixed-point iterations

Idea: Reformulate for given continuous $f: [a,b] \to \mathbb{R}$ the rootfinding problem $f(x) = 0$ into an equivalent problem $x - g(x) = 0$. Here $g(x): [a,b] \to \mathbb{R}$ has to be chosen such that $g(x^*) = x^*$ whenever $f(x^*) = 0$.

This approximating zeros x^* of f is turned into finding **fixed-points** of g:

given x_0, $x_{k+1} = g(x_k)$, $k \geq 0$ (fixed-point iteration)

g: associated iteration function.

Remark 1.6: Note that the choice of g is in general not unique.

E.g. solving

$$f(x) = xe^x - 3 = 0$$ can result in

1) $x = 3 \cdot e^{-x} \Rightarrow g_1(x) = 3e^{-x}$

2) $x = \ln(3/x) \Rightarrow g_2(x) = \ln(3) - \ln(x)$

Theorem 1.7 (contraction mapping theorem)

Suppose $g: \mathbb{R}^n \to \mathbb{R}^n$ is a contraction mapping, i.e. $\exists$ a constant $L < 1$ where

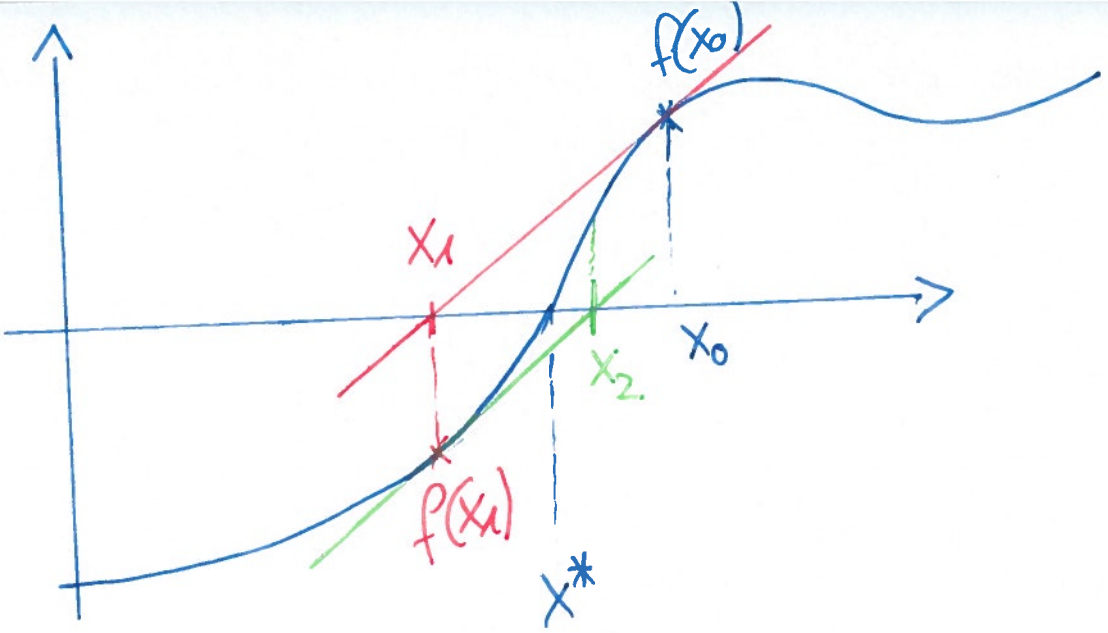
$$\|g(u) - g(v)\| \leq L\|u - v\| \text{ for all } u, v \in \mathbb{R}^n$$

then the iterates x_k of $x_{k+1} \leftarrow g(x_k)$ for $k = 0, 1, 2, \ldots$ converge to the unique fixed-point x^* with $g(x^*) = x^*$.

Section 1.3 Newton's Method

Assume that f is continuously differentiable.

Idea: Use tangent line through a point on the graph to obtain a new estimate for the solution x^* of $f(x) = 0$.



If $x \approx x_k$, then $f(x) \approx f(x_k) + f'(x_k)(x - x_k)$.
Instead of solving $f(x) = 0$, we solve

$$f(x_k) + f'(x_k)(x - x_k) = 0,$$

which leads to the iteration procedure

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Remark 1.8: Newton's method is a particular kind of fixed-point iteration with

$$g(x) = x - \frac{f(x)}{f'(x)}$$

Theorem 1.9: (Convergence of Newton's method)
Let $f \in C^2([a,b])$ with $f(x^*) = 0$ and $f'(x^*) \neq 0$ (i.e. x^* is a simple zero).

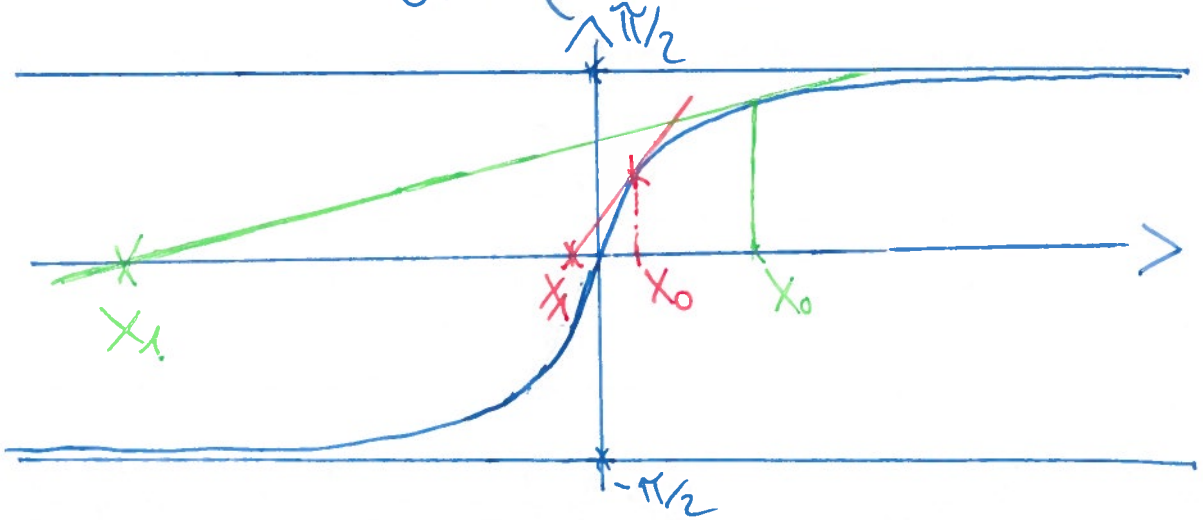
Then there exists a $\delta > 0$ where $|x_0 - x^*| < \delta$ implies that $x_k \to x^*$ and $(x_{k+1} - x^*)/(x_k - x^*)^2$ converges to $f''(x^*)/(2f'(x^*))$ as $k \to \infty$.

Proof: see [Stewart, 2022, Theorem 3.8]

Remark 1.10: This means that the convergence of Newton's method is local and that the order of convergence is $p = 2$, i.e. Newton's method is locally quadratic.

Remark 1.11: (Reliability of Newton's method)
A problem is how x_0 has to be chosen in order that it is "close enough". Newton's method can also fail to converge for e.g.

$$\tan^{-1}(x) = 0.$$



Section 2 Number representation and types of errors

On a computer with finite memory, one can never have exact computations, as e.g. saving $\pi = 3.1415926\ldots$ would take an infinite amount of memory. In addition it can be that the computational operations itself are not exact. In this section, we explore how numbers are represented in the computer, what errors can occur and how we can control these errors.

Section 2.1. Number representation

Only a finite subset of the real numbers can be represented on the computer. How do we e.g. choose this representation if we have N memory positions available?

One could: - fix sign with 1 position
- k positions after the point
- $N-k-1$ positions before the point.

We can then represent $x \in \mathbb{R}$ by

$$x = \pm (d_{N-2} d_{N-1} \dots d_k . d_{k-1} \dots d_0)_b$$

with the so-called fixed-point system.
This can be rewritten by

$$x = \pm\, b^{-k} \sum_{j=0}^{N-2} d_j b^j \,, \text{ with basis } b.$$

But this strongly limits the value of the minimum and maximum that can be represented on the computers, unless we have very large N.

Idea: The use of the fixed point limits the representable values. Use instead fixed number for positions after/before point and leave the

point position variable.

Definition 2.1. (Floating-point representation)
The floating-point representation of a real number $x \in \mathbb{R}$ is of the form

$$x = \pm (0.d_1 d_2 \dots d_m)_b \cdot b^e,$$

where
- b is the chosen basis
- $f = d_1 d_2 \dots d_m$ is is the mantissa or significand, which is the number of significant digits d_i with $(0 \leq d_i \leq b-1)$
- e is the exponent, an integer number and can vary between a finite interval of admissible values, $L \leq e \leq U$, $L < 0$; $U > 0$).

Definition 2.2 (Set of floating-point numbers)
We denote by

$$F(b, m, L, U) = \{0\} \cup \left\{ x \in \mathbb{R} : x = \pm b^e \sum_{i=1}^{m} d_i b^{-i} \right\}$$

the set of floating-point numbers with m significant digits, base $b \geq 2$ and range $L \leq e \leq U$ for the exponent.

Remark 2.3: (Uniqueness)

In order to enforce uniqueness, we assume that $d_1 \neq 0$ and that $f \geq b^{m-1}$. Then d_1 is called the leading significant digit and d_m is called the last significant digit. The representation is then called normalized.

Example 2.4

Without the above assumption, one could have the number 1 in $\mathbb{F}(10,4,-1,4)$ represented by:

$$\boxed{0.1000 \cdot 10^1}, \quad 0.0100 \cdot 10^2$$

$$0.0010 \cdot 10^3, \quad 0.0001 \cdot 10^4$$

Remark 2.5:

If $x \in \mathbb{F}(b,m,L,U) \Rightarrow -x \in \mathbb{F}(b,m,L,U)$.

Example 2.6:

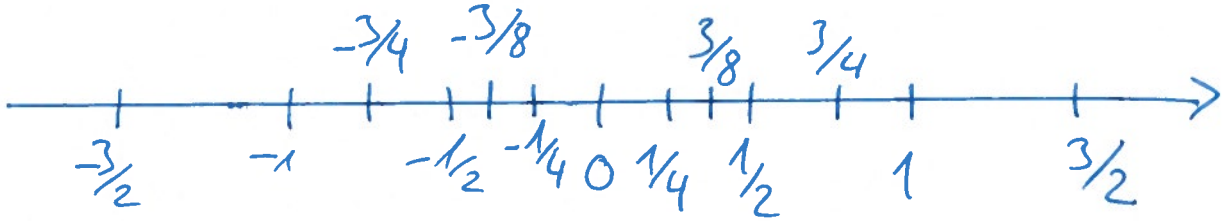
Consider $\mathbb{F}(2,2,-1,1)$. Its elements are

$(0.11)_2 \cdot 2^1 = 3/2$ $\quad$ $(0.10)_2 \cdot 2^1 = 1$

$(0.11)_2 \cdot 2^0 = 3/4$ $\quad$ $(0.10)_2 \cdot 2^0 = 1/2$

$(0.11)_2 \cdot 2^{-1} = 3/8$ $\quad$ $(0.10)_2 \cdot 2^{-1} = 1/4$

and $\{0\}$ and their respective negative values.

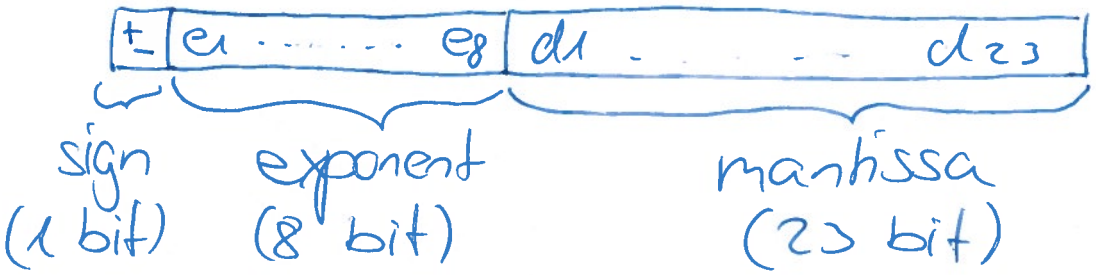


Sketch of $\mathbb{F}(2,2,-1,1)$ on the real line:
not equally distributed.

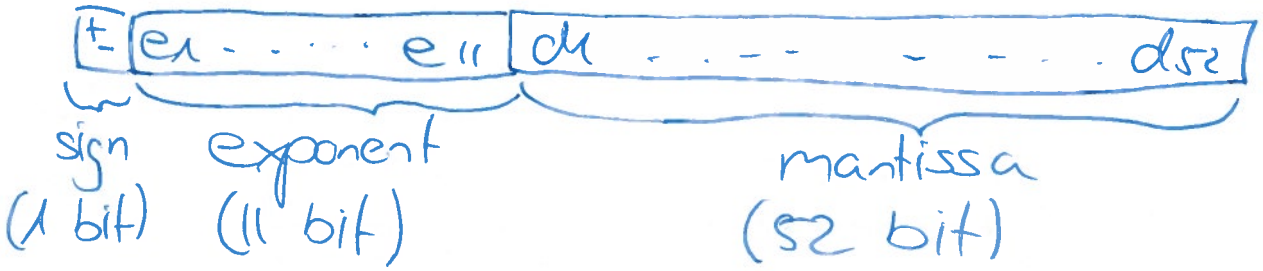
Remark 2.7 (IEEE standard)

In 1985, the Institute for Electrical and Electronic Engineers (IEEE) published the IEEE 754 Standard. Basis $b=2$ was chosen and

$\mathbb{F}(2,24,-125,128)$ for single precision (32 bit)



and $\mathbb{F}(2,53,-1021,1024)$ for double precision (64 bit)



Section 2.2 Types of Errors

The fact that $\mathbb{F}(b,m,L,U)$ only approximates a subset of $\mathbb{R}$ poses several problems:

- representation of $x \in \mathbb{R}$ in $\mathbb{F}(b,m,L,U)$
- If $x_1, x_2 \in \mathbb{F}(b,m,L,U)$ it might be that $x_1 + x_2 \notin \mathbb{F}(b,m,L,U)$

For the first problem one can introduce rounding:

Definition 2.8 ("Round-to-nearest", rarding mapping) we introduce a mapping

$$fl: \mathbb{R} \to \mathbb{F}(b,m,L,U),$$

which rounds the element in $\mathbb{R}$ to one in $\mathbb{F}(b,m,L,U)$. One common strategy would be to round-to-nearest given by

$$fl(x) = \pm (0.d_1 d_2 \ldots \tilde{d}_m)_b \cdot b^e,$$

with

$$\tilde{d}_m = \begin{cases} d_m, & \text{if } d_{m+1} < b/2 \\ d_{m+1}, & \text{if } d_{m+1} \geq b/2. \end{cases}$$

Instead of trying to model the exact behavior, we try to determine bounds for the results of floating-point operations.

Definition 2.9 (Absolute and relative error)

Let $x \in \mathbb{R}$ and let x_{approx} be an approximation.

The <u>absolute error</u> is the difference

$$|x - x_{approx}|,$$

and the <u>relative error</u> is given by

$$\frac{|x - x_{approx}|}{|x|}.$$

If $x \in \mathbb{R}$ is in the range of $\mathbb{F}(b, m, L, U)$, then the closest floating-point number to x satisfies

$$|x - fl(x)| \leq u|x|,$$

where u is the <u>unit roundoff</u>(*) (or <u>machine precision</u>). For IEEE 754, we have $u \approx 1.2 \times 10^{-7}$ for single precision and $u \approx 2.2 \times 10^{-16}$ for double precision.

For floating-point number x and y and operations $* = +, -, \times, /$, we have

$$fl(x * y) = (x * y)(1 + \varepsilon) \quad \text{for some } |\varepsilon| \leq u.$$

(*) maximum error in mantissa after given operation.

Section 2.3. When things go wrong

A) Overflow and underflow.

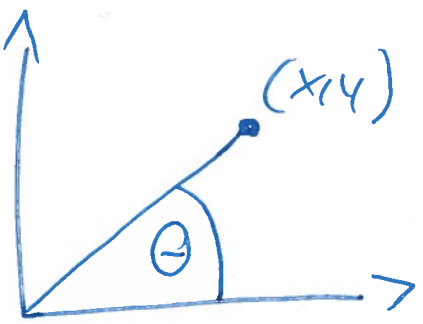
Definition 2.10 (Overflow, underflow)

If $|x| > \max\{|z| \in F(b,m,L,U)\}$, we call that overflow.

If $|x| < \min\{|z| \in F(b,m,L,U) \setminus \{0\}\}$, then $x=0$ is set and we speak of underflow.

consider computing $\cos(\theta)$ from (x,y) coordinates of a point by

$$\cos(\theta) = \frac{x}{\sqrt{x^2+y^2}}.$$



In Matlab, if using $x=y=10^{200}$, we get $\cos(\theta)=0$ (should be $1/\sqrt{2}$).

For $x=y=10^{100}$, we get

$$|\cos(\theta) - 1/\sqrt{2}| \approx 1.1 \times 10^{-16}.$$

This indicates that there is a threshold in the size of x for which bad behavior occurs $\rightsquigarrow$ overflow.

Already for $fl(x^2) = fl((10^{200})^2) = \inf$ such that we get finally $\cos(\theta) = \frac{10^{200}}{\inf} = 0$

How to prevent this? Scale x, y with s

$$\frac{x}{\sqrt{x^2+y^2}} \cdot \frac{1/s}{1/s} = \frac{x/s}{\sqrt{(x/s)^2+(y/s)^2}}$$

and set $s = |x|$ which yields

$$\frac{\operatorname{sign}(x)}{\sqrt{1+(y/|x|)^2}}.$$

This avoids overflow if $y/|x|$ is not too large.

B) Subtracting nearly equal quantities

If we compute "exactly"

$$0.73568 - 0.73445 = 0.00123,$$

and assuming that $m=3$, we get

$$0.736 - 0.734 = 0.002 = 0.2 \cdot 10^{-2},$$

which means that the absolute error

$$|0.00123 - 0.002| = 0.00077,$$ which compares to the rounding error.

For the relative error, we get

$$\frac{|0.00123 - 0.002|}{|0.00123|} \approx 63\%.$$

This is called cancellation error.

Consider computing $\frac{1-\cos(x)}{x^2}$ for $x \approx 0$.
With l'Hospitals rule, we can show that

$$\lim_{x \to 0} \frac{(1-\cos(x))}{x^2} = 1/2 .$$

If we compute the values in Matlab, we get values around $1/2$ for $x \geq 10^{-7}$, but for $x < 10^{-8}$ the result is 0.

What happens?
When $x \approx 0$, we are subtracting with $1-\cos(x)$ two equally large numbers.

We can reformulate:

$$\frac{1-\cos(x)}{x^2} = \frac{1-\cos(x)}{x^2} \cdot \frac{1+\cos(x)}{1+\cos(x)}$$

$$= \frac{1-\cos^2(x)}{x^2(1+\cos(x))}$$

Now using $1-\cos^2(x) = \sin^2(x)$, we get

$$\frac{1-\cos(x)}{x^2} = \frac{\sin^2(x)}{x^2(1+\cos(x))} = \left(\frac{\sin(x)}{x}\right)^2 \cdot \frac{1}{1+\cos(x)} .$$

Def, this formula should not work well for $x \approx \pm\pi$ as there $1+\cos(x) \approx 0$.

Section 3 Gaussian Elimination

Our goal is to solve a linear system with n linear equations and n unknowns $x_1, \dots, x_n \in \mathbb{R}$

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ \vdots \qquad\quad \vdots \qquad\qquad \vdots \qquad\quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

or short

$$Ax = b$$

with $A \in \mathbb{R}^{n \times n}$, $b, x \in \mathbb{R}^n$.

When is this solvable?

Theorem 3.1: Let $A \in \mathbb{R}^{n \times n}$ with $\det(A) \neq 0$ and $b \in \mathbb{R}^n$. Then, there exists exactly one $x \in \mathbb{R}^n$ such that $Ax = b$.

For solving this system, we first look at a special case of a triangular matrix: Consider a linear system $Ly = b$ where $L \in \mathbb{R}^{n \times n}$ is a lower triangular matrix:

$$
\begin{aligned}
l_{11} y_1 &= b_1 \\
l_{21} y_1 + l_{22} y_2 &= b_2 \\
&\vdots \\
l_{n1} y_n + l_{n2} y_2 + \dots + l_{nn} y_n &= b_n
\end{aligned}
$$

We see that we can calculate the components of y one-by-one with

$$y_i = \left(b_i - \sum_{j=1}^{i-1} l_{ij} y_j \right) / l_{ii} \quad i = 1, 2, \dots, n.$$

The same can be done for a system $Ux = y$ with $U \in \mathbb{R}^{n \times n}$ being an upper triangular matrix

$$
\begin{aligned}
u_{11} x_1 + u_{12} x_2 + \dots + u_{1n} x_n &= y_1 \\
&\vdots \\
u_{nn} x_n &= y_n ,
\end{aligned}
$$

where we get

$$x_i = \left(y_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) / u_{ii}$$

$$i = n, n-1, \dots, 1.$$

This leads to the LU factorization.

Section 3.1 LU factorization

We observe that if a matrix $A \in \mathbb{R}^{n \times n}$ can be written as a product of a lower triangular matrix $L \in \mathbb{R}^{n \times n}$ and an upper triangular matrix $U \in \mathbb{R}^{n \times n}$ as

$$A = LU,$$

we can solve the linear system

$$Ax = b \quad (\underbrace{LUx}_{=:y} = b)$$

in two steps:

(1) Find y from $Ly = b$ (forward substitution)

(2) Find x from $Ux = y$ (back substitution).

Questions are:

- how to construct L and U?
- is such a factorization always possible?

We start with the construction. The following 3 operations can be used:

1) swapping rows
2) Multiplying a row by a non-zero scalar
3) adding a multiple of one row to another

We start with

$$A = A_0 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & & & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \leftarrow \text{II} - \left(\frac{a_{21}}{a_{11}}\right) \cdot \text{I}$$

We want a_{21} to become 0, thus we choose adding row 1 multiplied by $\left(-\frac{a_{21}}{a_{11}}\right)$

$$a_{21} - a_{11}\frac{a_{21}}{a_{11}} = 0.$$

Repeating that step n-2 times yields

$$A_1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn}^{(1)} \end{bmatrix}$$

and we save the multipliers in L_1 with

$$L_1 = \begin{bmatrix} 1 & & & 0 \\ -l_{21} & & & \\ & 0 & & \\ -l_{n1} & 0 & \cdots & 0 \; 1 \end{bmatrix}$$

$$l_{21} := \frac{a_{21}}{a_{11}}$$

$$l_{n1} := \frac{a_{n1}}{a_{11}}$$

Then we have

$$L_1 A_0 = A_1$$

The elimination step is repeated $n-2$ times resulting in

$$L_{n-1} L_{n-2} \cdots L_1 A_0 = A_{n-1} =: U$$

This procedure is called Gaussian elimination.

Then, we have

$$L_1 A_0 = A_1$$

This elimination step is repeated $n-2$ times, resulting in

$$L_{n-1} L_{n-2} \cdots L_1 A_0 = A_{n-1} =: U.$$

This procedure is called Gaussian elimination. The inverse of the matrix

$$L_k = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{k+1,k} & \ddots & \\ & & \vdots & & \\ & & -l_{n,k} & & 1 \end{pmatrix}$$

(column k)

can be computed by switching the signs in the off-diagonal entries, i.e.

$$L_k^{-1} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & l_{k+1,k} & \ddots & \\ & & \vdots & & \\ & & l_{n,k} & & 1 \end{pmatrix}$$

(column k)

For the product of the L_k^{-1} holds

$$L := L_1^{-1} \cdots L_{n-1}^{-1} = \begin{pmatrix} 1 & & 0 \\ l_{21} & \ddots & \\ \vdots & \ddots & \ddots \\ l_{n1} & \cdots & l_{n,n-1} \; 1 \end{pmatrix}$$

Thus, we arrive at $A = LU$.

Is it always possible to find such L and U for given nonsingular matrix A?

Example 3.2

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

The eigenvalues of A are $\lambda_{1,2} = \pm 1$, thus $\det(A) \neq 0$. If an LU factorization exists, it has to be of form

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix} \begin{bmatrix} b & c \\ 0 & d \end{bmatrix} = \begin{bmatrix} b & c \\ ab & ac+d \end{bmatrix}.$$

We get $b = 0$, but also $ab = 1$ $\lightning$.

$\Rightarrow$ no LU factorization can exist.

Theorem 3.3. Let $A \in \mathbb{R}^{n\times n}$. The LU factorization of A with $l_{ii}=1$ for $i=1,\dots,n$ exists and is unique with U nonsingular iff the principal submatrices A_i (i.e. $A(1:i,1:i)$) of A of order $i=1,\dots,n-1$ are nonsingular and A is nonsingular.

Proof: [see additional notes for lecture]

Revisiting Example 3.2, we realize that swapping the rows yields an LU factorization.

Remark 3.4 (Pivoted LU factorization)

Let $A \in \mathbb{R}^{n\times n}$ be nonsingular. Then, there exists a permutation matrix $P \in \mathbb{R}^{n\times n}$ such that

$$PA = LU,$$

with $l_{ii}=1$ for $i=1,\dots,n$ and nonsingular U.

$P \in \mathbb{R}^{n\times n}$ is called a permutation matrix if one entry is "1" in each column and row and all others are 0.

Often, the so-called partial pivoting strategy is used, where one finds the largest entry (in absolute value) in the column k in which the

entries below the diagonal are set to 0. This element is called the pivot element. Then, one swaps the row k with the row containing the pivot element.

Remark 3.5 (Storage of LU factorization)
In each elimination step, one can save the l_{ik} in the resulting zero entries of the matrix. By consecutively overwriting the entries in A_0, one arrives at

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \boxed{l_{21}} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ l_{n1} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix} \to \cdots \to \begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ \boxed{l_{21}} & \boxed{a_{22}^{(1)}} & \cdots & \cdots & a_{2n}^{(1)} \\ \vdots & \ddots & \ddots & & \vdots \\ l_{n1} & \cdots & \boxed{l_{n,n-1}} & & \boxed{a_{nn}^{(n-1)}} \end{pmatrix}.$$

Section 3.2 Condition number of matrix

In this section, we examine how changes in the input data $b \in \mathbb{R}^n$ induce changes in the solution x of

$$Ax = b.$$

We need the following definitions.

Definition 3.6 (Properties of matrix norm)
A matrix norm $\|\cdot\|_M$ on $\mathbb{R}^{n\times n}$ is called

1) submultiplicative, if

$$\|AB\|_\mu \le \|A\|_\mu \|B\|_\mu \quad \forall A, B \in \mathbb{R}^{n\times n}$$

2) consistent with vector norm $\|\cdot\|_*$ on $\mathbb{R}^n$, if (or compatible)

$$\|Ax\|_* \le \|A\|_\mu \|x\|_* \quad \forall A \in \mathbb{R}^{n\times n}, x \in \mathbb{R}^n.$$

Definition/Theorem 3.7: Let $\|\cdot\|_*$ be a vector norm in $\mathbb{R}^n$. By

$$\|A\|_\mu := \sup_{x\neq 0} \frac{\|Ax\|_*}{\|x\|_*} = \sup_{\|x\|_*=1} \|Ax\|_* \quad , A \in \mathbb{R}^{n\times n}$$

we define the so-called induced matrix norm. The induced matrix norm is submultiplicative and consistent with $\|\cdot\|_*$.

Example 3.8.: Let $x \in \mathbb{R}^n$. Then

$\|x\|_1 = \sum_{i=1}^{n} |x_i|$ induces the matrix norm

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}| \quad \text{and} \quad \|x\|_\infty = \max_{i=1,\dots,n} |x_i|$$

induces the norm $\|A\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|$

with $A \in \mathbb{R}^{n\times n}$.

Note that the definitions of submultiplicative, consistent with vector norm and induced matrix norms can be extended to non-square matrices, see Stewart Sec 1.5.

Now let $\|\cdot\|_*$ be a vector norm on $\mathbb{R}^n$ and let $\|\cdot\|_M$ be the by $\|\cdot\|_*$ induced matrix norm.

Let $0 \neq b \in \mathbb{R}^n$ be the exact right-hand-side and let Δb be an additive perturbation. For the solution of the linear system, we get

$$x := A^{-1}b \quad \text{and} \quad x + \Delta x := A^{-1}(b + \Delta b)$$

which leads to $\Delta x = A^{-1}\Delta b$. We get

$$\frac{\|\Delta x\|_*}{\|x\|_*} = \frac{\|A^{-1}\Delta b\|_*}{\|x\|_*} \leq \|A^{-1}\|_M \frac{\|\Delta b\|_*}{\|b\|_*} \frac{\|Ax\|_*}{\|x\|_*}$$

$$\leq \|A^{-1}\|_M \|A\|_M \frac{\|\Delta b\|_*}{\|b\|_*}.$$

Definition 3.9 (Condition number of matrix)
For $A \in \mathbb{R}^{n \times n}$ nonsingular, the number

$$K_M(A) = \|A\|_M \|A^{-1}\|_M$$

is called condition number of A with respect to induced matrix norm $\|\cdot\|_M$.
We call the problem of solving the linear system $Ax = b$

1) well-conditioned, if $K_M(A)$ is "small"
2) ill-conditioned, if $K_M(A)$ is "big"

Note, that $\kappa_M(A) \geq 1$ as

$$1 = \|AA^{-1}\|_M \leq \|A\|_M \|A^{-1}\|_M = \kappa_M(A).$$

Also note, that the condition number is dependent on the problem at hand, not on the solution process.

Example 3.10: We solve the linear system of equations

$$\begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.86419999 \\ 0.14400001 \end{pmatrix} =: b$$

The exact solution is

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.9911 \\ -0.4870 \end{pmatrix}.$$

Due to flawed measurement, we obtain the right-hand-side

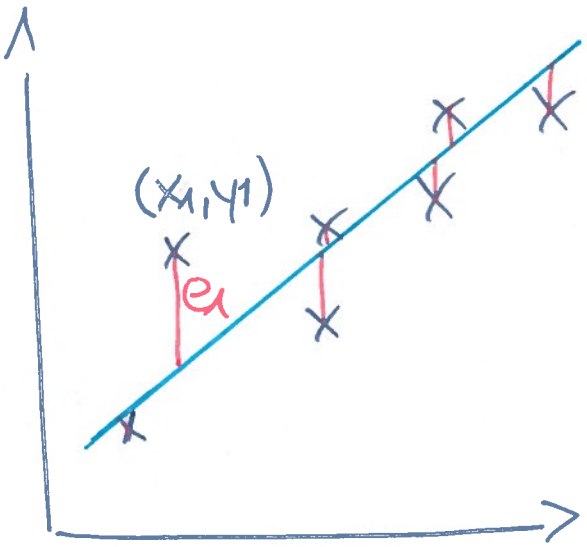
$$\tilde{b} = \begin{pmatrix} 0.8642 \\ 0.1440 \end{pmatrix}.$$

For this right-hand side, we obtain the solution $\begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \end{pmatrix}$.

What happened?

Section 4: Least squares problems

Least squares problems are commonly used in statistics for fitting data to a model. For example, we want to fit a set of data points (x_i, y_i), $i = 1, \dots, n$ to a straight line $g(x) = ax + b$. The "best" fit is



typically minimizing the <u>sum of squares</u> of errors

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(g(x_i) - y_i \right)^2$$

over all possible choices of coefficients a and b.

Section 4.1: The Normal Equations

We set $c := \begin{bmatrix} b \\ a \end{bmatrix}$ in the above setting and have $e_i = [1, x_i] \begin{bmatrix} b \\ a \end{bmatrix} - y_i$ and thus

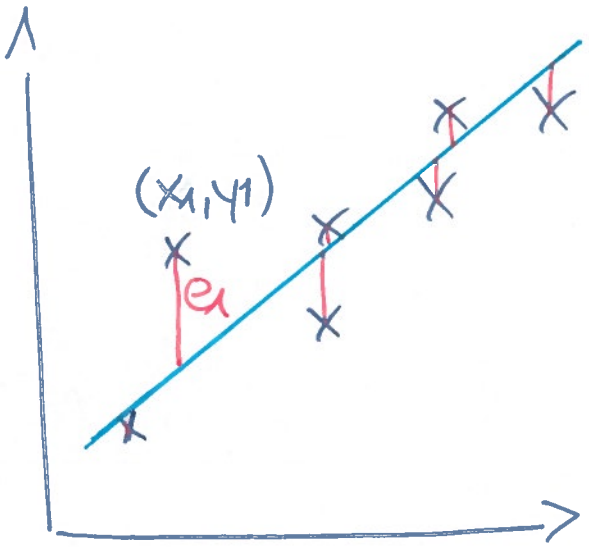
$$e = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = Ac - y.$$

Thus, we want to minimize

$$\sum_{i=1}^{n} e_i^2 = e^T e = \|e\|_2^2 = \|Ac - y\|_2^2$$

Section 4: Least squares problems

Least squares problems are commonly used in statistics for fitting data to a model. For example, we want to fit a set of data points (x_i, y_i), $i = 1, \dots, n$ to a straight line $g(x) = ax + b$. The "best" fit is



typically minimizing the sum of squares of errors

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(g(x_i) - y_i\right)^2$$

over all possible choices of coefficients a and b.

Section 4.1: The Normal Equations

We set $c := \begin{bmatrix} b \\ a \end{bmatrix}$ in the above setting and have $e_i = [1, x_i]\begin{bmatrix} b \\ a \end{bmatrix} - y_i$ and thus

$$e = \begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = Ac - y.$$

Thus, we want to minimize

$$\sum_{i=1}^{n} e_i^2 = e^T e = \|e\|_2^2 = \underbrace{\|Ac - y\|_2^2}_{=: \varphi(c)}.$$

We get

$$\varphi(c) = (Ac-y)^T(Ac-y) = c^T A^T A c - 2c^T A^T y + y^T y$$

For a minimum, we set the gradient of φ to 0

$$\nabla\varphi(c) = 2A^T Ac - 2A^T y \stackrel{!}{=} 0,$$

it follows that c is solution of square system

$$A^T Ac = A^T y,$$

known as the <u>normal equations</u>.

<u>Remark 4.1</u> (Solution of normal equations)
If $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ has rank n, then the solution to the normal equations exists and is unique.

In order to see how perturbations in initial data affect the solution, we would like to have some analogue of the condition number for rectangular matrices.

If $A \in \mathbb{R}^{m \times n}_{m \geq n}$ has n linearly independent columns, then $A^T A$ is invertible and we get

$$c = (A^T A)^{-1} A^T y .$$

Definition 4.2 (Pseudo-inverse)

Let $A \in \mathbb{R}^{m \times n}$, $m \geq n$ with $\operatorname{rank}(A) = n$. The matrix

$$A^+ := (A^T A)^{-1} A^T \in \mathbb{R}^{n \times m}$$

is called the pseudo-inverse of A.

This means the solution to the least-squares problem $\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$ is given by

$$x = A^+ b .$$

Definition 4.3 (Least squares condition number)

Let $A \in \mathbb{R}^{m \times n}$. We call

$$K_2(A) = \|A\|_2 \, \|A^+\|_2$$

the least squares condition number. It holds $K_2(A) \geq 1$ if $A \in \mathbb{R}^{m \times n}$.

Remark 4.4 (Condition number of normal equations)
When we solve the normal equations system with square matrix A^TA, the condition number is squared $K_2^2(A)$.

Section 4.2: QR factorization

Definition 4.5: (Orthogonal matrix)
A quadratic matrix $Q \in \mathbb{R}^{n\times n}$ is called orthogonal if $Q^T = Q^{-1}$ or equivalently

$$Q^TQ = QQ^T = \mathbb{I}_n .$$

Remark 4.6: Note, that orthogonal matrices preserve the 2-norm, as for $Q \in \mathbb{R}^{n\times n}$, $x \in \mathbb{R}^n$ we have

$$\|Qx\|_2^2 = (Qx)^T(Qx) = x^TQ^TQx = x^Tx = \|x\|_2^2,$$

which means $\|Qx\|_2 = \|x\|_2$.

We now assume, that the matrix $A \in \mathbb{R}^{m\times n}$, $m \geq n$ admits a QR factorization

$$A = QR$$

with $Q \in \mathbb{R}^{m\times m}$ orthogonal and $R \in \mathbb{R}^{m\times n}$ being an upper triangular matrix.

For $m > n$, we can split Q and R consistently:

$$A = QR = \begin{bmatrix} Q_1 | Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1 ,$$

where $Q_1 \in \mathbb{R}^{m \times n}$ has orthonormal columns and $R_1 \in \mathbb{R}^{n \times n}$ is upper triangular. This is called <u>skinny, thin or reduced QR factorization</u> of A.

Consider $\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$

with $A \in \mathbb{R}^{m \times n}$ with $\operatorname{rank}(A) = n < m$, $b \in \mathbb{R}^m$.

Then $\|Ax - b\|_2 = \|QRx - b\|_2$

$$= \|Q(Rx - Q^T b)\|_2 = \|Rx - Q^T b\|_2$$

$$= \left\| \begin{bmatrix} R_1 \\ 0 \end{bmatrix} x - \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} b \right\|_2$$

$$= \sqrt{\|R_1 x - Q_1^T b\|_2^2 + \|Q_2^T b\|_2^2} .$$

Thus, we minimize $\|Ax - b\|_2$ by solving

$$R_1 x = Q_1^T b$$

by back substitution. Since R_1 is nonsingular, we obtain a unique solution.

This means, when solving the least-squares problem, we obtain

$$\min_{x \in \mathbb{R}^n} \|Ax-b\|_2 = \|Q_2^T b\|_2 = \sqrt{\|b\|_2^2 - \|Q_1^T b\|_2^2}$$

Theorem 4.7 (Existence and Uniqueness of QR factorization)

Each $A \in \mathbb{R}^{m \times n}$, $m \geq n$ with $\operatorname{rank}(A) = n$ admits a unique QR factorization with unitary $Q \in \mathbb{R}^{m \times n}$ and $R \in \mathbb{R}^{n \times n}$ with

$$A = QR \quad \text{with} \quad R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \; R_1 \in \mathbb{R}^{n \times n}$$

where R_1 is a nonsingular upper triangular matrix with positive diagonal entries.

Proof: tutorial.

Remark 4.8: (Condition number of QR factorization)

For the QR factorization, we obtain

$$K_2(R) = K_2(A).$$

Section 4.3: Givens Rotation

In the following, we have $c = \cos(\theta)$ and $s = \sin(\theta)$.

Idea: multiply matrix A with a suitable rotation matrix

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \quad c^2 + s^2 = 1,$$

such that entries below diagonal are set to zero.

Let $x, y \in \mathbb{R}$. Our goal is to choose $c, s \in \mathbb{R}$ such that

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix},$$

with $\alpha = \pm\sqrt{x^2 + y^2}$ since rotations preserve the 2-norm.

From the second equation, we get that

$$sx + cy = 0.$$

This is achieved by

$$s = \frac{-y}{\sqrt{x^2+y^2}}, \quad c = \frac{x}{\sqrt{x^2+y^2}}.$$

By systematically zeroing out entries in A, we can create a QR factorization. For this, we choose indexes (i, j), where

$$G(i,j,c,s) = \begin{bmatrix} \ddots & & & \\ & c & \cdots & -s \\ & \vdots & \ddots & \vdots \\ & s & \cdots & c \\ & & & & \ddots \end{bmatrix} \begin{matrix} \\ \leftarrow \text{row } i \\ \\ \leftarrow \text{row } j \\ \\ \end{matrix}$$

$\uparrow$ column i $\quad$ $\uparrow$ column j

and is otherwise equal to the identity matrix.

Remark 4.9 (Storage of Givens rotation)

We do not need to store c, s both, we can save the angle θ in the resulting zero of the matrix.

Section 5: Polynomial Interpolation

With the Taylor series from analysis, we could represent certain functions, but knowledge of derivatives of arbitrarily high order was required.
Interpolation instead only requires function values and/or a few low-order derivatives at interpolation points.
This can be useful if e.g.

- one has data points, but no function through these points is known
- one wants to approximate a function by simpler functions.

For polynomial interpolation, given data points (x_i, y_i), $i = 0, \dots, n$. we want to find a polynomial P of degree $\leq d$, where

$$p(x_i) = y_i \quad , i = 0, \dots, n.$$

Section 5.1 Existence and Uniqueness

We can show that for distinct interpolation points $x_0, \dots, x_n$, there is exactly one

Section 5: Polynomial Interpolation

With the Taylor series from analysis, we could represent certain functions, but knowledge of derivatives of arbitrarily high order was required.
Interpolation instead only requires function values and/or a few low-order derivatives at interpolation points.
This can be useful if e.g.

- one has data points, but no function through these points is known
- one wants to approximate a function by simpler functions.

For polynomial interpolation, given data points (x_i, y_i), $i = 0, \dots, n$. We want to find a polynomial P of degree $\leq d$, where

$$p(x_i) = y_i \quad , i = 0, \dots, n.$$

Section 5.1 Existence and Uniqueness

We can show that for distinct interpolation points $x_0, \dots, x_n$, there is exactly one

interpolant $p(x)$ of degree $\leq n$.

Theorem 5.1: If (x_i, y_i), $i = 0, 1, \ldots, n$ and $x_i \neq x_j$ for $i \neq j$, then there is one and only one polynomial $p(x)$ of degree $\leq n$, where

$$p(x_i) = y_i \quad \text{for all } i = 0, \ldots, n.$$

Proof: 1) Existence: we use a constructive approach, providing an expression for $p(x)$. To this end, we examine the to the interpolation points $(n+1)$ Lagrange polynomials

$$L_i(x) := \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)}.$$

These polynomials are of degree n and

$$L_i(x_k) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{(x_k - x_j)}{(x_i - x_j)} = \begin{cases} 1, & \text{if } i = k \\ 0, & \text{if } i \neq k \end{cases}.$$

Then the polynomial $p(x)$

$$p(x) := \sum_{i=0}^{n} y_i L_i(x)$$

satisfies $p(x_i) = y_i$ and is of degree $\leq n$.

2) Uniqueness: Let $p(x)$ and $q(x)$ be two polynomials of degree $\leq n$ with

$$p(x_k) = q(x_k) = y_k \quad 0, \dots, n.$$

Then it follows that $r(x) := p(x) - q(x)$ is a polynomial with degree $\leq n$ with the $(n+1)$ zeros $x_0, \dots, x_n$. This can only happen if $r(x)$ is the zero polynomial, i.e. $r(x) \equiv 0$, thus $p(x) = q(x)$. □

Section 5.2 Computing the Polynomial Interpolant

The equations to be satisfied to find the interpolating polynomial $p(x) := \sum_{k=0}^{n} a_k x^k$ are linear equations with respect to the coefficients, i.e. we could solve

$$V_n a = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

directly. The matrix V_n is called a Vandermonde matrix and due to Theorem 5.1, this matrix must be invertible for distinct x_i's.

Remark 5.2: While it is possible to compute the coefficients with the above linear system, it is generally not recommended, as the condition number of Vandermonde matrices V_n grows exponentially in n see Stewart, Table 4.1.1..

Remark 5.3: The Lagrange polynomials, that we used in Theorem 5.1 could be used, but are in practice usually to expensive. Often, one is interested in the evaluation of the interpolation polynomial at a given point, thus we do not need $p(x)$ explicitely.

5.3. Polynomial Interpolation with Divided Differences

Notation: the starting point are the so-called divided differences.

Definition 5.4 (Divided Differences)

Let (x_i, y_i) $i = 0, \dots, n$ be given data points and let $f: \mathbb{R} \to \mathbb{R}$ with $f(x_i) = y_i$ be the function to be approximated. We denote by

$$f[x_0, x_1, \dots, x_n] := \frac{f[x_0, x_1, \dots, x_{n-1}] - f[x_1, \dots, x_n]}{x_0 - x_n}$$

the n-th divided difference of f at $x_0, \dots, x_n$. We set $f[x_i] := f(x_i)$, $i = 0, \dots, n$.

For the first divided difference, we get

$$f[x_0, x_1] = \frac{f[x_0] - f[x_1]}{x_0 - x_1} = \frac{f(x_0) - f(x_1)}{x_0 - x}.$$

Theorem 5.5 (Independence of ordering)

If $y_0, y_1, \dots, y_n$ is any permutation of $x_0, x_1, \dots, x_n$ and all x_i's are distinct, then

$$f[x_0, \dots, x_n] = f[y_0, \dots, y_n].$$

Proof: see Stewart, Theorem 4.2.

How do divided differences relate to polynomial interpolation?

We note that

$$f(x) = f(x_0) + \frac{f(x) - f(x_0)}{x - x_0}(x - x_0)$$
$$= f[x_0] + f[x_1, x_0](x - x_0).$$

We repeat this for

$$f[x_1, x_0] = f[x_{11}, x_0] + \frac{f[x_1, x_0] - f[x_{11}, x_0]}{x - x_1}(x - x_1)$$

$$= f[x_0, x_1] + f[x_1, x_0, x_1](x - x_1).$$

We get for general n:

$$f[x, x_0, \dots, x_{n-1}] = f[x_n, x_0, \dots, x_{n-1}] + \frac{f[x, x_0, \dots x_{n-1}] - f[x_n, x_0 \dots, x_{n-1}]}{x - x_n}$$
$$\cdot (x - x_n)$$

$$= f[x_0, \dots, x_n] + f[x, x_0, \dots, x_n](x - x_n)$$

We rewrite

$$f(x) = f[x_0] + (x - x_0) f[x, x_0]$$
$$= f[x_0] + (x - x_0)\left(f[x_0, x_1] + (x - x_1) f[x, x_0, x_1]\right)$$
$$= f[x_0] + (x - x_0) f[x_0, x_1] + (x - x_0)(x - x_1) f[x, x_0, x_1]$$

For general n, we get:

$$f(x) = \sum_{i=0}^{n} f[x_0, x_1, \dots, x_i] \prod_{j=0}^{i-1} (x - x_j) \Bigg\} \text{polynomial of degree } \leq n.$$

$$+ \prod_{j=0}^{n} (x - x_j) f[x, x_0, x_1, \dots, x_n] \Bigg\} \text{polynomial of degree } n+1.$$

Thus we set the interpolating polynomial to

$$p_n(x) := \sum_{i=0}^{n} f[x_0, x_1, \dots, x_i] \prod_{j=0}^{i-1} (x - x_j).$$

Note that to compute $p_n(x)$, we only need to compute $n+1$ divided differences

$$f[x_0, x_1, \dots, x_i], \quad i = 0, 1, \dots, n.$$

How do we evaluate this efficiently for each new x?

Consider divided difference table:

$$\begin{array}{lccccc}
x_0: & f[x_0] & & & & \\
x_1: & f[x_1] & f[x_0, x_1] & & & \\
x_2: & f[x_2] & f[x_1, x_2] & f[x_0, x_1, x_2] & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \\
x_n: & f[x_n] & f[x_{n-1}, x_n] & f[x_{n-2}, x_{n-1}, x_n] & \cdots & f[x_0, x_1, \dots, x_n]
\end{array}$$

To compute each value, we need the

value left and left-and-up of the entry,

e.g. $$f[x_0,x_1,x_2] = \frac{f[x_0,x_1] - f[x_1,x_2]}{x_0 - x_2}.$$

Once the divided differences $D_i = f[x_0,x_1,\dots,x_i]$ have been computed, we evaluate $p_n(x)$ at x

$$p_n(x) = D_0 + D_1(x-x_0) + D_2(x-x_0)(x-x_1) + \dots + D_n(x-x_0)(x-x_1)\cdots(x-x_{n-1}),$$

which requires $\sum_{i=0}^{n}(1+2i) = (n+1)^2$ operations.

This can be improved by pulling out common factors:

$$p_n(x) = D_0 + (x-x_0)\left[D_1 + (x-x_1)\left\{D_2 + (x-x_2)\left(D_3 + \dots \right.\right.\right.$$
$$\left.\left.\left. \dots \left[D_{n-1} + (x-x_{n-1})D_n\right]\dots\right)\right\}\right]$$

This nested approach only needs $3n$ operations for evaluating $p_n(x)$ at x.

5.4. The Interpolation Error

Theorem 5.6: Let $x_0, x_1, \dots, x_n$ be $n+1$ distinct points and let x be a point belonging to the domain of a given function f.

Assume that $C^{n+1}(I_x)$, where I_x is the smallest interval containing $x_0, x_1, \ldots, x_n$ and x. Then, the interpolation error at the point is given by

$$f(x) - p(x) = \frac{f^{(n+1)}(\theta)}{(n+1)!}(x-x_0)\cdot\ldots\cdot(x-x_n)$$

where $\theta \in I_x$.

Proof: The result is true if x coincides with any $x_k, k=0,\ldots,n$.

Otherwise for $t \in I_x$ define

$$h(t) := (f(x)-p(x))\prod_{k=0}^{n}(t-x_k) - (f(t)-p(t))\prod_{k=0}^{n}(x-x_k)$$

Since $f \in C^{(n+1)}(I_x)$, it follows that $h \in C^{(n+1)}(I_x)$. In addition h has $(n+2)$ zeros in I_x since for x_i, $i=0,1,\ldots,n$, we have

$$h(x_i) = (f(x)-p(x))\prod_{k=0}^{n}(x_i-x_k) - (f(x_i)-p(x_i))\prod_{k=0}^{n}(x-x_k) = 0,$$

and

$$h(x) = (f(x)-p(x))\prod_{k=0}^{n}(x-x_k) - (f(x)-p(x))\prod_{k=0}^{n}(x-x_k) = 0.$$

Between every adjacent pair of zeros, there is a zero of h' due to Rolle's theorem.

Thus h' has at least $(n+1)$ zeros in I_x.
Repeating this argument, we have that $h^{(n+1)}$ has at least 1 zero in I_x. We pick one and call it θ.

Then

$$h^{(n+1)}(t) = (f(x)-p(x))\left(\prod_{k=0}^{n}(t-x_k)\right)^{(n+1)} - \underbrace{\left(f^{(n+1)}(t)-p^{(n+1)}(t)\right)}_{=0,\text{ since degree} \leq n}\cdot\prod_{k=0}^{n}(x-x_k)$$

We have

$$\prod_{k=0}^{n}(t-x_k) = t^{n+1} - q(t), \text{ where } q(t) \text{ is polynomial}$$

of degree $\leq n$. Thus

$$\left(\prod_{k=0}^{n}(t-x_k)\right)^{(n+1)} = (n+1)! - 0 = (n+1)!.$$

As θ is zero of $h^{(n+1)}$ it follows

$$0 = h^{(n+1)}(\theta) = (f(x)-p(x))(n+1)! - f^{(n+1)}(\theta)\cdot\prod_{k=0}^{n}(x-x_k)$$

Rearranging the terms gives the formula for the interpolation error.

Theorem 5.6 (Interpolation error estimate)

Let $a \leq x_0 < x_1 < x_2 < \dots < x_n = b$, $a<b$, $y_i = f(x_i)$ and $f \in C^{n+1}[a,b]$.
Let p_n be the interpolating polynomial of degree n.
Then, for each $x \in [a,b]$, there exist $c_x \in [a,b]$ with

$$f(x) - p_n(x) = \frac{f^{(n+1)}(c_x)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

Proof: See previous notes, or [Steward, Theorem 4.3]

Corollary 5.7

Let $x_i = a + ih$ with $h = \frac{b-a}{n}$. Then

$$\frac{1}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \leq \frac{h^{n+1}}{n+1}$$

In particular $\| f - p_n \|_\infty \leq \frac{h^{n+1}}{n+1} \| f^{(n+1)} \|_\infty$

Proof: Let $x = a + th \in [a,b]$ for $0 \leq t \leq n$.

Then $\prod_{i=0}^{n} (x - x_i) = \prod_{i=0}^{n} ((t-i)h) = h^{n+1} \underbrace{\prod_{i=0}^{n} (t-i)}_{\leq n!} \leq n! h^{n+1}$

$x_i = a + ih$

[w.l.o.g consider $0 < t < 1$. Then $\frac{1}{n!} \prod_{i=0}^{n} |t-i| = |t| \prod_{i=1}^{n} \underbrace{\frac{i-t}{i}}_{\leq 1} \leq 1$.
in general use permutation.]

The estimate for $\| f - p_n \|_\infty$ then follows from Theorem 5.6. □

Remark 5.8

(i) Note that $\| f^{(n+1)} \|_\infty$ can grow quickly for $n \to \infty$, and the estimate in Corollary 5.7 has to be used with care.

Consider $f(x) = \frac{1}{1+x^2}$, $-5 \leq x \leq 5$: $\| f^{(n+1)} \|_\infty \sim (n+1)!$

and $\frac{n!}{h^{n+1}} \to \infty$ as $n \to \infty$.

Remark 5.9

Let $\Delta^n := \{x_0^n, \dots, x_n^n\} \subseteq [a,b]$ be a set of interpolation points. Then

(i) For every sequence $\{\Delta^n\}$ there exists a continuous function $f \in C[a,b]$, s.t. p_n does not converge uniformly to f:

$$\forall \{\Delta^n\}_n \; \exists f \in C[a,b] : \|f - p_n\|_\infty \not\to 0 \text{ as } n \to \infty.$$

(cf Runge phenomenon Sec 4.1.1.13 and Lebesgue numbers Sec 4.1.2)

(ii) For all continuous functions there exists a sequence of interpolating polynomials that converges uniformly to that function

$$\forall f \in C[a,b] \; \exists \{\Delta^n\}_n : \|f - p_n\|_\infty \to 0 \text{ as } n \to \infty.$$

(cf Weierstrass' Theorem, Theorem 4.12)

Summary

- Lagrange polynomial interpolation is well-posed for distinct mutually different interpolation points, i.e., there exists a unique interpolating polynomial p_n
- There are different representations (= choice of basis) and algorithms to compute p_n. (Monomials $\{x^j\}_{j=0}^n$, Lagrange $\{L_j\}_{j=0}^n$, Newton $\{\prod_{i<j}(x - x_i)\}_{j=0}^n$)
- Conditioning / stability depends strongly on the choice of interpolation points: condition number $\sim$ Lebesgue constant Λ_n (Sec 4.1.2)
  - equidistant: $\Lambda_n \sim \frac{2^{n+1}}{e\, n \ln(n)}$ for $n \to \infty$
  - Chebyshev: $\Lambda_n \sim \frac{2}{\pi} \ln(n)$ for $n \to \infty$
- There are more ways to interpolate (Hermite Sec 4.1.1.14, Splines Sec 4.2, Orthogonal Polynomials Sec 4.7.2, trigonometric polynomials Sec 4.7.3.)

6 Numerical Integration
(also called numerical quadrature)

aim: Compute approximations of $\int_a^b f(x)\,dx$

($a<b$, $a,b \in \mathbb{R}$, $f:[a,b] \to \mathbb{R}$ integrable)

recall: For $f \in C[a,b]$, the definition of the Riemann integral implies

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{n-1} (x_{i+1} - x_i)\, f(\xi_i)$$

where

- $a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$ is a partition of the interval $[a,b]$ into smaller intervals $[x_i, x_{i+1}]$ with mesh-size
$$h = \max_i h_i, \quad h_i := x_{i+1} - x_i$$
and $x_i \le \xi_i \le x_{i+1}$.

approach:

a) Decompose $[a,b]$ into smaller intervals $[x_i, x_{i+1}]$

b) Approximate f by a simple function f_n on each interval $[x_i, x_{i+1}]$

c) Use the integral of simple function f_n over $[x_i, x_{i+1}]$ to approximate $\int_a^b f(x)\,dx$:

$$I(f) := \int_a^b f(x)\,dx \approx \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f_n(x)\,dx =: I_n(f)$$

Definition 6.1

(i) The largest number $m \in \mathbb{N}$ for which

$$\int_a^b x^j\,dx = I_n(x^j) \quad \forall\, 0 \le j \le m,$$

is called the degree of exactness.

(ii) The term $e_i := \int_{x_i}^{x_{i+1}} f(x)\,dx - \int_{x_i}^{x_{i+1}} f_n(x)\,dx$ is called the local error.

Let us now consider several examples

6.1 (Composite) Rectangle rule

Use piecewise constant approximation

$$f_h(x) = f(x_i) \quad \text{for } x \in (x_i, x_{i+1}).$$

This yields $I(f) \approx I_h^{\text{rect}}(f) := \sum_{i=0}^{n-1} (x_{i+1} - x_i) f(x_i)$

(compare to Riemann sums and the choice $\xi_i = x_i$)

Lemma 6.2 (Error estimate for composite rectangle rule)

Let $f \in C^1[a,b]$. Then

$$|I(f) - I_h^{\text{rect}}(f)| \le \frac{1}{2} \|f'\|_\infty (b-a) h,$$

and the degree of exactness for I_h^{rect} is 0.

Proof

Observe that $|I(f) - I_h(f)| \overset{\text{triangle ineq.}}{\le} \sum_{i=0}^{n-1} |e_i|$ ⊛

Moreover, $e_i = \int_{x_i}^{x_{i+1}} f(x)\,dx - (x_{i+1} - x_i) f(x_i) = \int_{x_i}^{x_{i+1}} f(x) - f(x_i)\,dx$

Taylor's formula (Thm 1.2, Sec 1.6.1) yields

$$f(x) = f(x_i) + \int_{x_i}^{x} f'(t)\,dt$$

Thus $|e_i| \overset{\text{triangle ineq.}}{\le} \left| \int_{x_i}^{x_{i+1}} \int_{x_i}^{x} |f'(t)|\,dt\,dx \right|$

$$\le \|f'\|_\infty \int_{x_i}^{x_{i+1}} x - x_i\,dx$$

$$= \|f'\|_\infty \tfrac{1}{2} h_i^2$$

$\sum_{i=0}^{n-1} h_i = b-a$

Therefore: $|I(f) - I_h(f)| \le \sum_{i=0}^{n-1} \frac{1}{2} \|f'\|_\infty h_i^2 \le \frac{1}{2} \|f'\|_\infty (b-a) h.$

The degree of exactness is obviously 0. □

6.2 (Composite) Trapezoidal rule

Use piecewise linear approximation

$$f_n(x) = f(x_i)\, L_{0,i}(x) + f(x_{i+1})\, L_{1,i}(x)$$

with Lagrange interpolation polynomials of order 1, i.e.

$$L_{0,i}(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}}, \quad L_{1,i}(x) = \frac{x - x_i}{x_{i+1} - x_i}$$

Then $I_h^{tr}(f) = \sum_{i=0}^{n-1} (x_{i+1} - x_i)\, \frac{1}{2}\left(f(x_{i+1}) + f(x_i)\right)$,

because $\int_{x_i}^{x_{i+1}} f_n(x)\,dx = f(x_i) \underbrace{\int_{x_i}^{x_{i+1}} L_{0,i}(x)\,dx}_{= \frac{1}{2}(x_{i+1} - x_i)} + f(x_{i+1}) \underbrace{\int_{x_i}^{x_{i+1}} L_{1,i}(x)\,dx}_{= \frac{1}{2}(x_{i+1} - x_i)}$

$$= (x_{i+1} - x_i)\, \frac{1}{2}\left(f(x_i) + f(x_{i+1})\right)$$

Lemma 6.3 (Error estimate for trapezoidal rule)

For $f \in C^2[a,b]$ it holds

$$|I(f) - I_n^{tr}(f)| \le \frac{1}{12} \|f''\|_\infty (b-a)\, h^2$$

and the degree of exactness is 1.

proof: Consider the local error

$$e_i = \int_{x_i}^{x_{i+1}} f(x) - f_n(x)\,dx \overset{(4.1.7) \text{ in Steward}}{=} \int_{x_i}^{x_{i+1}} \frac{f''(c_{x,i})}{2} (x - x_i)(x - x_{i+1})\,dx$$

$c_{x,i} \in (x_i, x_{i+1})$

$$\overset{\text{MVT}}{=} \frac{1}{2} f''(c_i) \underbrace{\int_{x_i}^{x_{i+1}} (x - x_i)(x - x_{i+1})\,dx}_{= -\frac{1}{6}(x_{i+1} - x_i)^3} = -\frac{1}{12} f''(c_i)\, h_i^3$$

(sign of $(x - x_i)(x - x_{i+1})$ does not change for $x \in (x_i, x_{i+1})$; see Wade, Thm 5.24)

The error estimate follows by summation of local errors.

The degree of exactness is obviously 1. □

6.3 Composite midpoint rule

Use piecewise constant approximation

$$f_n(x) = f\left(\frac{x_i + x_{i+1}}{2}\right), \quad x \in (x_i, x_{i+1})$$

... to obtain $I_n^{mid}(f) = \sum_{i=0}^{n-1} h_i f\left(\frac{x_i + x_{i+1}}{2}\right)$.

Lemma 6.4 (Error estimate for midpoint rule)

Let $f \in C^2[a,b]$. Then

$$|I(f) - I_n^{mid}(f)| \leq \frac{1}{24} \|f''\|_\infty h^2 (b-a)$$

and the degree of exactness is 1.

proof: Set $x_i^m = \frac{1}{2}(x_i + x_{i+1})$

Taylor: $f(x) = f(x_i^m) + f'(x_i^m)(x - x_i^m) + \int_{x_i^m}^{x} (x-t) f''(t)\, dt$

Then $e_i = \int_{x_i}^{x_{i+1}} f(x) - f(x_i^m)\, dx$

$$= \underbrace{\int_{x_i}^{x_{i+1}} f'(x_i^m)(x - x_i^m)\, dx}_{= f'(x_i^m) \int_{x_i}^{x_{i+1}} (x - x_i^m)\, dx = 0} + \int_{x_i}^{x_{i+1}} \underbrace{\int_{x_i^m}^{x} (x-t) f''(t)\, dt}_{\geq 0}\, dx$$

Therefore $|e_i| \leq \|f''\|_\infty \underbrace{\int_{x_i}^{x_{i+1}} \int_{x_i^m}^{x} (x-t)\, dt\, dx}_{= \frac{1}{2}\int_{x_i}^{x_{i+1}} (x - x_i^m)^2 dx = \frac{1}{24} h_i^3} = \frac{1}{24} \|f''\|_\infty h_i^3$.

The error estimate follows from summation of the local errors.

Degree of exactness = 1 follows immediately. □

Remark 6.5 Although both, $I_n^{rect}(f)$ as well as $I_n^{mid}(f)$, use piecewise constant approximations, the order of convergence of $I_n^{mid}(f)$ is higher, i.e., quadratic in h, which is higher than expected at first. This phenomenon is called superconvergence.

⑤ 6.4 Simpson rule

Use piecewise quadratic interpolation

$$f_n(x) = f(x_i) L_{0,i}(x) + f(x_i^m) L_{1,i}(x) + f(x_{i+1}) L_{2,i}(x)$$

with quadratic Lagrange polynomials $L_{j,i}$, $j = 0,1,2$.

Since $\int_{x_i}^{x_{i+1}} L_{0,i}(x)dx = h_i \int_0^1 L_0(t)\,dt = h_i \frac{1}{6} = \int_{x_i}^{x_{i+1}} L_{2,i}(x)dx$

$$L_{0,i}(x) = L_0\left(\frac{x-x_i}{h_i}\right) \qquad L_{2,i}(x) = L_2\left(\frac{x-x_i}{h_i}\right)$$

$$L_0(t) = 2\left(t-\tfrac{1}{2}\right)(t-1) \qquad L_2(t) = 2t\left(t-\tfrac{1}{2}\right)$$

and $\int_{x_i}^{x_{i+1}} L_{1,i}(x)dx = h_i \int_0^1 L_1(t)\,dt = \frac{4}{6}$

$$L_{1,i}(x) = L_1\left(\frac{x-x_i}{h_i}\right)$$

$$L_1(t) = -4t(t-1)$$

we obtain $I_n^{si}(f) = \sum_{i=0}^{n-1} \frac{h_i}{6}\left[f(x_i) + 4f(x_i^m) + f(x_{i+1})\right]$

Lemma 6.6 (Error estimate for Simpson rule)

Let $f \in C^4[a,b]$. Then the degree of exactness of $I_n^{si}(f)$ is 3, and

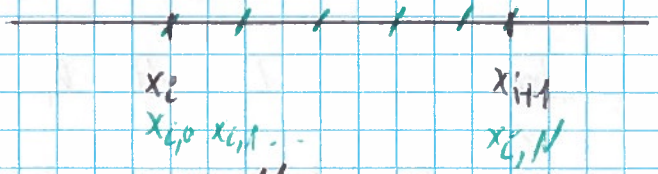
$$|I(f) - I_n^{si}(f)| \le \frac{1}{180} \|f^{(4)}\|_\infty h^4 (b-a).$$

proof: See Steward, pp. 332-333.

6.5 Newton-Cotes Methods

... are generalizations of the midpoint, trapezoidal and Simpson rules.

- quadrature points $x_{i,k} := x_i + H_i k$, $H_i = \frac{h_i}{N}$, $k = 0, \dots, N$ (equally spaced)



- Set $f_n(x) = \sum_{k=0}^{N} f(x_{i,k})\, L_{i,k}(x)$

with Lagrange polynomials $L_{i,k}$ of degree N associated to $\{x_{i,k}\}_k$

$$L_{i,k}(x) = \prod_{l: l \neq k} \frac{x - x_{i,l}}{x_{i,k} - x_{i,l}}$$

- Observe that $L_{i,k}(x) = L_k\left(\frac{x - x_i}{h_i}\right)$, with $L_k(t)$ Lagrange polynomials of degree N associated to $\left\{\frac{k}{N}\right\}_{k=0}^{N}$, i.e.

$$L_k(t) = \prod_{l: l \neq k} \frac{(t - l/N)}{\left(\frac{k}{N} - \frac{l}{N}\right)}.$$

- Define the quadrature weights $w_k := \int_0^1 L_k(t)\, dt$,

we obtain $I_{n,N}^{NC}(f) = \sum_{i=0}^{n-1} h_i \left(\sum_{k=0}^{N} f(x_{i,k})\, w_k \right)$

Theorem 6.7 (Error estimates for Newton-Cotes)

The degree of exactness of $I_{n,N}^{NC}(f)$ is $\begin{cases} N, & N \text{ odd} \\ N+1, & N \text{ even} \end{cases}$

For $f \in C^{N+1}[a,b]$ it holds

$$|I(f) - I_{n,N}^{NC}(f)| \leq \frac{h^{N+1}}{(N+1)\, N^{N+1}} \|f^{(N+1)}\|_\infty (b-a).$$

Proof: Follows from interpolation error estimates (see Theorem 4.3, p226 in Stewart)

Corollary 5.7 with $H_i = \frac{h_i}{N} = \frac{x_{i+1} - x_i}{N}$ and $\|f - p_n\|_{\infty, [x_i, x_{i+1}]} \leq \frac{H_i^{N+1}}{N+1} \|f^{(N+1)}\|_\infty$

(7) **Remark 6.8**

(i) If N is even and $f \in C^{N+2}[a,b]$ the error is $O(h^{N+2})$ as $h \to 0$ (superconvergence of Simpson rule).

(ii) Since $x_{i,0} = x_i$ and $x_{i,N} = x_{i+1}$, $I_{n,N}^{NC}(f)$ are called closed formulas.

If $x_{i,0} = x_i + \frac{h}{N+2}$, $x_{i,N} = x_{i+1} - \frac{h}{N+2}$, $x_{i,k} = x_{i,0} + k\frac{h}{N+2}$

for $k = 0, \dots, N$, $N \geq 0$, then the corresponding formulae are called open. Similar error estimates hold, (compare to mid-point rule).

(iii) Since f is approximated by its interpolating polynomial, $I_{n,N}^{NC}(f)$ is called interpolatory quadrature formula.

6.6 Stability of quadrature formulae

Consider a quadrature rule $I_n(f) = (b-a)\sum_{k=0}^{n} w_k f(x_k)$

to approximate $\int_a^b f(x)\,dx = I(f)$

Lemma 6.9

If $f, \tilde{f} \in C[a,b]$, then

$$|I_n(f) - I_n(\tilde{f})| \le (b-a)\left(\sum_{k=1}^{n} |w_k|\right) \|f - \tilde{f}\|_\infty$$

proof: Follows from linearity of $f \mapsto I_n(f)$ and the triangle inequality.

Remark 6.10

(i) Integration as a mathematical problem $f \mapsto I(f)$ is stable, i.e.

$$|I(f) - I(\tilde{f})| \le \left|\int_a^b f(x) - \tilde{f}(x)\,dx\right| \le (b-a)\|f - \tilde{f}\|_\infty.$$

(ii) If $w_k \ge 0$, and the degree of exactness is ≥ 0, then $\sum_{k=1}^{n} |w_k| = 1$, i.e. the integration rule has the same stability as $f \mapsto I(f)$.

(iii) For Newton-Cotes with N large, w_k may become negative (s. p. 336). Then $\sum_{k=1}^{n} |w_k| \gg 1$, and the Newton-Cotes methods becomes unstable. Compare to Runge's phenomenon (Sec 4.1.1.13) and Lebesgue constants in Sec 4.1.2.

(9) 6.7 Gaussian quadrature

Consider $I_n(f) := \sum_{i=1}^{n} w_i f(x_i)$

with n interpolation (quadrature) points x_i

and weights w_i.

How can ... we choose $\{x_i\}, \{w_i\}$ to maximize the degree of exactness k?

$$\int_a^b p(x)\,dx = I_n(p; a, b) \quad \forall \text{ polynomials of degree } \leq k$$

have $2n$ free parameters x_i, w_i.

require $k+1$ conditions to integrate all polynomials of degree k exactly

↳ $2n = k+1$ or $k = 2n-1$.

6.7.1 orthogonal polynomials

Consider inner product $(f,g) := \int_a^b w(x) f(x) g(x)\,dx$

for integrable weight $w(x) > 0$.

Definition 6.11

A family of polynomials $\{\phi_j\}_{j \in \mathbb{N}}$ is called orthogonal if

- $(\phi_i, \phi_j) = 0$ if $i \neq j$.
- $\deg(\phi_j) = j$

Theorem 6.12

Let $\{\phi_j\}_{j \in \mathbb{N}}$ be a family of orthogonal polynomials. Then ϕ_j has exactly j simple roots in (a,b).

proof: We argue by contradiction.

Let $x_1, \dots, x_m$ be all zeros of ϕ_j satisfying

- $a < x_1 < x_2 < \dots < x_m < b$
- $\phi_j(x)$ changes sign at x_i. ($\rightsquigarrow \phi_j(x) = h(x) \prod_{i=1}^{m} (x-x_i)^{r_i}$, r_i odd)

Since $\deg \phi_j = j$, we have $m \le j$. Assume $m < j$.

Define $\psi(x) = \prod_{i=1}^{m} (x-x_i)$, $\deg \psi = m$.

By definition $\phi_j \psi$ does not change sign in (a,b).

Therefore,

$$0 = (\psi, \phi_j) = \int_a^b w(x)\, \psi(x)\, \phi_j(x)\, dx \neq 0,$$

($\deg \psi < j$)

which is a contradiction, i.e. $j = m$. ∎

6.7.2 Integration rule

Let $n \in \mathbb{N}$. Denote $\{x_i\}_{i=1}^{n}$ the (distinct) roots of ϕ_n.

Let $w_i = \int_a^b w(x)\, L_i(x)\, dx$

with $L_i(x)$ the ith Lagrange polynomial of degree $n-1$ associated to $\{x_i\}_{i=1}^{n}$.

We define the Gauß rule of degree $n-1$ associated to the weight w by

$$I_{n-1,w}(f) := \sum_{i=1}^{n} w_i f(x_i).$$

Theorem 6.15

The Gauß-rule of order $n-1$ has degree of exactness $2n-1$ and

$$|I_w(f) - I_{n-1,w}(f)| \le \frac{C(a,b,w,n)}{(2n)!} \|f^{(2n)}\|_\infty$$

where $I_w(f) = \int_a^b w(x) f(x)\, dx$, for $f \in C^{2n}[a,b]$ and

$$C(a,b,w,n) = \int_a^b w(x) \prod_{i=1}^{n} (x-x_i)^2\, dx.$$

6.7 Gaussian quadrature

Consider $I_n(f) := \sum_{i=1}^{n} w_i f(x_i)$

with n interpolation (quadrature) points x_i
and weights w_i.

How can we choose $\{x_i\}, \{w_i\}$ to maximize the degree of exactness k?

$$\int_a^b p(x)\,dx = I_n(p; a, b) \quad \forall \text{ polynomials of degree } \leq k$$

have $2n$ free parameters x_i, w_i.
require $k+1$ conditions to integrate all polynomials of degree k exactly
↳ $2n = k+1$ or $k = 2n-1$.

6.7.1 orthogonal polynomials

Consider inner product $(f, g) := \int_a^b w(x) f(x) g(x)\,dx$

for integrable weight $w(x) > 0$.

Definition 6.11
A family of polynomials $\{\phi_j\}_{j \in \mathbb{N}}$ is called orthogonal if

- $(\phi_i, \phi_j) = 0$ if $i \neq j$.
- $\deg(\phi_j) = j$

Theorem 6.12
Let $\{\phi_j\}_{j \in \mathbb{N}}$ be a family of orthogonal polynomials. Then ϕ_j has exactly j simple roots in (a, b).

proof: We argue by contradiction.

Let $x_1, \dots, x_m$ be all zeros of ϕ_j satisfying

- $a < x_1 < x_2 < \dots < x_m < b$
- $\phi_j(x)$ changes sign at x_i. ($\Rightarrow \phi_j(x) = h(x) \prod_{i=1}^{m} (x-x_i)^{r_i}$, r_i odd)

Since $\deg \phi_j = j$, we have $m \le j$. Assume $m < j$.

Define $\psi(x) = \prod_{i=1}^{m} (x-x_i)$, $\deg \psi = m$.

By definition $\phi_j \psi$ does not change sign in (a,b).

Therefore,

$$0 = (\psi, \phi_j) = \int_a^b w(x)\, \psi(x)\, \phi_j(x)\, dx \neq 0,$$

$\deg \psi < j$

which is a contradiction, i.e. $j = m$. ∎

6.7.2 Integration rule

Let $n \in \mathbb{N}$. Denote $\{x_i\}_{i=1}^{n}$ the (distinct) roots of ϕ_n.

Let $$w_i = \int_a^b w(x)\, L_i(x)\, dx$$

with $L_i(x)$ the ith Lagrange polynomial of degree $n-1$ associated to $\{x_i\}_{i=1}^{n}$.

We define the Gauss rule of degree $n-1$ associated to the weight w by

$$I_{n-1,w}(f) := \sum_{i=1}^{n} w_i\, f(x_i).$$

Theorem 6.15

The Gauss-rule of order $n-1$ has degree of exactness $2n-1$ and

$$|I_w(f) - I_{n-1,w}(f)| \le \frac{C(a,b,w,n)}{(2n)!} \|f^{(2n)}\|_\infty$$

where $I_w(f) = \int_a^b w(x) f(x)\, dx$, for $f \in C^{2n}[a,b]$ and

$$C(a,b,w,n) = \int_a^b w(x) \prod_{i=1}^{n} (x-x_i)^2\, dx.$$

(11)

proof: Let f be a polynomial of degree $\leq 2n-1$.

Using synthetic division

$$f(x) = q(x)\,\phi_n(x) + r(x)$$

with q, r polynomials of degree $\leq n-1$. Then

$$\int_a^b w(x)\,f(x)\,dx = \underbrace{\int_a^b w(x)\,q(x)\,\phi_n(x)\,dx}_{=0} + \int_a^b w(x)\,r(x)\,dx$$

$$\overset{\deg(r)=n-1}{=} \sum_{i=1}^{n} w_i\,r(x_i) = \sum_{i=1}^{n} w_i \left(q(x_i)\underbrace{\phi_n(x_i)}_{=0} + r(x_i)\right)$$

↑ definition of w_i

↳ degree of exactness $\leq n-1$

$$= \sum_{i=1}^{n} w_i\,f(x_i),$$

i.e., $I_{n-1,w}(f)$ has degree of exactness $\geq 2n-1$. (*)

For the error estimate consider the Hermite interpolating polynomial $\tilde{p}_n$ of degree $2n-1$ satisfying

$$\tilde{p}_n(x_i) = f(x_i) \quad \text{and} \quad \tilde{p}_n'(x_i) = f'(x_i). \qquad \text{(see Sec. 4.1.1.14)}$$

Then

$$\int_a^b w(x)\,f(x)\,dx - \sum_{i=1}^{n} w_i\,f(x_i)$$

$$\overset{\text{deg of exactness} \geq 2n-1}{=} \int_a^b w(x)\,\underbrace{\left[f(x) - \tilde{p}_n(x)\right]}_{}\,dx - \sum w_i \underbrace{\left(f(x_i) - \tilde{p}_n(x_i)\right)}_{=0}$$

$$\overset{(4.1.1.8)}{=} \frac{f^{(2n)}(c_x)}{(2n)!} \prod_{i=1}^{n} (x-x_i)^2$$

$$\overset{\text{GMVT}}{=} \frac{f^{(2n)}(c)}{(2n)!} \int_a^b w(x) \prod_{i=1}^{n} (x-x_i)^2\,dx,$$

which proves the error estimate. □

(*) Since $0 < \int_a^b w(x)\,\phi_n^2(x)\,dx$ and $\sum w_i\,\phi_n(x_i)^2 = 0$ and $\deg(\phi_n^2) = 2n$, the degree of exactness $= 2n-1$.

(17)

Lemma 6.14

The weights w_i of the Gauß-rule $I_{n-1,w}$ are positive.

proof:

Let L_j be the Lagrange polynomial of degree $n-1$ associated to $\{x_i\}_{i=1}^n$.

Since $I_{n-1,w}$ is exakt for polynomials of degree $2n-1$ and $\deg(L_j^2) = 2n-2$, we have that

$$0 < \int_a^b w(x)\,(L_j(x))^2\,dx = \sum_{i=1}^n w_i \underbrace{L_j(x_i)^2}_{\begin{cases} = 0 & \text{if } i \neq j \\ = 1 & \text{if } i = j \end{cases}} = w_j.$$

□

Remark 6.15

Since $w_i > 0$, we have that $\sum_{i=1}^n |w_i| = \sum_{i=1}^n w_i$.

In view of Lemma 6.9, we conclude that for $f, \tilde{f} \in C[a,b]$

$$|I_{n-1,w}(f) - I_{n-1,w}(\tilde{f})| \leq (b-a)\,\|f - \tilde{f}\|_\infty,$$

i.e. the Gauß-rule has the same (good) stability properties as the math. problem $f \mapsto I(f)$, see Remark 6.10.

Example 6.16 $a = -1, b = 1$.

For $w(x) = 1$, ϕ_j are given by the Legendre polynomials, defined by

$$\phi_0(x) = 1, \qquad \phi_1(x) = x.$$

$$(n+1)\,\phi_n(x) = (2n+1)\,x\,\phi_n(x) - n\,\phi_{n-1}(x), \quad n \geq 1.$$

It holds $$\int_{-1}^1 \phi_n(x)\,\phi_m(x)\,dx = \frac{\delta_{m,n}}{2n+1}$$

We have the rules

$n-1$	x_i	deg. exact	w_i
0	0	1	2
1	$\pm\sqrt{1/3}$	3	1, 1
2	$0, \pm\sqrt{\frac{3}{5}}$	5	$\frac{5}{9}, \frac{8}{9}, \frac{5}{9}$

Remark 6.17

(i) Let $\{x_i\}, \{w_i\}$ define an integration rule to approximate

$$\int_a^b f(x)\,dx \approx (b-a) \sum_{i=1}^{n} w_i f(x_i),$$

then we can obtain an integration rule to approximate

$$\int_c^d g(z)\,dz : \approx (d-c) \sum_{i=1}^{n} w_i\, g\left((d-c)\frac{x_i - a}{b-a} + c\right)$$

$$= (d-c) \sum_{i=1}^{n} w_i\, g(z_i)$$

with $z_i = \frac{d-c}{b-a}(x_i - a) + c$.

(ii) In view of (i), we can obtain composite Gauß-Legendre rules by shifting and scaling of integration points, c.f. Sec 6.5.

We have for $a = y_0 < y_1 < \dots y_M = b$,

$$\int_a^b f(z)\,dz = \sum_{j=0}^{M-1} \int_{y_j}^{y_{j+1}} f(z)\,dz$$

$$\approx \sum_{j=0}^{M-1} (y_{j+1} - y_j) \sum_{i=1}^{n} \frac{w_i}{2} f\left(\frac{y_{j+1} - y_j}{2}(x_i + 1) + y_j\right)$$

where x_i is the ith root of the Legendre polynomial ϕ_n (in $[-1,1]$)

[Note $\sum_{i=1}^{n} w_i = \int_{-1}^{1} 1\,dx = 2$]

Remark 6.18

There is no choice of integration points $\{x_i\}_{i=1}^{n}$ and weights $\{w_i\}_{i=1}^{n}$ which achieves a degree of exactness $> 2n-1$. [Quarteroni et al, p. 430]

(1)

Numerical solution of ordinary differential equations

Consider the initial value problem

$$(IVP) \quad \begin{cases} y'(t) = f(t, y(t)) & , \ 0 \le t \le T \\ y(0) = y_0 \end{cases}$$

with continuous $f : [0,T] \times \mathbb{R} \to \mathbb{R}$ satisfying the Lipschitz condition

$$(L) \quad |f(t,y) - f(t,\tilde{y})| \le L \, |y - \tilde{y}| \qquad \forall \, y, \tilde{y} \in \mathbb{R}, \ t \in [0,T],$$

with constant $L > 0$ (independent of $t, y, \tilde{y}$).

recall: Differential equation course: $\exists ! \, y \in C^1[0,T]$ solution to (IVP). and

$$(IEq) \quad y(t) = y(\tilde{t}) + \int_{\tilde{t}}^{t} f(s, y(s)) \, ds \ , \ 0 \le \tilde{t} \le t, \ (\text{integral equation})$$

7 One-step methods

Denote $T_h = \{ 0 = t_0 < t_1 < \dots < t_n = T \}$ a grid.

aim: Compute $y_n \in \mathbb{R}$ such that $y_i \approx y(t_i) \ \forall \, 0 \le i \le n$.

approach: Discretize (IEq) with t replaced by t_{i+1} and $\tilde{t}$ replaced by t_i:

$$(OSM) \quad y_{i+1} = y_i + \underbrace{h_i \, \Phi(t_i, y_i, h_i)}_{\approx \int_{t_i}^{t_{i+1}} f(s, y(s)) \, ds} \ , \quad h_i = t_{i+1} - t_i .$$

Definition 7.1

A method of the form (OSM) is called a one-step method with increment function $\Phi : [0,T] \times \mathbb{R} \times [0,1] \to \mathbb{R}$

Remark 7.2

(i) Using piecewise linear interpolation, we reconstruct a continuous function $y_h : [0,T] \to \mathbb{R}$ with $y_h(t_i) = y_i, \ 0 \le i \le n$.

(ii) Since $y_h'(t) = \frac{y_{i+1} - y_i}{h_i}, \ t_i \le t \le t_{i+1}$, (OSM) can be seen as an

approximation to (IVP).

Example 7.3 (Explicit Euler method)

Apply (left-sided) rectangle rule:

$$\int_{t_i}^{t_{i+1}} f(s, y(s))\,ds \approx h_i \, f(t_i, y(t_i))$$

Obtain:

(EE) $\quad y_{i+1} = y_i + h_i \, f(t_i, y_i) \quad ; \quad \phi(t, y, h) = f(t, y)$

Example 7.4 (Implicit Euler method)

Apply (right-sided) rectangle rule to obtain

(IE) $\quad y_{i+1} = y_i + h_i \, f(t_i, y_{i+1}) \quad ;$

Note that y_{i+1} is given implicitly (we need to solve an equation).
The increment function exists only implicitly in general.

Example 7.5 (Crank-Nicholson method / trapezoidal method)

Apply trapezoidal rule

$$\int_{t_i}^{t_{i+1}} f(s, y(s))\,ds \approx \frac{h_i}{2} \left(f(t_i, y(t_i)) + f(t_{i+1}, y(t_{i+1})) \right)$$

to obtain

(CN) $\quad y_{i+1} = y_i + \frac{h_i}{2} \left(f(t_i, y_i) + f(t_{i+1}, y_{i+1}) \right),$

which is again implicit.

Example 7.6 (Heun's method)

Make (CN) explicit by appropriately replacing y_{i+1} in RHS of (CN) using (EE):

$$\tilde{y}_{i+1} = y_i + h_i \, f(t_i, y_i)$$

$$y_{i+1} = y_i + \frac{h_i}{2} \left(f(t_i, y_i) + f(t_{i+1}, \underbrace{y_i + h_i \, f(t_i, y_i)}_{= \tilde{y}_{i+1}}) \right)$$

increment fct $\quad \phi(t, y, h) = \frac{1}{2} \left(f(t, y) + f(t+h, y + h \, f(t, y)) \right)$

7.1 Analysis of one-step methods

For simplicity we assume $h = h_i$, $0 \le i \le n-1$.

7.1.1 Consistency

Definition 7.7

Let $y : [0,T] \to \mathbb{R}$ be a solution to (IVP), and Φ an increment function.

(i) The local truncation error is: $\tau_h(t,y) = \frac{y(t+h)-y(t)}{h} - \Phi(t, y(t), h)$.

(ii) The local error is: $e_h(t,y) = h\,\tau_h(t,y)$.

(iii) The corresponding OSM is called consistent with (IVP) if for all solutions y to (IVP):

$$\|\tau_h\|_{0,h} = \max_{t \in \mathcal{T}_h} |\tau_h(t,y)| \longrightarrow 0 \quad \text{as } h \to 0.$$

(iv) The scheme (OSM) is of consistency order p if $\exists C$ such that

$$\|\tau_h\|_{0,h} \le C h^p,$$

where C may depend on y, f and their derivatives (but not on h).

Theorem 7.8

A OSM with increment function Φ is consistent with (IVP) if and only if

$$\forall t, y : \lim_{h \to 0} \Phi(t,y,h) = f(t,y).$$

Proof: Follows from Definition 7.7 (i), (iii), and

$$\lim_{h \to 0} \frac{y(t+h)-y(t)}{h} = y'(t) \overset{(IVP)}{=} f(t, y(t)).$$

□

Example 7.9 (Explicit Euler method)

The local error is

$$e_h(t,y) = y(t+h) - y(t) - h f(t, y(t))$$

$$\overset{\text{Taylor}}{=} y(t) + y'(t)h + y''(\tilde{t})\frac{h^2}{2} - y(t) - h f(t, y(t))$$

$$t \le \tilde{t} \le t+h$$

$$\overset{(IVP)}{=} \frac{1}{2} y''(\tilde{t})\, h^2.$$

Differentiating $y'(t) = f(t, y(t))$ yields

$$y''(t) = f_t(t, y(t)) + f_y(t, y(t))\, y'(t)$$

$$\overset{(IVP)}{=} f_t(t, y(t)) + f_y(t, y(t))\, f(t, y(t))$$

Therefore,

$$|\tau_h(t,y)| = \left| \frac{e_h(t,y)}{h} \right| \le \frac{h}{2} \left(\|f_t\|_\infty + \|f_y\|_\infty \|f\|_\infty \right).$$

and (EE) is of consistency order $p = 1$ (for $f \in C^1$).

(5) 7.1.2 Stability

Definition 7.10

A OSM with increment function Φ is called zero-stable, if for $y_h, \tilde{y}_h$ defined via

$$y_{i+1} = y_i + h\Phi(t_i, y_i, h) \; ; \quad \tilde{y}_{i+1} = \tilde{y}_i + h(\Phi(t_i, \tilde{y}_i, h) + \theta_i)$$

with perturbations $\theta_i = \theta_h(t_i)$, the error satisfies

$$\| y_h - \tilde{y}_h \|_{\infty,h} \leq C\left(|y_0 - \tilde{y}_0| + \| \theta_h \|_{\infty,h} \right).$$

with a constant independent of h.

Theorem 7.11

Suppose the increment function Φ of a OSM satisfies the Lipschitz condition

$$\exists L_\Phi > 0 : \quad |\Phi(t,y,h) - \Phi(t,\tilde{y},h)| \leq L_\Phi |y - \tilde{y}| \quad \forall y, \tilde{y} \in \mathbb{R}, \; 0 \leq h \leq 1, \; 0 \leq t \leq T.$$

with L_Φ independent of h and I_h. Then the OSM is zero-stable.

proof: We verify Definition 7.10. The error $w_h = y_h - \tilde{y}_h$ satisfies

$$w_{i+1} = w_i + h\left(\Phi(t_i, y_i, h) - \Phi(t_i, \tilde{y}_i, h) - \theta_i \right)$$

Thus $|w_{i+1}| \leq |w_i| + h L_\Phi |w_i| + h|\theta_i|$

and the Theorem follows from Theorem 7.12 ∎

Theorem 7.12 (discrete Gronwall lemma)

Suppose $0 \leq w_{i+1} \leq (1 + hL) w_i + h c_i$ with $h, c_i \geq 0$, $1 + hL \geq 0$.

Then $\forall n \geq 0$:

$$w_n \leq w_0 e^{Lhn} + \left(\max_{0 \leq i \leq n-1} c_i \right) \left| \frac{e^{Lhn} - 1}{L} \right|$$

⑥ proof Set $q = 1 + hL \geq 0$. Then

$$w_n \leq q\, w_{n-1} + h\, c_{n-1}$$

$$\leq q(q\, w_{n-2} + h\, c_{n-2}) + h c_{n-1}$$

$$\leq \dots \leq q^n w_0 + h \sum_{i=0}^{n-1} q^i c_{n-i-1}$$

$$\leq q^n w_0 + \left(\max_{0 \leq i \leq n-1} c_i \right) h \sum_{i=0}^{n-1} q^i$$

$$= q^n w_0 + \left(\max_{0 \leq i \leq n-1} c_i \right) h \frac{q^n - 1}{q - 1}$$

The assertion follows from $q - 1 = hL$ and $q = 1 + hL \leq e^{hL}$ □

Example 7.13 (Explicit Euler method)

We have $\Phi(t, y) = f(t, y)$. Thus

$$|\Phi(t, y) - \Phi(t, \tilde{y})| = |f(t, y) - f(t, \tilde{y})| \overset{(L)}{\leq} L\, |y - \tilde{y}|$$

and $L_\Phi = L$, ie (EE) is zero-stable.

(7) 7.1.3 Convergence

Definition 7.14

Denote y solution to (IVP) and y_h its approximation using a OSM.

(i) The global error of the OSM is $E_h(t) := y(t) - y_h(t)$.

(ii) The scheme is called convergent if

$$\|E_h\|_{h,\infty} = \max_{t \in I_h} |y(t) - y_h(t)| \to 0 \quad \text{as } h \to 0.$$

(iii) The scheme is called convergent of order p if

$$\|E_h\|_{h,\infty} \leq C h^p$$

for some constant $C > 0$, which may depend on y, f and their derivatives (but not on h).

Theorem 7.15 (Consistency + Stability $\Rightarrow$ Convergence)

Consider a OSM that is consistent (with order p) with (IVP) and zero-stable. Then the scheme is convergent.

proof: Let y_h be generated by (OSM) and set $\tilde{y}_i = y(t_i)$.

By definition of τ_h:

$$\tilde{y}_{i+1} = \tilde{y}_i + h\left(\phi(t_i, \tilde{y}_i, h) - \tau_h(t_i, y)\right)$$

Theorem 7.11 and Definition 7.10 with $\theta_i = \theta_h(t_i) = \tau_h(t_i, y)$, implies

$$\|y_h - y\|_{\infty,h} = \|y_h - \tilde{y}_h\|_{\infty,h} \leq C \|\tau_h\|_{\infty,h}.$$

The claim follows from consistency of the scheme (with order p).

Corollary 7.16

If $f \in C^1$, then the explicit Euler method is convergent with order $p = 1$.

proof: Follows from Examples 7.9, 7.13 and Theorem 7.15 $\square$

8 Runge-Kutta methods

aim: Systematic construction of stable OSM

$$y_{i+1} = y_i + h\,\Phi(t_i, y_i, h)$$

with high consistency order (thus high convergence order, Theorem 7.15)

approach: Use high order numerical integration rules to approximate (IEq):

$$(RK.1)\qquad y_{i+1} = y_i + h \underbrace{\sum_{j=1}^{s} \beta_j f(t_i + \gamma_j h, g_j)}_{\approx \int_{t_i}^{t_{i+1}} f(s, y(s))\,ds \;=\; h\int_0^1 f(t_i+\gamma h, y(t_i+\gamma h))\,d\gamma}$$

$$s = t_i + \gamma h$$

β_j quadrature weights, γ_j quadrature points; $g_j \approx y(t_i + \gamma_j h)$

To obtain (unknown) g_j consider a similar scheme

$$(RK.2)\qquad g_j = y_i + h \sum_{k=1}^{s} \alpha_{jk} f(t_i + \gamma_k h, g_k)\,, \quad 1 \le j \le s.$$

with quadrature weights α_{jk}.

Definition 8.1

(i) (RK.1)-(RK.2) is called a Runge-Kutta scheme with s stages.

(ii) If $\alpha_{jk} = 0$ for $k \ge j$, the scheme is called explicit (ERK); otherwise implicit (IRK).

Remark 8.2 (Butcher tableau)

(RK.1)-(RK.2) is already determined by $\beta_j, \gamma_j, \alpha_{jk}$, $j = 1,\dots,s$, $k = 1,\dots,s$

The Butcher tableau is

$$\begin{array}{c|ccc} \gamma_1 & \alpha_{11} & \cdots & \alpha_{1s} \\ \gamma_2 & \vdots & \ddots & \vdots \\ \gamma_s & \alpha_{s1} & \cdots & \alpha_{ss} \\ \hline & \beta_1 & \cdots & \beta_s \end{array} \quad \text{or} \quad \begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

with $A_{jk} = \alpha_{jk}$

$c_j = \gamma_j$, $b_j = \beta_j$.

Example 8.3 (Explicit Euler Method)

Writing

$$g_1 = y_i + h\,(0 \cdot f(t_i + 0\,h,\ g_1))$$

$$y_{i+1} = y_i + h\,(1 \cdot f(t_i + 0\,h,\ g_1))$$

we see that (EE) is an ERK scheme with $s=1$ stage.

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

general plan:

(i) Show that (RK.1)-(RK.2) can be executed, i.e. (RK.2) has a solution.
(consequence (RK.1)-(RK.2) is a OSM: $y_{i+1} = y_i + h\,\phi(t_i, y_i, h)$)

(ii) Show stability (cf. Definition 7.10, Theorem 7.11)

(iii) Show consistency (with order p)

(iv) Conclude convergence (with order p) (Theorem 7.15).

8.1 Explicit Runge Kutta Methods

Since $\alpha_{jk} = 0$ for $k \geq j$, we obtain

$$g_j = y_i + h \sum_{k=1}^{j-1} \alpha_{jk}\, f(t_i + \gamma_k h,\ g_k),$$

i.e. (RK.1)-(RK.2) can be executed. The increment function is

$$\phi(t_i, y_i, h) = \sum_{j=1}^{s} \beta_j\, f(t_i + \gamma_j h,\ g_j).$$

Theorem 8.4 (Zero-stability)

ERK are zero-stable.

proof: In view of Theorem 7.11, we need to show that ϕ is Lipschitz continuous in y.
Since the composition of Lipschitz functions is Lipschitz, it suffices to show that g_j depends Lipschitz continuously on y_i.

8 Runge-Kutta methods

aim: Systematic construction of stable OSM

$$y_{i+1} = y_i + h\,\Phi(t_i, y_i, h)$$

with high consistency order (thus high convergence order, Theorem 7.15)

approach: Use high order numerical integration rules to approximate (IEq):

$$\text{(RK.1)}\quad y_{i+1} = y_i + h \underbrace{\sum_{j=1}^{s} \beta_j f(t_i + \gamma_j h, g_j)}_{\approx \int_{t_i}^{t_{i+1}} f(s, y(s))\,ds \;=\; h\int_0^1 f(t_i+\gamma h, y(t_i+\gamma h))\,d\gamma}$$

$$s = t_i + \gamma h$$

β_j quadrature weights, γ_j quadrature points; $g_j \approx y(t_i + \gamma_j h)$

To obtain (unknown) g_j consider a similar scheme

$$\text{(RK.2)}\quad g_j = y_i + h \sum_{k=1}^{s} \alpha_{jk} f(t_i + \gamma_k h, g_k), \quad 1 \le j \le s.$$

with quadrature weights α_{jk}.

Definition 8.1

(i) (RK.1)-(RK.2) is called a Runge-Kutta scheme with s stages.

(ii) If $\alpha_{jk} = 0$ for $k \ge j$, the scheme is called explicit (ERK), otherwise implicit (IRK).

Remark 8.2 (Butcher tableau)

(RK.1)+(RK.2) is already determined by $\beta_j, \gamma_j, \alpha_{jk}$, $j = 1,\dots,s$, $k = 1,\dots,s$

The Butcher tableau is

γ_1	α_{11}	⋯	α_{1s}
γ_2	⋮	⋱	⋮
⋮	⋮	⋱	⋮
γ_s	α_{s1}	⋯	α_{ss}
	β_1	⋯	β_s

or

c	A
	b^T

with $A_{jk} = \alpha_{jk}$, $c_j = \gamma_j$, $b_j = \beta_j$.

Example 8.3 (Explicit Euler Method)

Writing

$$g_1 = y_i + h\,(0 \cdot f(t_i + 0h, g_1))$$

$$y_{i+1} = y_i + h\,(1 \cdot f(t_i + 0h, g_1))$$

we see that (EE) is an ERK scheme with $s=1$ stage.

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

general plan:

(i) Show that (RK.1)-(RK.2) can be executed, i.e. (RK.2) has a solution.
(consequence (RK.1)-(RK.2) is a OSM: $y_{i+1} = y_i + h\,\phi(t_i, y_i, h)$)

(ii) Show stability (cf. Definition 7.10, Theorem 7.11)

(iii) Show consistency (with order p)

(iv) Conclude convergence (with order p) (Theorem 7.15).

8.1 Explicit Runge Kutta Methods

Since $\alpha_{jk} = 0$ for $k \geq j$, we obtain

$$g_j = y_i + h \sum_{k=1}^{j-1} \alpha_{jk}\, f(t_i + \gamma_k h, g_k),$$

i.e. (RK.1)-(RK.2) can be executed. The increment function is

$$\phi(t_i, y_i, h) = \sum_{j=1}^{s} \beta_j\, f(t_i + \gamma_j h, g_j).$$

Theorem 8.4 (Zero-stability)

ERK are zero-stable.

proof: In view of Theorem 7.11, we need to show that ϕ is Lipschitz continuous in y.
Since the composition of Lipschitz functions is Lipschitz, it suffices to show that g_j depends Lipschitz continuously on y_i.

(3) We argue by mathematical induction:

$j=1$: $|g_1(y) - g_1(\tilde{y})| = |y - \tilde{y}|$,

i.e. g_1 is Lipschitz with L-constant $L_1 = 1$.

hypothesis: Suppose (*) $|g_j(y) - g_j(\tilde{y})| \le L_j |y - \tilde{y}| \quad \forall j \le k$.

Then $|g_{k+1}(y) - g_{k+1}(\tilde{y})|$

$$\le |y - \tilde{y}| + h \sum_{j=1}^{k} |\alpha_{k+1,j}| \underbrace{\left| f(t_i + \gamma_j h, g_j(y)) - f(t_i + \gamma_j h, g_j(\tilde{y})) \right|}_{\overset{(L)}{\le} L |g_j(y) - g_j(\tilde{y})| \overset{(*)}{\le} L L_j |y - \tilde{y}|}$$

$$\le \left(1 + hL \sum_{j=1}^{k} |\alpha_{k+1,j}| L_j\right) |y - \tilde{y}|,$$

which proves the claim by induction. □.

Next, we consider consistency.

Lemma 8.5 (necessary condition)

ERK is consistent, if and only if

(01) $\sum_{j=1}^{s} \beta_j = 1$

proof: We have that $\lim_{h \to 0} g_j(y) = y$. Thus

$$\lim_{h \to 0} \phi(t, y, h) = \lim_{h \to 0} \sum_{j=1}^{s} \beta_j f(t + \gamma_j h, g_j) \underset{f \text{ continuous}}{=} \left(\sum_{j=1}^{s} \beta_j\right) f(t, y)$$

The claim follows from Theorem 7.8 □

Example 8.6 (ERK with $s=1$ stage)

Consider $g_1 = y_i$

$$y_{i+1} = y_i + h\, f(t_i + \gamma_1 h,\ \underbrace{g_1}_{=y_i})$$

with free parameter γ_1. Suppose $f \in C^1$.

Taylor yields for the local error:

$$e_h(t,y) = y(t+h) - y(t) - h\, f(t+\gamma_1 h,\ y(t))$$

$$\overset{\text{Taylor}}{=} y(t) + y'(t)h + y''(t)\frac{h^2}{2} - y(t) - h\Big(f(t,y(t)) + f_t(t,y(t))\gamma_1 h\Big) + O(h^3)$$

$$\overset{\substack{y' \overset{\text{IVP}}{=} f \\ y'' = f_t + f_y f}}{=} h^2\Big(\frac{1}{2}(f_t + f_y f) - \gamma_1 f_t\Big) + O(h^3)$$

(see Example 7.9)

Since $\tau_h(t,y) = e_h(t,y)/h$, we conclude

(i) consistency order is $p=1$ for all γ_1.

(ii) $p \geq 2$ cannot be achieved with $s=1$ stage.

(iii) $\gamma_1 = 0$ yields (EE).

Example 8.7 (ERK with $s=2$ stages)

γ_1	0	
γ_2	α_{21}	0
	β_1	β_2

with $\beta_1 + \beta_2 = 1$.

(Lengthy) Taylor yields as in Example 8.6, for $f \in C^2$,

$$e_h(t,y) = h^2\Big(\Big(\frac{1}{2} - \beta_1\gamma_1 - \beta_2\gamma_2\Big)f_t + \Big(\frac{1}{2} - \beta_2\alpha_{21}\Big) f_y f\Big) + O(h^3)$$

$\Rightarrow$

(i) If $\beta_1\gamma_1 = \frac{1}{2}$ and $\beta_2\alpha_{21} = \frac{1}{2}$, then consistency order $p \geq 2$.

(ii) $p \geq 3$ is not possible with $s=2$ stages

(iii)

0	0	
$1/2$	$1/2$	
	0	1

(improved Euler) and

0	0	
1	1	0
	$1/2$	$1/2$

(Heun) have $p \geq 2$, and are convergent.

(5)

Without proof, we summarize, see Table 6.1.1, p. 405, Steward.

(notation: $e = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^s$, $u \circ v := \begin{pmatrix} u_1 v_1 \\ \vdots \\ u_s v_s \end{pmatrix}$ for $u, v \in \mathbb{R}^s$)

Theorem 8.8 (Consistency conditions)

(i) Let (01) hold. If $f \in C^1$, then $p \geq 1$ for an ERK with s stages.

(ii) In addition to (i), let $f \in C^2$ and $b^T c = \frac{1}{2}$ and $Ae = c$, then $p \geq 2$.

(iii) In addition to (i)-(ii), let $f \in C^3$ and $b^T A c = \frac{1}{6}$ and $b^T (c \circ c) = \frac{1}{3}$, then $p \geq 3$

(iv) In addition to (i)-(iii), let $f \in C^4$ and

$$b^T A (c \circ c) = \frac{1}{12}, \quad (b \circ c)^T A c = \frac{1}{8}, \quad b^T A c^2 = \frac{1}{24}, \quad b^T (c \circ c \circ c) = \frac{1}{24}, \text{ then } p \geq 4.$$

Remark 8.9

The order of an s-stage ERK cannot be greater than s. Also there is no ERK with order s if $s \geq 5$. [Quarteroni et al, p. 529].

Remark 8.10 (Alternative form)

Denoting $k_j = f(t_i + \gamma_j h, g_j)$ we can write (RK1)-(RK2) equivalently as

$$y_{i+1} = y_i + h \sum_{j=1}^{s} \beta_j k_j$$

$$k_j = f\left(t_i + \gamma_j h, \; y_i + h \sum_{\ell=1}^{s} \alpha_{j\ell} k_\ell\right).$$

Remark 8.11 (Implicit RK)

(i) If $hL \|A\|_\infty < 1$, then (IRK can be executed. (see Sec 6.1.4.1 in [Steward]).

(ii) If $hL \|A\|_\infty < 1$, one can show that IRK is stable, and a QSM.

(iii) Theorem 8.8 applies to IRK as well.

(iv) If f is smooth, $hL\|A\|_\infty < 1$ and

$$\sum_{k=1}^{s} \alpha_{jk} \gamma_k^\ell = \gamma_j^{\ell+1}/(\ell+1), \quad \ell = 0, \dots, q$$

$$\sum_{j=1}^{s} \beta_j \gamma_j^\ell = \frac{1}{\ell+1}, \quad \ell = 0, \dots, r$$

with $q = p-2$ and $r = p-1$, then the consistency order is $\hat{p} \geq p$.

(v) If (iv) holds with ($r = s$, $s \leq p \leq 2s$) or ($r = s-1$, $r \leq p \leq 2s-1$ and $\gamma_s = 1$, $\alpha_{js} = 0$) then consistency order $\geq p$.

↳ "Good" quadrature rules (γ_j, β_j), (γ_i, α_{ji}) yield high order.

↳ the maximal order is $p = 2s$ (see Gauss quadrature)

9 Absolute stability

While for zero-stability, we kept $[0,T]$ fixed and considered the behavior of y_h as $h \to 0$, we now turn to the behavior of y_h for fixed h but $t = t_n \to \infty$.

Consider the test problem

$$(TP) \quad \begin{cases} y'(t) = \lambda y(t) \\ y(0) = 1 \end{cases}, \quad t > 0$$

$\lambda \in \mathbb{C}$, with solution $y(t) = e^{\lambda t}$.

If $\mathrm{Re}(\lambda) < 0$, then $\lim_{t \to \infty} |y(t)| = 0$. ($\to$ asymptotic stability).

Definition 9.1

(i) A numerical method for approximating (TP) is absolutely stable if

$$(AS) \qquad \lim_{t \to \infty} |y_h(t)| = 0.$$

(ii) The region of absolute stability is $\mathcal{A} = \{ z = h\lambda \in \mathbb{C} : (AS) \text{ holds} \}$.

Example 9.2 (Explicit Euler)

(EE) applied to (TP) yields

$$y_{n+1} = (1 + h\lambda) y_n = \dots = (1 + h\lambda)^{n+1} \underbrace{y_0}_{=1}, \quad n \geq 0.$$

Hence, (AS) holds if $|1 + h\lambda| < 1$, i.e. $h\lambda$ is in the unit circle centred at -1 in the complex plane.

Setting $\lambda = x + iy$, $|1 + h\lambda|^2 = (1 + hx)^2 + (hy)^2 = 1 + 2hx + h^2(x^2 + y^2)$

Hence, $|1 + h\lambda| < 1$ is equivalent to $2x + h|\lambda|^2 < 0$, i.e. $h < \frac{-2\,\mathrm{Re}(\lambda)}{|\lambda|^2}$

For very large λ (negative), this puts a strong condition on h.

(EE) is conditionally absolutely stable.

Definition 9.3

A method is called A-stable if $\mathcal{A} \cap \mathbb{C}^- = \mathbb{C}^-$ with left half plane

$$\mathbb{C}^- = \{ z \in \mathbb{C} : \mathrm{Re}(z) < 0 \}.$$

Example 9.4 (Implicit Euler)

(IE) applied to (TP) yields

$$y_{n+1} = y_n + h\lambda y_{n+1}, \text{ i.e. } y_{n+1} = \frac{1}{1-h\lambda} y_n = \ldots = \frac{1}{(1-h\lambda)^{n+1}}.$$

The condition $1 < |1-h\lambda|^2 = 1 - 2hx + h^2(x^2+y^2)$ is equivalent to $\frac{2x}{x^2+y^2} < h$. ($h\lambda$ should be outside the unit circle centred at 1 in the complex plane). Hence if $x = \mathrm{Re}(\lambda) < 0$, no extra condition is put on h. Thus, (IE) is A-stable (unconditionally).

Remark 9.5

There exist implicit unstable and implicit conditionally stable schemes.

There are no explicit unconditionally absolutely stable schemes.

10 Stiff problems

Consider non-homogeneous linear system of ODEs with constant coefficients

$$(S) \qquad y'(t) = A\,y(t) + b(t) \quad , \quad A \in \mathbb{R}^{n\times n} \quad , \quad b(t) \in \mathbb{R}^n$$

Assumption: A has n distinct eigenvalues λ_j, $j = 1, \dots, n$.

ODE course: $y(t) = \sum_{j=1}^{n} C_j e^{\lambda_j t} v_j + \psi(t)$

with constants C_j, $\{v_j\}_{j=1}^n$ basis of eigenvectors of A.

Assumption: $\mathrm{Re}(\lambda_j) < 0$.

Hence $\lim_{t\to\infty} \underbrace{y(t) - \psi(t)}_{= y_{hom},\ \text{transient solution}} = 0$, $\psi(t)$ steady-state solution.

"Paradox" If the region of absolute stability of a numerical scheme is bounded, and the modulus of $|\lambda_j|$ is large, we need very small step-size h for a meaningful numerical solution, while the solution component $C_j e^{\lambda_j t} v_j$ is close to zero.

We say that a linear system of ODEs (S) is stiff if $\mathrm{Re}(\lambda_j) < 0$ and σ/τ is very large, where

$$\sigma \le \mathrm{Re}(\lambda_j) \le \tau < 0 \qquad \forall j = 1, \dots, n.$$

Note if $\tau \approx 0$ and $|\sigma|$ moderate, the "paradoxon" does not apply, i.e. stiffness is also asking that $|\sigma|$ is very large, see [Definition 11.14, Quarteroni] for a more detailed definition.

Remark 10.1

Gauss-Legendre IRK methods are A-stable for any number of stages [Iserles, p. 63] / [Stewart, p. 416].