Computational Mathematics 1

Lecture notes, University (of Technology) Graz
based on the lecture by Tobias Breiten

Lukas Prokop

December 25, 2018

Contents

Course

↓ *This lecture took place on 2018/10/02.*

- Homepage at uni-graz.at
- Contents:
  1. Linear equation system
  2. Numerical error analysis
  3. Curve fitting (dt. "Lineare Ausgleichsrechnung")
  4. Non-linear equations
  5. Interpolation
  6. Numerical integration
- Literature:
  1. Deuflhard, Hohmann: Numerische Mathematik 1
  2. Schwarz, Köckler: Numerische Mathematik

1 What is Computational Mathematics?

Computational Mathematics as branch of mathematics considers the construction and analysis of algorithms for continuous mathematical problems. More specifically, based on a model and input data, we construct specific output data of interest. Formally, we associate a model using function $f : X \to Y$ where X is the set of input data and Y the set of output data. The problem is to find output $f(x) \in Y$ for given $x \in X$.

Examples:

- Addition of two numbers: $X = \mathbb{R}^2, Y = \mathbb{R}, f : (x_1, x_2) \mapsto x_1 + x_2$
- Solution of a linear equation system: $Ax = b$

$$X = \mathbb{R}^{n \times n} \times \mathbb{R}^n, Y = \mathbb{R}^n, f : (A, b) \mapsto A^{-1}b$$

An algorithm is defined as unambiguous sequence of steps to solve a given problem. A problem is characterized as

- the input data required for computation
- the output data that represent the solution of the algorithm
- the description of the steps to be done, including auxiliary values

We require, the algorithm is

executable by a machine

terminating after a finite number of steps

deterministic as the same input always leads to the same output

Furthermore we analyze the algorithm in terms of

stability small errors in the input lead to small errors in the output

precision output data should solve the problem with maximum accuracy

efficiency large problems can be solved in practical time

Consider the linear equation system

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b \\ \vdots &= \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b \end{aligned}$$

The same can be specified in compact notation like $Ax = b$ with $A \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n$.

Theorem 1.1 (Cramer's Rule). *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 $Ax = b$. We can determine x using "Cramer's Rule":*

$$x_i = \frac{\det(A_i)}{\det(A)} \text{ where } A_i = \begin{bmatrix} a_{11} & \dots & a_{1,i-1} & b_1 & a_{1,i+1} & \dots & a_{1n} \\ \vdots & & & & & & \vdots \\ a_{n1} & \dots & a_{n,i-1} & b_n & a_{n,i+1} & \dots & a_{nn} \end{bmatrix}$$

Remark (Problem). *Computation using $\det(A) = \sum_{\sigma\in S_n} \operatorname{sign}(\sigma) a_{1,\sigma(1)} \dots a_{n,\sigma(n)}$ requires $n \cdot n!$ operations.*

1.1 Solving linear eq. system by forward/backwards substitution

Consider the special case of a linear equation system of structure

$$\begin{array}{rrrrl} \gamma_{11}x_1 & +\gamma_{12}x_2 & +\dots & +\gamma_{1n}x_n & = z_1 \\ & \gamma_{22}x_2 & +\dots & +\gamma_{2n}x_n & = z_2 \\ & & \ddots & & =\vdots \\ & & & \gamma_{nn}x_n & = z_n \end{array}$$

and accordingly, $Rx = z$ with an upper triangular matrix, hence $\gamma_{ij} = 0$ for $i > j$. We retrieve x by recursive solution (by so-called "backwards substitution") starting with row n:

$$\begin{aligned} x_n &= \frac{z_n}{\gamma_{nn}} \text{ if } \gamma_{nn} \neq 0 \\ x_{n-1} &= \frac{z_{n-1} - \gamma_{n-1,n}x_n}{\gamma_{n-1,n-1}} && \text{if } \gamma_{n-1,n-1} \neq 0 \\ x_1 &= \frac{z_1 - \gamma_{12}x_2 - \dots - \gamma_{1n}x_n}{\gamma_{11}} && \text{if } \gamma_{11} \neq 0 \end{aligned}$$

Obviously, it holds that

$$\det(R) = \gamma_{11} \dots \gamma_{nn} \neq 0 \iff \gamma_{ii} \neq 0 \qquad \forall i = 1, \dots, n$$

The algorithm is applicable (just like Cramer's rule), if $\det(R) \neq 0$. Then the existence of a solution is guaranteed and this solution is unique.

Remark (Computing time).

1. *for the i-th row, we need $(n-i)$ additions, $(n-i)$ multiplications and one division*
2. *in total for the rows n to 1,*

$$\sum_{i=1}^{n}(i-1) = \frac{n(n-1)}{2} \doteq \frac{n^2}{2}$$

Multiplication and the same amount of additions. The notation $\doteq$ is used to denote equivalence up to terms of same degree. Analogously, the equation system $Lx = z$ can be solved with a lower triangular matrix L by forward substitution starting with the first row.

1.2 Gaussian elimination process

Basic idea:

Beginning with a linear equation system of structure,

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

we apply equivalence transformations to end up with an upper triangular matrix.

As a first step, we eliminate variable x_1 of rows 2 to n.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a'_{22}x_2 + \cdots + a'_{2n}x_n &= b_1 \\ \vdots &= \vdots \\ a'_{12}x_2 + \cdots + a'_{nn}x_n &= b'_n \end{aligned} \tag{2}$$

So we apply this elimination step ($r_i := \frac{r_i}{a_{i1}} - \frac{r_1}{a_{11}}$ where r_i denotes the i-th row) to the last $n-1$ rows and determine the triangular matrix recursively this way.

Consider the transformation of (1) to (2). Let $a_{11} \neq 0$ and we recognize $\text{row}_i := \text{row}_i - l_{i_1} \cdot \text{row}_1$. Formally,

$$\underbrace{(a_{i1} - l_{i1}a_{11})x_1}_{=0} + \underbrace{(a_{i2} - l_{i1}a_{12})}_{a'_{i2}} x_2 + \cdots + \underbrace{(a_{in} - l_{i1}a_{1n})}_{a'_{in}} x_n = \underbrace{b_i - l_{in}b_1}_{b'_i}$$

$$\implies l_{in} = \frac{a_{i1}}{a_{11}}$$

hence, the first elimination step is applicable if $a_{11} \neq 0$ is given. If we apply this step, we retrieve a sequence of matrices:

$$A = A^{(1)} \to A^{(2)} \to \ldots \to A^{(n)} = R$$

where

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \ldots & & & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \ldots & & & a_{2n}^{(2)} \\ & & \ddots & & & \vdots \\ & & & a_{kk}^{(k)} & \ldots & a_{kn}^{(k)} \\ & & & \vdots & & \vdots \\ & & & a_{nk}^{(k)} & \ldots & a_{nn}^{(k)} \end{bmatrix}$$

with a remaining matrix of dimension $(n-k+1) \times (n-k+1)$. For every remaining matrix we can apply the elimination step

$$\begin{aligned} l_{ik} &= \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} && \text{for } i = k+1, \ldots, n \\ a_{ij}^{(k+1)} &= a_{ij}^{(k)} - l_{ik}a_{kj}^{(k)} && \text{for } i, j = k+1, \ldots, n \\ b_i^{(k+1)} &= b_i^{(k)} - l_{ik}b_k^{(k)} && \text{for } i = k+1, \ldots, n \end{aligned}$$

under the assumption that $a_{kk}^{(k)} \neq 0$ (pivot element).

Remark. *Every elimination step is a linear operation of the rows of A. Hence it is representable by left multiplication with $L_k \in \mathbb{R}^{n\times n}$ according to $A^{k+1} = L_k A^{(k)}$ and $b^{(k+1)} = L_k b^{(k)}$.*

The following matrix is a Frobenius matrix, *which has an interesting property regarding its inverse:*

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

and

$$L = L_1^{-1} \quad \dots \quad L_{n-1}^{-1} = \begin{pmatrix} 1 & & & & 0 \\ l_{21} & \ddots & & & \\ l_{31} & l_{32} & \ddots & & \\ \vdots & \vdots & & & \\ l_{n1} & \dots & & l_{n,n-1} & 1 \end{pmatrix}$$

$$L^{-1} = L_{n-1} \dots L_1$$

This way, we retrieve an equivalent linear equation system to $Rx = z$, $R = L^{-1}A$, $z = L^{-1}b$. The representation $A = LR$ is called *LR decomposition of A*. If they exists, L and R are unambiguously determined.

$$A = L_1 R_1 = L_2 R_2 \implies R_1 = L_1^{-1} L_2 R_2 \implies R_1 R_2^{-1} = L_1^{-1} L_2$$

$$\implies \text{diagonal representation of } L_1^{-1} L_2 \dots, R_1 R_2^{-1}$$

Algorithm 1 (Gaussian elimination)**.**

1. *$A = LR$ with R as upper triangular matrix, L as lower triangular matrix*
2. *Solve $Lz = b$ by forward substitution*
3. *Solve $Rx = z$ by backwards substitution*

The entries $l_{ik} \neq 1$ can be stored in the remaining null-entries of the matrix $A^{(k)}$. Thus the total memory requirements are bounded by $n(n+1)$.

Remark (Computing time)**.**

- $\sum_{k=1}^{n-1} k^2 \doteq \frac{n^3}{3}$ *for step 1*
- $\sum_{k=1}^{n-1} k \doteq \frac{n^2}{2}$ *for step 2 and 3*

1.3 Pivot strategies

Problem: The algorithm above does not work for the simple example:

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \det(A) = -1 \neq 0$$

We *need* row exchanges. We can evaluate, that no LU-decomposition exists.

But exchange of the rows gives

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \rightsquigarrow \hat{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I = LU$$

with $L = U = I$.

Example 1.2. *Consider the equation system*

$$\begin{aligned} 1 \cdot 10^{-4} x_1 + 1 \cdot x_2 &= 1 \\ 1 \cdot x_1 + 1 \cdot x_2 &= 2 \end{aligned}$$

Assumption: *Compute up to three decimal points. Then for the "accurate" solution, we get* $x_1 = 1$ *and* $x_2 = 1$.

Using the Gaussian elimination process, we get

$$l_{21} = \frac{a_{21}}{a_{11}} = 10^4$$

$$\implies 0 \cdot x_1 + (1 - 10^4 \cdot 1) x_2 = 2 \cdot 10^4 \cdot 1$$

Therefore we get the triangular system

$$10^{-4} x_1 + 1 \cdot x_2 = 1$$

$$-10^4 x_2 = -10^4$$

By this, we get an approximation $x_2 = 1$ and $x_1 = 0$.

In contrast, if we exchange the two rows,

$$\begin{aligned} 1 \cdot x_1 + 1 \cdot x_2 &= 2 \\ 10^{-4} x_1 + 1 \cdot x_2 &= 1 \end{aligned}$$

then $\tilde{l}_2 1 = \frac{10^4}{1}$ follows and therefore the triangular system

$$\begin{aligned} 1 \cdot x_1 + 1 \cdot x_2 &= 2 \\ 1 \cdot x_2 &= 1 \end{aligned}$$

with the "correct" solution is $x_2 = 1$ and $x_1 = 1$.

Our exchange of the rows has given us $\left|\tilde{l}_{21}\right| < 1$ and $|\tilde{a}_{11}| \geq |\tilde{a}_{21}|$.

↓ *This lecture took place on 2018/10/04.*

The new pivot element $\tilde{a}_{11}$ is the largest absolute value in the first column. Thus, we apply the column pivot strategy: In every step, choose the row with the largest, absolute value in the pivot column.

Algorithm 2 (Gaussian elimination with column pivot strategy)**.**

1. *In step* $A^{(k)} \to A^{(k+1)}$*, choose some* $p \in \{k, \dots, n\}$ *such that* $\left|a_{p_k}^{(k)}\right| \geq \left|a_{jk}^{(k)}\right|$ *for* $j = k, \dots, n$
2. *Exchange rows* p *and* k

$$A^{(k)} \rightsquigarrow \tilde{A}^{(k)} \text{ with } \tilde{a}_{ij}^{(k)} = \begin{cases} a_{kj}^{(k)} & i = p \\ a_{pj}^{(k)} & i = k \\ a_{ij}^{(k)} & \text{else} \end{cases}$$

3. *Apply the elimination step* $\tilde{A}^{(k)} \to \tilde{A}^{(k+1)}$

Remark 1.3. *We can apply the row pivot strategy with column exchange instead of the column pivot strategy with row exchange.*

We use permutation matrices $P \in \mathbb{R}^{n \times n}$ to analyze the column pivot search. By permutation $\pi \in S_n$, we associate the *permutation matrix*

$$P_\pi = [e_{\pi(1)}, e_{\pi(2)}, \dots, e_{\pi(n)}]$$

where e_j denotes the j-th unit vector in $\mathbb{R}^n$. Row and column exchanges of a matrix A can followingly be represented as

$$\pi_z : A \mapsto P_\pi A, \qquad \pi_S : A \mapsto A P_\pi$$

Remark. *It holds that* $\det(P_\pi) = \operatorname{sign}(\pi) \in \{\pm 1\}$ *and* $P_\pi^{-1} = P_\pi^T$*. [Consider* $P_\pi = e_{\pi(1)} \cdot e_1^T + e_{\pi(2)} e_2^T + \dots + e_{\pi(n)} e_n^T$*.]*

$$\begin{aligned} P_\pi^T &= e_1 e_{\pi(1)}^T + e_2 e_{\pi(2)}^T + \dots + e_n e_{\pi(n)}^T \\ P_\pi P_\pi^T &= e_{\pi(1)} e_{\pi(1)}^T + e_{\pi(2)} e_{\pi(2)}^T + \dots + e_{\pi(n)} e_{\pi(n)}^T \end{aligned}$$

Theorem 1.4. *Let* $A \in \mathbb{R}^{n \times n}$ *and* $\det(A) \neq 0$*. Then there exists some permutation matrix* $P \in \mathbb{R}^{n \times n}$ *such that*

$$PA = LU$$

with lower and upper triangular matrices $L, U \in \mathbb{R}^{n \times n}$*. Furthermore* P *can be chosen such that* $|L| := \max_{i,j} \left|l_{ij}\right| \leq 1$*.*

Proof. We use Algorithm 2. Because $\det(A) \neq 0$, $\max_i |a_{i1}| > 0$ and therefore there exists some transposition $\tau_1 \in S_n$ such that for $A^{(1)} = P_{\tau_1} A$,

$$\max_i \left|a_{i1}^{(i)}\right| = \left|a_{11}^{(1)}\right| > 0$$

We can apply the elimination step and retrieve a matrix of structure

$$A^{(2)} = L_1 A^{(1)} = L_1 P_{\tau_1} A = \left[\begin{array}{ccccc} a_{11}^{(1)} & x & x & \dots & x \\ \hline 0 & & & & \\ \vdots & & & B^{(2)} & \\ 0 & & & & \end{array}\right]$$

Especially, it holds that $\max_{i,j} \left| l_{i,j}^{(1)} \right| \leq 1$ and $\det(L_1) = 1$. And

$$0 \neq \operatorname{sign}(\tau_1) \cdot \det(A) = \det(A^{(2)}) = a_{11}^{(1)} \det(B^{(2)})$$

and thus $\det(B^{(2)}) \neq 0$. Inductively, we get $U = A^{(n)} = L_{n-1} P_{\tau_{n-1}} \dots L_1 P_{\tau_1} A$ with $|L_k| \leq 1$ and transposition of two numbers $\geq k$. If $\pi \in S_n$ exchanges only numbers $\geq k+1$, then

$$L_k = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & -l_{k+1,k} & 1 & & \\ 0 & \vdots & & \ddots & \\ & l_{n,k} & & 0 & 1 \end{bmatrix}, \hat{L}_k = P_\pi L_k P_\pi^{-1} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & -l_{\pi(k+1),k} & 1 & & \\ 0 & \vdots & & \ddots & \\ & l_{\pi(n),k} & & 0 & 1 \end{bmatrix}$$

By insertion of the identities $P_{\tau_k}^{-1} P_{\tau_k}$ we get,

$$\begin{aligned} &= L_{n-1} P_{\tau_{n-1}} L_{n-2} P_{\tau_{n-1}}^{-1} P_{\tau_{n-1}} P_{\tau_{n-2}} L_{n-3} P_{\tau_{n-3}} \dots L_1 P_{\tau_1} A \\ &= \underbrace{(L_{n-1})}_{\hat{L}_{n-1}} (P_{\tau_{n-1}} L_{n-2} P_{\tau_{n-1}}^{-1}) (P_{\tau_{n-1}} P_{\tau_{n-2}} L_{n-3} P_{\tau_{n-2}}^{-1} P_{\pi_{n-1}}^{-1} P_{\tau_{n-1}} P_{\tau_{n-2}} P_{\tau_{n-3}} \dots L_1 P_{\tau_n} A) \\ &= \hat{L}_{n-1} \hat{L}_{n-2} \hat{L}_{n-3} \dots \hat{L}_1 P_{\pi_0} A \end{aligned}$$

with $\hat{L}_k = P_{\pi_k} L_k P_{\pi_k}^{-1}$, $\pi_{n-1} = \mathrm{id}$, $\pi_k = \tau_{n-1} \dots \tau_{k+1}$, $k = 0, \dots, n-2$. Be aware that π_k only exchanges numbers $\geq k+1$ and the matrices $\hat{L}_k$ are therefore of structure previously mentioned. As a consequence, we created the decomposition $P_{\pi_0} A = LU$ with

$$L := \hat{L}_1^{-1} \dots \hat{L}_{n-1}^{-1}, \quad L = \begin{bmatrix} 1 & & & & \\ l_{\pi_1(2),1} & 1 & & & \\ l_{\pi_2(3),1} & l_{\pi_2(3)} & \ddots & & \\ \vdots & \vdots & & \ddots & \\ l_{\pi_1(n),1} & \dots & & l_{\pi_{n-1}(n),n-1} & 1 \end{bmatrix}$$

Especially, $|L| = 1$. □

Remark 1.5. *Recognize that the method mentioned in the proof can be used to compute the determinant of A.*

$$\det(A) = \det(P) \cdot \det(LU) = \operatorname{sign}(\pi_0) \cdot \gamma_{11} \dots \gamma_{nn}$$

1.4 Cholesky process for symmetric positive definite matrices

↓ *This lecture took place on 2018/10/09.*

In the following, let $A = A^T > 0$ be symmetric positive definite.

Remark. *By symmetric positive definite property,*

1. $\langle x, Ax \rangle > 0 \forall x \neq 0$
2. *there exists an orthonormal basis* $q_1, \dots, q_n$ *of eigenvectors such that* $Q^T Q = I$

Theorem 1.6. *For every symmetric positive definite matrix* $A \in \mathbb{R}^{n \times n}$ *it holds that*

1. A *is invertible*
2. $a_{ii} > 0$ *for* $i = 1, \dots, n$
3. $\max_{i,j=1,\dots,m} |a_{ij}| = \max_{i=1,\dots,n} a_{ii}$
4. *Gaussian elimination without pivot search gives a symmetric positive definite remainder matrix in every step*

Proof. 1. follows by $\langle x, Ax \rangle > 0 \forall x \neq 0$. Assume A is not invertible, then $\exists x \in \text{kernel}(A) \implies \langle x, Ax \rangle = \langle x, 0 \rangle = 0$ which contradicts with A being symmetric positive definite.

2. also follows by $\langle x, Ax \rangle > 0$ with choice $x = e_i$ for $i = 1, \dots, n$ because $\langle e_i, Ae_i \rangle = a_{ii}$
3. Left as an exercise
4. We denote $A = A^{(1)}$ according to

$$A^{(1)} = \begin{bmatrix} a_{11} & z^T \\ z & B^{(1)} \end{bmatrix}$$

with $z = [a_{21}, \dots, a_{11}]^T$. After the first eliminiation step,

$$A^{(2)} = L_1 A^{(1)} = \begin{bmatrix} a_{11}^{(1)} & z^T \\ 0 & \\ \vdots & B^{(2)} \\ 0 & \end{bmatrix}$$

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

Right-side multiplication with L_1^T gives

$$L_1 A^{(1)} L_1^T = \begin{bmatrix} a_{11}^{(1)} & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & B^{(2)} & \\ 0 & & & \end{bmatrix}$$

Furthermore with $A^{(1)} = A > 0$ it also holds that $L_1 A^{(1)} L_1^T > 0$. By this, it is especially true that $B^{(2)} > 0$.

□

Theorem 1.7. *For every symmetric positive definite matrix there exists a unique decomposition $A = LDL^T$ where L is an upper triangular matrix (with $l_{ii} = 1$) and D is a positive diagonal matrix.*

Proof. This follows by the construction made in proof 1.4 for $k = 2, \dots, n-1$. Thus we get $L = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1}$ and D as diagonal matrix of the pivot elements. □

Because all diagonal elements d_{ii} are positive definite, there exists

$$D^{\frac{1}{2}} := \text{diag}\left(\sqrt{d_{11}}, \dots, \sqrt{d_{nn}}\right)$$

Corollary 1.8. *There exists some $\overline{L} := LD^{\frac{1}{2}}$ (lower triangular matrix) such that $A = \overline{L}\overline{L}^T$.*

Algorithm 3 (Cholesky decomposition)**.**

- *For* $k = 1, \dots, n$
  - $l_{kk} := (a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2)^{\frac{1}{2}}$
  - *For* $i = k+1, \dots, n$
    - $l_{ik} := \frac{a_{ik} - \sum_{j=1}^{k-1} l_{ij} l_{kj}}{l_{kk}}$

2 Error analysis and matrix conditioning

So far, we use input data (A, b) to retrieve the result $A^{-1}b$.

Consider an abstract problem characterized by (f, x) with given map f and input data x. Thus in theory, we have the input, apply the algorithm and retrieve the result. But in practice, we have input with some error, an error in the algorithm and an error in the result.

In the following, we investigate input errors, which we cannot avoid in the general case. In the best case we can change the problem task. This refers to *conditioning of the problem*.

The algorithm triggers errors, which we can reduce or avoid by adapting the algorithm. This refers to *stability of the algorithm*.

2.1 Number representation and rounding errors

Definition 2.1. *Let x be a real number. $\tilde{x} \in \mathbb{R}$ is an approximating value for x. Then $\tilde{x} - x$ is called* absolute error *of $\tilde{x}$ and, assuming $x \neq 0$, $\frac{x - \tilde{x}}{x}$ is the* relative error *of x.*

Even if we know the input know exactly, the representation of continuous numbers yields rounding errors.

A number $x \in M$ is represented as $x = \text{sign}(x) \cdot a \cdot E^{e-k}$. The number system M is defined by the following four integer parameters:

basis $E \in \mathbb{N}$ with $E > 1$; mostly $E = 2$

precision $k \in \mathbb{N}$

exponent e in domain $e_{\text{min}} \leq e \leq e_{\text{max}}$ where $e_{\text{min}}, e_{\text{max}} \in \mathbb{Z}$

mantissa $a \in \mathbb{N}_0$ is defined as

$$0 \leq a = a_1 E^{k-1} + a_2 \cdot E^{k-2} + \cdots + a_{k-1} E^1 + a_k E^0 \leq E^k - 1$$

The mantissa length k and a_i are numbers of the number system, so 0 or 1 in case $E = 2$. If $x \neq 0$, then we require, that the first number is non-zero, then

$$E^{k-1} \leq q < E^k \qquad \text{if } x \neq 0$$

We call x *k-digit normalized floating point number with basis E.*

Calculating with these numbers is called *calculation with k significant digits.* This way, we get the domain of normalized floating points $x \neq 0$:

$$E^{e_{\text{min}}-1} \leq |x| \leq E^{e_{\text{max}}}(1 - E^{-k})$$

Example 2.2. *Let $k = 3$, $E = 2$, $e_{\text{min}} = -1$ and $e_{\text{max}} = 3$. Let $x > 0$. Then the smallest number is 0.25 and the largest number is 7. Now look at the distribution of numbers. There are 3 numbers between 1 and 2.*

Remark. *The numbers are not equidistantly distributed.*

After every arithmetic operation, the result x is rounded to a uniquely defined value $\text{rd}(x)$. We define the *machine precision* $\varepsilon := \frac{E}{2} E^{-k}$.

Lemma 2.3. *If $x \neq 0$ is within the domain of normalized floating point values and $\text{rd}(x) \in M$, then*

$$|\text{rd}(x) - x| \leq \frac{E^{e-k}}{2}$$

is the maximum absolute error *with respect to rounding.*

$$\frac{|\text{rd}(x) - x|}{|x|} \leq \frac{1}{2} E^{1-k} = \varepsilon$$

is the maximum relative error *with respect to rounding.*

Proof. Without loss of generality, let $x > 0$. Then we can denote,

$$x = \mu E^{e-k} \qquad E^{k-1} \leq \mu \leq E^k - 1$$

So, x lies in between the neighboring floating point numbers $x_1 = \lfloor \mu \rfloor E^{e-k}$ and $x_2 = \lceil \mu \rceil E^{e-k}$. By this, we either get $\text{rd}(x) = x_1$ or $\text{rd}(x) = x_2$.

Rounding gives

$$|\mathrm{rd}(x) - x| \leq \frac{x_2 - x_1}{2} \leq \frac{E^{e-k}}{2}$$

Thus, it follows that

$$\frac{|\mathrm{rd}(x) - x|}{|x|} \leq \frac{\frac{E^{e-k}}{2}}{\mu E^{e-k}} \leq \frac{1}{2} E^{1-k} = \varepsilon$$

□

precision	k	e_{min}	e_{max}	ε
single	24	-125	128	$2^{-24} \approx 6 \cdot 10^{-8}$
double	53	-1021	1024	$2^{-53} \approx 1 \cdot 10^{-16}$
extended	64	-16381	16384	$2^{-64} \approx 5 \cdot 10^{-20}$

Table 1: Examples for common number systems. Assuming basis $E = 2$

Lemma 2.4.

1. *It holds that* $\mathrm{rd}(x) = x(1+\delta)$ *with* $|\delta| \leq \varepsilon$
2. *Let* $\circ$ *be one of the elementary operations* $\{+, -, \cdot, /\}$. *Then* $\mathrm{rd}(x \circ y) = (x \circ y)(1+\delta)$ *with* $|\delta| \leq \varepsilon$.
3. *The machine precision* ε *is the smallest positive number* g *for which* $\mathrm{rd}(1+g) > 1$ *is true.*

Definition 2.5.

1. *The significant digits of a number are mantissa with normalized floating point representation.*
2. *When calculating with* k *significant digits, all input values must be rounded to* k *significant digits. Followingly the results of every elementary operation will be rounded to* k *significant digits before continuing. Thus the result is also affected.*
3. *Elimination is the cancellation of leading mantissa in the subtraction of two numbers of same sign.*

Example 2.6. *Depending on the chosen expressions, we can get different results. One example is:*

$$99 - 70\sqrt{2} = \sqrt{9801} - \sqrt{9800} = \frac{1}{\sqrt{9801} + \sqrt{9800}} \approx 0.05050633883346584\ldots$$

2.2 Condition of a problem

Question: What effects do deviations in input data have independent of the chosen algorithm?

	$99 - 70\sqrt{2}$	$\sqrt{9801} - \sqrt{9800}$	$\frac{1}{\sqrt{9801}-\sqrt{9800}}$
$k = 2$	1	0	0.0050
$k = 4$	0.02000	0.02000	0.005051
$k = 10$	0.005050660000	0.005050660000	0.005050633884

Table 2: Difference in result depending on the expression. Apparently elimination occurs for subtraction $99 - 70\sqrt{2}$ yielding ill-conditioned results

We already observed, that no difference occurs between input x and all inputs $\tilde{x}$ with an absolute error smaller than machine precision. Therefore, consider the input set E, which contains all disrupted inputs $\tilde{x}$

$$E = \{\tilde{x} \in \mathbb{R} \mid |\tilde{x} - x| < \varepsilon\varepsilon \cdot |x|\}$$

The map f (description of the problem) maps an input set E in a resulting set $U = f(E)$.

Example (Intersection of two lines)**.** *If we compute the intersection of lines with angles close to $0°$, the corresponding LES is ill-conditioned. Computing intersection for almost orthogonal lines is well-conditioned.*

↓ *This lecture took place on 2018/10/11.*

We consider the problem (f, X) given by the map $f : U \subset \mathbb{R}^n \to \mathbb{R}^m$ with $U \subset \mathbb{R}^n$ open. Let $x \in U$ and δ as (relative or absolute) precision of the input data be given.

Distinction between:

- $\|\tilde{x} - x\| \leq \delta$ (absolute) or $\|\tilde{x} - x\| \leq \delta \|x\|$ (relative) for some norm $\|\cdot\|$ on $\mathbb{R}^n$
- $|\tilde{x}_i - x_i| \subseteq \delta$ (absolute) or $|\tilde{x}_i - x_i| \leq \delta |x_i|$ (relative), hence componentwise for $i = 1, \dots, n$.

Normwise condition analysis

Assumption: The input error δ is sufficiently small, so we use *linearized error theory* for the asymptotic behavior of $\delta \to 0$.

Notation: Two functions $g, h : \mathbb{R}^n \to \mathbb{R}^m$ are equal in the *first approximation* or *leading approximation* for $x \to x_0$.

$$g(x) \doteq h(x) \text{ for } x \to x_0$$

if $g(x) = h(x) + \mathcal{O}(\|h(x)\|)$ for $x \to x_0$. The Landau symbol $o(\|h(x)\|)$ for $x \to x_0$ denotes a function φ such that

$$\lim_{x \to x_0} \frac{\|\varphi(x)\|}{\|h(x)\|} = 0$$

If f is differentiable in x, we have $f(\tilde{x}) - f(x) \doteq f'(x)(\tilde{x} - x)$ for $\tilde{x} \to x$. Analogously, "$g(x) \dot{\leq} h(x)$ for $x \to x_0$" (componentwise).

Definition 2.7. *The* absolute normwise condition *of problem (f, X) is the smallest number $\kappa_{\text{abs}} \geq 0$, such that*

$$\|f(\tilde{x}) - f(x)\| \dot{\leq} \kappa \|\tilde{x} - x\| \text{ for } \tilde{x} \to x$$

The problem (f, X) is ill-posed, *if there is no such number (formally $\kappa_{\text{abs}} = \infty$).*

The relative normwise condition of (f, X) is the smallest number $\kappa_{\text{rel}} \geq 0$ such that

$$\frac{\|f(\tilde{x}) - f(x)\|}{\|f(x)\|} \dot{\leq} \kappa_{\text{rel}} \frac{\|\tilde{x} - x\|}{\|x\|} \text{ for } \tilde{x} \to x$$

Remark. *κ_{abs} give the amplification of the absolute error, κ_{rel} is the relative error.*

If f is differentiable in x, then

$$\kappa_{\text{abs}} = \|f'(x)\|, \kappa_{\text{rel}} = \frac{\|x\|}{\|f(x)\|} \|f'(x)\|$$

where $\|f'(x)\|$ is the number of the Jacobian matrix $f'(x) \in \mathbb{R}^{m \times n}$ in regards of the norm

$$\|A\| := \sup_{x \neq 0} \frac{\|ax\|}{\|x\|} = \sup_{\|x\|=1} \|Ax\| \text{ for } A \in \mathbb{R}^{m \times n}$$

Example 2.8 (Condition number addition/subtraction)**.**

$$\textit{Addition:} \quad f : \mathbb{R}^2 \to \mathbb{R} \qquad \begin{pmatrix} a \\ b \end{pmatrix} \mapsto f(a, b) = a + b$$

with the derivative $f'(a, b) = (1, 1) \in \mathbb{R}^{1 \times 2}$. For the 1-norm on $\mathbb{R}^2$ it holds that

$$\left\| \begin{pmatrix} a \\ b \end{pmatrix} \right\| = |a| + |b|$$

The norm (associated operator norm) is

$$\|f'(a, b)\| = \sup_{\|x\|_1 = 1} \left| \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right| = 1$$

We retrieve the condition numbers of addition as

$$\kappa_{\text{abs}} = 1 \text{ and } \kappa_{\text{rel}} = \frac{|a| + |b|}{|a + b|}$$

Consequence:

- *$\kappa_{\text{rel}} = 1$ for addition of two numbers with same sign*
- *subtraction of two almost equal numbers given $|a + b| \ll |a| + |b| \implies \kappa_{\text{rel}} \gg 1$.*

Example 2.9 (Condition of a linear equation system $Ax = b$)**.** *If we only consider $b \in \mathbb{R}^n$ as input data, the problem is specified by the* linear map $f : \mathbb{R}^n \to \mathbb{R}^n$ *and $b \mapsto f(b) = A^{-1}b$. The derivative is $f'(b) = A^{-1}$ and we retrieve the condition numbers*

$$\kappa_{\text{abs}} = \left\|A^{-1}\right\| \text{ and } \kappa_{\text{rel}} = \frac{\|b\|}{\|A^{-1}b\|}\left\|A^{-1}\right\| = \frac{\|Ax\|}{\|x\|}\left\|A^{-1}\right\|$$

What about the deviations in A? For this, consider A as input data

$$f : \mathbb{R}^{n\times n} \supset \mathrm{GL}(n) \to \mathbb{R}^n, A \mapsto f(A) = A^{-1}b$$

for some fixed $b \in \mathbb{R}^n$. The map f is non-linear, but differentiable.

Lemma 2.10. *The map $g : \mathbb{R}^{n\times n} \supset \mathrm{GL}(n) \to \mathrm{GL}(n)$ with $g(A) = A^{-1}$ is differentiable and $g'(A) \cdot C = -A^{-1}CA^{-1}$ for all $C \in \mathbb{R}^{n\times n}$.*

Proof. Consider $I = (A + tC)(A + tC)^{-1}$ for sufficiently small t and derive by t:

$$0 = C(A + tC)^{-1} + (A + tC)\frac{d}{dt}(A + tC)^{-1}$$

For $t = 0$, it especially follows that $g'(A)C = \frac{d}{dt}(A + tC)^{-1}\big|_{t=0} = -A^{-1}CA^{-1}$. □

By Lemma 2.10, by the derivative of the solution $f(A) = A^{-1}b$ to A,

$$f'(A)C = -A^{-1}CA^{-1}b = -A^{-1}Cx \text{ for } C \in \mathbb{R}^{n\times n}$$

We retrieve the condition numbers

$$\kappa_{\text{abs}} = \|f'(A)\| = \sup_{\|C\|=1}\left\|A^{-1}Cx\right\| \leq \left\|A^{-1}\right\|\|x\|$$

$$\kappa_{\text{rel}} = \frac{\|A\|}{\|x\|}\|f'(A)\| \leq \|A\|\left\|A^{-1}\right\|$$

By the sub-multiplicativity, $\|Ax\| \leq \|A\|\|x\|$, of the matrix norm, it also holds for the relative condition in regards of the input b, that $\kappa_{\text{rel}} \leq \|A\| \cdot \left\|A^{-1}\right\|$. The value $\kappa(A) := \|A\|\left\|A^{-1}\right\|$ is called *condition number* of matrix A.

↓ *This lecture took place on 2018/10/16.*

We can show that for invertible matrices, we have $\kappa(A) = \frac{\max_{\|x\|=1}\|Ax\|}{\min_{\max_{\|x\|=1}\|Ax\|}}$ depending on the choice of the norm. By this it follows that:

1. $\kappa(A) \geq 1$
2. $\kappa(\alpha A) = \kappa(A) \forall \alpha \in \mathbb{R}, \alpha \neq 0$
   hence the condition number is invariant under scalar transformations (in contrast to determinants).
   Remember that $\det(\alpha A) = \alpha^n \det(A)$.
3. $A \in \mathbb{R}^{n\times n}$ with $A \neq 0$ is singular iff $\kappa(A) = \infty$

Componentwise condition analysis

The normwise condition analysis is inappropriate in particular special cases.

Example 2.11. *Consider the equation system $Ax = b$ with a diagonal matrix*

$$A = \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix} \qquad A^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{\varepsilon} \end{pmatrix}$$

The problem is completely decoupled and we expect a well-conditioned problem (at least for diagonal deviations). For normwise conditioning (in regards of $\|\cdot\|_\infty$), it holds that

$$\kappa_\infty(A) = \left\|A^{-1}\right\|_\infty \|A\|_\infty = \frac{1}{\varepsilon} \qquad \varepsilon \to 0$$

For $\varepsilon \to 0$ the condition number will be arbitrary large, because it permits arbitrary deviations in the matrix.

By componentwise condition analysis, we get $(f, x) \rightsquigarrow$ smallest number κ_{rel} such that

$$\max_i \frac{|f_i(\tilde{x}) - f_i(x)|}{|f_i(x)|} \leq \max_i \frac{|\tilde{x}_i - x_i|}{|x_i|}$$

2.3 Stability of an algorithm

We now consider error, that are created, if the map f is approximated by some map $\tilde{f}$ (algorithmic realization). It encompasses all rounding errors and approximation errors and return an approximation $\tilde{f}(x)$ instead of $f(x)$.

Question: Is $\tilde{f}(x)$ acceptable as an substitute of $f(x)$? Compare with Figure 1.

Stability of the componentwise forward analysis

Definition 2.12 (Forward stability)**.** *Let $\tilde{f}$ be the floating point implementation of an algorithm to solve problem (f, x) of relative normwise condition κ_{rel}. The* stability indicator *of a normwise forward analysis is the smallest number $\sigma \geq 0$, such that for all $\tilde{x} \in E$ it holds that*

$$\frac{\left\|\tilde{f}(\tilde{x}) - f(\tilde{x})\right\|}{\|f(\tilde{x})\|} \leq \sigma \kappa_{\text{rel}} \varepsilon \qquad \textit{for } \varepsilon \to 0$$

$\overline{\kappa}_{\text{rel}}$ is the smallest number $\sigma \geq 0$ such that

$$\max_i \frac{\left|\tilde{f}_i(\tilde{x}) - f_i(\tilde{x})\right|}{|f_i(\tilde{x})|} \dot{\leq} \sigma \cdot \overline{\kappa}_{\text{rel}} \varepsilon \qquad \text{for } \varepsilon \to 0$$

where $\overline{\kappa}_{\text{rel}}$ is the componentwise relative conditioning of (f, x). We call algorithm $\tilde{f}$ *stable* in regards of forward analysis if σ is smaller than the number of consecutively executed elementary operations.

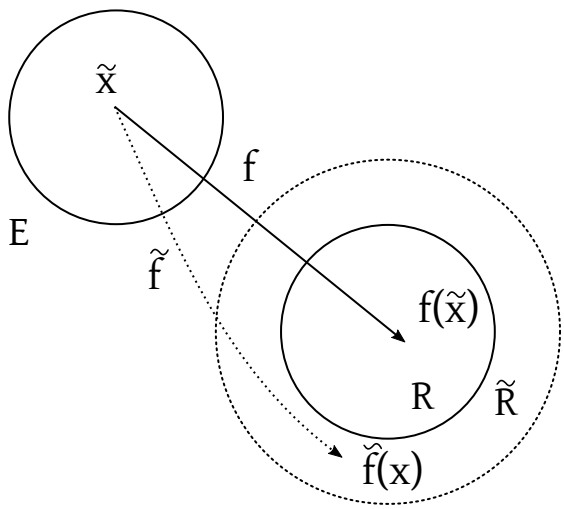

Figure 1: Map of the error with $\tilde{f}$ instead of f

Lemma 2.13. *For the elementary operations* $\{+, -, \cdot, /\}$ *and the floating point implementations* $\left\{\hat{+}, \hat{-}, \hat{\cdot}, \hat{/}\right\}$ *it holds that* $\sigma\kappa_{\text{rel}} \leq 1$.

Proof. For every floating point implementation it holds that

$$x\hat{*}y = (x * y)(1 + \delta) \qquad * \in \{+, -, \cdot, /\}$$

by Lemma 2.4 for some δ with $|\delta| \leq \varepsilon$. Thus,

$$\frac{|x\hat{*}y - x * y|}{x * y} = \frac{|(x * y)(1 + \delta) - x * y|}{|x * y|} = |\delta| \leq 1 \cdot \varepsilon$$

□

This approach is rather impractical (determination of the conditioning number of the problem is difficult).

Stability of the componentwise backwards analysis

Idea: interpret the error in the algorithm like an input error. The result $\tilde{y} = \tilde{f}(\tilde{x})$ is considered as the exact result $\tilde{y} = f(\hat{x})$ for distorted data $\hat{x}$. $E = \{\tilde{x} \mid \|\tilde{x} - x\| \leq \varepsilon \|x\|\}$. This is only possible if $\tilde{y}$ is a admissible solution of f at all. In the other case, we call the algorithm *unstable*.

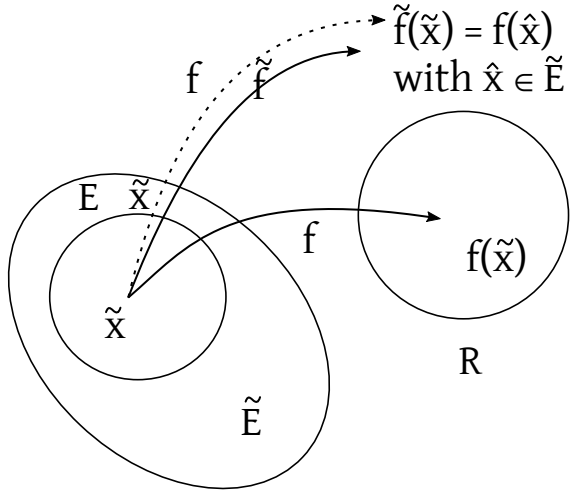

Figure 2: Backwards analysis

Definition 2.14. *The* normwise backwards error *of the algorithm $\tilde{f}$ to solve the problem (f, x) is the smallest number $\eta \geq 0$, for which $\forall \tilde{x} \in E_\varepsilon$ some $\hat{x}$ exists such that*

$$\frac{\|\hat{x} - \tilde{x}\|}{\|\tilde{x}\|} \dot{\leq} \eta \qquad \text{for } \varepsilon \to 0$$

The componentwise backwards error *is defined as*

$$\max_i \frac{|\hat{x}_i - \tilde{x}_i|}{|\tilde{x}_i|} \dot{\leq} \eta \qquad \text{for } \varepsilon \to 0$$

The algorithm is called stable *(in regards of backwards analysis) with respect to the relative input error ε if $\eta < \varepsilon \cdot$ number of elementary operations. For given ε we define a* stability indicator *of the backwards analysis as quotient*

$$\sigma_R := \frac{\eta}{\varepsilon}$$

Lemma 2.15. *For stability indicators σ and σ_R, of the forward/backward analysis respectively, it holds that $\sigma \leq \sigma_R$. Especially by backwards stability, forwards stability follows.*

Proof. By definition of the backwards error, for every $\tilde{x} \in E$ there exists some $\hat{x}$ such that $f(\hat{x}) = \tilde{f}(\tilde{x})$ and

$$\frac{\|\hat{x} - \tilde{x}\|}{\|\tilde{x}\|} \dot{\leq} \eta = \sigma_R \cdot \varepsilon \qquad \text{for } \varepsilon \to 0$$

For the relative error in the result, we have

$$\frac{\left\|\tilde{f}(\tilde{x}) - f(\tilde{x})\right\|}{\|f(\tilde{x})\|} = \frac{\|f(\hat{x}) - f(\tilde{x})\|}{\|f(\tilde{x})\|} \dot{\leq} \kappa_{\text{rel}} \frac{\|\hat{x} - \tilde{x}\|}{\|\tilde{x}\|} \dot{\leq} \kappa_{\text{rel}} \cdot \sigma_R \cdot \varepsilon \qquad (\text{for } \varepsilon \to 0)$$

□

Remark. *$Ax = b$ with solution $\tilde{x}$, approximation $\hat{x}$. Error $\|\hat{x} - \tilde{x}\|$ is not computable, because $\tilde{x}$ is unknown. Residue $\|b - A\hat{x}\|$.*

2.4 Backwards analysis for linear equation systems

Question: Is the LU decomposition a backwardsstable algorithm?

In the following, we will use the following notation for vectors and matrices:

$$|A| \leq |B| \iff |a_{ij}| \leq |b_{ij}| \, \forall i, j$$

Lemma 2.16. *The floating point implementation of the scalar product*

$$\langle x, y \rangle \coloneqq x_n y_n + \langle x^{n-1}, y^{n-1} \rangle$$

with $x^{n-1} \coloneqq (x_1, \dots, x_{n-1})^T$ and $y^{n-1} \coloneqq (y_1, \dots, y_{n-1})^T$ denotes some solution $\langle x, y \rangle_{rd} \coloneqq \mathrm{rd}(\langle x, y \rangle)$ for some $x, y \in \mathbb{R}^n$ such that $\langle x, y \rangle_{\mathrm{rd}} = \langle \hat{x}, y \rangle$ for some $\hat{x} \in \mathbb{R}^n$ with $|x - \hat{x}| \dot{\leq} n\varepsilon |x|$, hence the relative componentwise backwards error is $\eta \leq n \cdot \varepsilon$ and the scalar product is stable in regards of backwards analysis (with $2n - 1$ elementary operations by the scalar product).

Proof. We need to prove:

$$\forall i : |x_i - \hat{x}_i| \leq \underbrace{n\varepsilon |x_i|}_{h(\varepsilon)} + \overbrace{\sigma(\|h(\varepsilon)\|)}^{\sigma} \qquad \text{for } \varepsilon \to 0 \text{ with } \lim_{\varepsilon \to 0} \frac{\|\varphi(\varepsilon)\|}{\|h(\varepsilon)\|} = 0$$

Thus, we look for a function that converges superlinear towards 0 for $\varepsilon \to 0$ ($\mathcal{O}(\varepsilon)$).

We use induction over n to prove this. For $n = 1$, the statement is true by Lemma 2.4. Now let $n > 1$ and let the statement be true for $n - 1$. For floating point implementation of recursion it is true that

$$\langle x, y \rangle_{\mathrm{rd}} = (x_n y_n (1 + \delta) + \langle x^{n-1}, y^{n-1} \rangle_{\mathrm{rd}})(1 + \tilde{\delta})$$

where $|\delta| \leq \varepsilon$ and $|\tilde{\delta}| \leq \varepsilon$ characterize the relative error of multiplication and addition respectively.

By the induction hypothesis, it holds that $\langle x^{n-1}, y^{n-1} \rangle_{\mathrm{rd}} = \langle z, y^{n-1} \rangle$ for some $z \in \mathbb{R}^{n-1}$ with

$$|x^{n-1} - z| \leq (n-1)\varepsilon |x^{n-1}| + \mathcal{O}(\varepsilon)$$

By $\hat{x}_n = x_n(1 + \delta)(1 + \tilde{\delta})$ and $\hat{x}_k = z_k(1 + \tilde{\delta})$ by $k = 1, \dots, n - 1$ it follows that

$$\begin{aligned}
\langle x, y \rangle_{\mathrm{rd}} &= x_n y_n (1 + \delta)(1 + \tilde{\delta}) + \langle z(1 + \varepsilon), y^{n-1} \rangle \\
&= \hat{x}_n y_n + \langle \hat{x}^{n-1}, y^{n-1} \rangle = \langle \hat{x}, y \rangle \\
&\text{with } |x_n - \hat{x}_n| \leq 2\varepsilon |x_n| + |x_n| |y_n| \delta\tilde{\delta} \\
&\leq n\varepsilon |x_n| + \mathcal{O}(\varepsilon) \\
\text{and } |x_k - \hat{x}_k| &\leq |x_k - z_k| + |x_k - \hat{x}_k| \\
&\leq (n-1)\varepsilon |x_k| + \mathcal{O}(\varepsilon) + \varepsilon |z_k| \\
&\leq n\varepsilon |x_k| + \mathcal{O}(\varepsilon) \text{ for } k = 1, \dots, n - 1
\end{aligned}$$

The last estimate can be made using

$$\begin{aligned} |z_k| - |x_k| \leq |x_k - z_k| &\leq (n-1)\varepsilon\,|x_k| + \mathcal{O}(\varepsilon) \\ \implies \varepsilon\,|z_k| &\leq \varepsilon\,|x_k| + \mathcal{O}(\varepsilon) \end{aligned}$$

□

Theorem 2.17. *The floating point implementation of forward substitution to solve a linear equation system $Lx = b$ with some lower triangular matrix L computes the solution $\hat{x}$ such that there exists some lower triangular matrix $\hat{L}$ with $\hat{L}\hat{x} = b$ and $\left|L - \hat{L}\right| \dot{\leq} n\varepsilon\,|L|$. Hence for componentwise relative backwards errors, we have $\eta \leq n \cdot \varepsilon$ and the forward substitution is stable in regards of backwards analysis.*

Proof. Again, we will use a recursive approach.

$$l_{kk}x_k = b_k - \left\langle l^{k-1}, x^{k-1}\right\rangle$$

For $k = 1, \dots, n$, we have

$$\begin{aligned} x^{k-1} &\coloneqq (x_1, \dots, x_{k-1})^T \text{ and} \\ l^{k-1} &\coloneqq (l_{k,1}, \dots, l_{k,k-1})^T \end{aligned}$$

In floating point arithmetics, the following recursion is yielded,

$$l_{kk}(1+\delta_k)(1+\tilde{\delta}_k)\hat{x}_k = b_k - \left\langle l^{k-1}, x^{k-1}\right\rangle_{\text{rd}}(1+\tilde{\delta}_k)$$

where $\delta_k, \tilde{\delta}_k$ with $|\delta_k| \leq \varepsilon$ and $\left|\tilde{\delta}_k\right| \leq \varepsilon$ are the relative errors of multiplication/addition. By Lemma 2.16, it follows that

$$\left\langle l^{k-1}, \hat{x}^{k-1}\right\rangle_{\text{rd}} = \left\langle \tilde{l}^{k-1}, \hat{x}^{k-1}\right\rangle$$

for some vector $\tilde{l}^{k-1} = (\tilde{l}_{k,1}, \dots, \tilde{l}_{k,k-1})^T$ with $\left|l^{k-1} - \tilde{l}^{k-1}\right| \dot{\leq} (k-1)\varepsilon\left|l^{k-1}\right|$. Now let $\hat{l}_{kk} \coloneqq l_{kk}(1+\delta_k)(1+\tilde{\delta}_k)$ and $\hat{l}^{k-1} \coloneqq (1+\tilde{\delta}_k)\tilde{l}^{k-1}$ then

$$\begin{aligned} \left|\hat{l}^{k-1} - l^{k-1}\right| &\dot{\leq} (k-1)\varepsilon\left|l^{k-1}\right| + \left|\tilde{\delta}_k\right|\left|\tilde{l}^{k-1}\right| \\ &\dot{\leq} k\varepsilon\left|l^{k-1}\right| \\ \left|\hat{l}_{kh} - l_{kj}\right| &\dot{\leq} 2\varepsilon \\ \implies \hat{L}\hat{x} = b \wedge \left|L - \hat{L}\right| &\dot{\leq} n\varepsilon\,|L| \end{aligned}$$

□

↓ *This lecture took place on 2018/10/18.*

Lemma 2.18. *Let A have a LU decomposition. Then Gaussian elimination determines $\hat{L}$ and $\hat{U}$ such that $\hat{L}\hat{U} = \hat{A}$ for some matrix $\hat{A}$ with*

$$\left|A - \hat{A}\right| \dot{\leq} n\left|\hat{L}\right|\left|\hat{U}\right|\varepsilon$$

Theorem 2.19. *A has some LU-decomposition. Then Gaussian elimination for linear equation system $Ax = b$ gives a solution $\hat{x}$ with $\hat{A}\hat{x} = b$ for some matrix $\hat{A}$ with*

$$\left|A - \hat{A}\right| \dot{\leq} 2n \left|\hat{L}\right| \left|\hat{U}\right| \varepsilon$$

Theorem 2.20. *The Gaussian elimination with column pivot strategy for the linear equation system $Ax = b$ determines some $\hat{x}$ such that $\hat{A}\hat{x} = b$ for some matrix $\hat{A}$ with*

$$\frac{\left\|A - \hat{A}\right\|_\infty}{\|A\|_\infty} \dot{\leq} 2n^3 \rho_n(A)\varepsilon \qquad \text{where } \rho_n(A) := \frac{\alpha_{\max}}{\max_{i,j} |a_{ij}|}$$

and where $\alpha_{\max}$ is the largest absolute value of an element that occurs during elimination in the remainder matrices $A^{(1)} = A$ to $A^{(n)} = U$.

$$A = \begin{bmatrix} 1 & 0 & \dots & 0 & 1 \\ -1 & \vdots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & & \\ -1 & \dots & \dots & -1 & 1 \end{bmatrix}$$

For matrix A it holds that $\rho_n(A) = 2^{n-1}$.

Consider that the previous result also provides *no* positive statement about the backwards stability of the LU decomposition.

3 Linear least squares

Goal: Solution of overdetermined linear equation systems

$$Ax = b \qquad A \in \mathbb{R}^{m \times n} \qquad m > n \qquad b \in \mathbb{R}^m$$

by the method of linear least squares.

For this, we are going to use (numerically stable) orthogonal transformations.

3.1 Gaussian process of least squares

Problem: Given m input data points (t_i, b_i) and $t_i, b_i \in \mathbb{R}$ for $i = 1, \dots, m$. They describe the values b_i of an object at timestamp t_i.

We assume that there exists an underlying law such that the dependency of b and t can be represented by some *modelling function* φ with $b(t) = \varphi(t_i; x_1, \dots, x_n)$. The modelling function therefore takes n unknowns p_i.

Example 3.1 (Ohm's law)**.** *Consider Ohm's law $b = xt = \varphi(t_i x)$ where t denotes the current, b the voltage and x resistance.*

So want to approximate data points in the $\mathbb{R}^2$ plane linearly.

If the model would be *exact* and if you have no errors in measurement, then we are supposed to evaluate parameters $x_1, \dots, x_n$ such that $b_i = b(t_i) = \varphi(t_i; x_1, \dots, x_n)$ for $i = 1, \dots, m$.

In reality, measurements are subject to errors and modelling functions are also just approximations of reality. We therefore require that $b_i \approx \varphi(t_i; x_1, \dots, x_n)$ for $i = 1, \dots, m$. We weight the individual deviations

$$\Delta i \coloneqq b_i - \varphi(t_i; x_1, \dots, x_n) \qquad i = 1, \dots, m$$

By Gauss, determine x_i such that $\sum_{i=1}^m \Delta i^2$ becomes minimal. We use the short notation $\Delta^2 \coloneqq \sum_{i=1}^m \delta_i^2 \implies \min$. Often it is more useful to weight the individual deviations differently. For some measurement precision δb, introduce weights

$$\sum_{i=1}^m \left(\frac{\Delta i}{\delta b_i} \right) \to \min$$

We consider the special case of linear (with respect to x) modelling functions φ, hence $\varphi(t_i; x_1, \dots, x_n) = a_1(t)x_1 + \dots + a_n(t)x_n$ where $a_1, \dots, a_n : \mathbb{R} \to \mathbb{R}$ are arbitrary functions. Furthermore we choose $\|\cdot\| = \|\cdot\|_2$. We can denote the least square problem as

$$\|b - Ax\| \to \min \qquad \text{where } b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}, x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

$$\text{and } A = (a_{ij}) \in \mathbb{R}^{m \times n} \text{ with } a_{ij} \coloneqq a_j(t_i)$$

Orthogonal equations

We look for some point $z = Ax$ of the image space $U(A)$ of A, that has the smallest distance to some given b. Ax is the orthogonal projection of b to the subspace $U(A) = \text{image}(A)$.

Theorem 3.2. *Let V be a vector space of* $\dim V < \infty$ *with scalar product $\langle , \rangle$. Let $U \subset V$ be a subspace. Let*

$$U^\perp = \{v \in V \mid \langle v, u \rangle = 0 \forall u \in U\}$$

be the orthogonal complement in V. Then for all $v \in V$ with respect to the norm induced by $\|v\| = \sqrt{\langle v, v \rangle}$ such that

$$\|v - u\| = \min_{\tilde{u} \in U} \|v - \tilde{u}\| \iff v - u \in U^\perp$$

Proof sketch:

- $V = U \oplus U^\perp$
- $v = u + w, u \in U, w \in U^\perp$ unique
- $\tilde{u} \in U$ arbitrary:

  $$\|v - \tilde{u}\|^2 = \|(v - u) + (u - \tilde{u})\|^2 = \|v - u\|^2 + \|u - \tilde{u}\|^2 \geq \|v - u\|^2$$

  with $v - u = w$ and $(u - \tilde{u}) \in U$.

↓ *This lecture took place on 2018/10/23.*

The unique solution $u \in U$ of $\min \|v - u\|$ is called *orthogonal projection* of V to U. The map $P : V \to U$ with $v \mapsto Pv$ with $Pv = u$ is linear and is called *orthogonal projection* of V to U.

Remark 3.3. *We get an analogous statement if we replace U by an affine subspace $W = w_0 + U \subseteq V$ where $w_0 \in V$. U is the parallel, to W, linear subspace of V.*

For all $v \in V$, $w \in W$: $\|v - w\| = \min_{\tilde{w} \in w} \|v - \tilde{w}\| \iff v - w \in U^\perp$.

Theorem 3.4. *The vector $x \in \mathbb{R}^n$ is a solution of the linear least squares problems $\min \|b - Ax\| \iff$ x satisfies the so-called orthogonal equations $A^T Ax = A^T b$ (dt. "Normalengleichungen").*

Especially, the linear least squares problem is uniquely solvable iff A has full column rank.

Proof. By Theorem 3.2, $V := \mathbb{R}^n$, $U := \text{image}(A) = \text{rank}(A)$.

$$\begin{aligned} \implies \max \|b - Ax\| &\iff \langle b - Ax, A\tilde{x} \rangle = 0 \forall \tilde{x} \in \mathbb{R}^n \\ &\iff 0 = \langle A^t(b - Ax), \tilde{x} \rangle \ \forall \tilde{x} \in \mathbb{R}^n \\ &\iff A^t(b - Ax) = 0 \\ &\iff A^t b = A^t Ax \end{aligned}$$

Thus the first statement is proven. How about the second statement?

$$A^t A \text{ regular} \iff \text{rank}(A) = n$$

□

Remark (Geometric interpretation). *$b - Ax$ is orthogonal to* $\text{range}(A)$ *with $A \subseteq \mathbb{R}^n$.*

Solution of the orthogonal equation system: Theoretically, we can create $A^T A$ and retrieve a symmetric positive definite matrix ($\implies$ decomposition with Cholesky process).

Remark (Computing time).

1. *Determination of $A^t A$ (only one half, because of symmetry)*

$$(A^t A)_{ij} = (\langle a_j, a_j \rangle)_{i,j} \qquad a_j = j\text{-th column of } A \to \frac{1}{2} n^2 n$$

2. *Cholesky decomposition $\sim \frac{1}{6} n^3$*

In case of $m \gg n$, the first approach will take over $\implies \sim \frac{1}{2} n^2 n$. $\frac{2}{3} m^3$ for $n \approx m$.

Lemma 3.5. *For some matrix $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ and full column rank* ($\text{rank}(A) = n$), *$\kappa_2(A^t A) = (\kappa_2(A))^2$.*

Proof. By definition of condition numbers:

$$
\begin{aligned}
(\kappa_2(A))^2 &= \frac{\max_{\|x\|=1} \|Ax\|_2^2}{\min_{\|x\|=1} \|Ax\|_2^2} \\
&= \frac{\max_{\|x\|=1} \langle Ax, Ax\rangle}{\min_{\|x\|=1} \langle Ax, Ax\rangle} \\
&= \frac{\max \langle A^t Ax, x\rangle}{\min \langle A^t Ax, x\rangle} && \text{“Rayleigh eigenvalue”} \\
&\overset{A^t A \text{sym.}}{=} \frac{\lambda_{\max}(A^t A)}{\lambda_{\min}(A^t A)} \\
&\overset{A^t A \text{sym.}}{=} \kappa_2(A^t A)
\end{aligned}
$$

□

Recognize that the amplification of the error for the orthogonal equation is larger. Therefore it makes sense to avoid computing $A^T A$ and use the original matrices A and A^T instead, e.g.

$$
\begin{pmatrix} I & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} r \\ 0 \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}
$$

$$
\implies A^T r = 0 \wedge r \neq Ax = b
$$

$$
\implies r = b - Ax \qquad \implies 0 = A^T r = A^T(b - Ax)
$$

$$
\|A_3\|_2^2 = \max_{\|x\|_2=1} \|Q_j x\|_2^2 = \max_{\|x\|_2=1} \langle x, Q^T Q x\rangle = 1
$$

Recognize that $Q^T Q = I$.

3.2 Orthogonalization process

We can represent the elimininatation process for linear equation systems (e.g. Gaussian elimination) formally as

$$
A \xrightarrow{f_1} B_1 A \xrightarrow{f_2} B_2 B_1 A \xrightarrow{f_3} \dots \xrightarrow{f_k} B_k \dots B_1 A \cdots \to U
$$

where

$$
U = \begin{pmatrix} 1 & * \\ 0 & \varepsilon \end{pmatrix}
$$

and where matrices B_j describe the operations applied to matrix A. Through this process, the condition numbers of the individual matrices can be amplified, such that instabilities can occur. If we instead use orthogonal transformations Q_j for elimination, we have

$$
\kappa_2(Q_j) = \|Q_j\|_2 \cdot \|Q_j\|_2^{-1} = \|Q_j\|_2 \|Q_j\|_2 = 1
$$

Orthogonalization process are necessarily stable, but have higher computing times compared to Gaussian elimination. If we assume we transformed a given matrix $A \in$

$\mathbb{R}^{m\times n}$ with $m \geq n$ with some orthogonal matrix $Q \in \mathbb{R}^{m\times m}$ into upper triangular matrix, then

$$Q^T A = \begin{bmatrix} * & \dots & x \\ & \ddots & \vdots \\ & & x \\ 0 & & * \\ \vdots & & \vdots \\ 0 & & 0 \end{bmatrix} = \begin{bmatrix} U \\ 0 \end{bmatrix}$$

Theorem 3.6. *Let $A \in \mathbb{R}^{m\times n}$, $m \geq n$, of maximum rank n, $b \in \mathbb{R}^m$ and $Q \in \mathbb{R}^{m\times m}$ be an orthogonal matrix with*

$$Q^T A = \begin{bmatrix} U \\ 0 \end{bmatrix} \text{ and } Q^T b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

where $b_1 \in \mathbb{R}^n$, $b_2 \in \mathbb{R}^{m-n}$ and $U \in \mathbb{R}^{n\times n}$ is an (invertible) upper triangular matrix. Then $x = U^{-1}b_1$ is the solution of the linear least squares problems $\min \|b - Ax\|_2$.

Proof. Because Q is orthogonal, $\forall x \in \mathbb{R}^n$ it follows that

$$\|b - Ax\|_2^2 = \left\|Q^T(b - Ax)\right\|_2^2 = \left\|\begin{pmatrix} b_1 - Ux \\ b_2 \end{pmatrix}\right\|_2^2 = \|b_1 - Ux\|_2^2 + \|b_2\|_2^2 \geq \|b_2\|_2^2$$

Because $\operatorname{rank}(A) = \operatorname{rank}(U) = n$, U is invertible. The term $\|b_1 - Ux\|_2^2$ disappear for $x = U^{-1}b_1$ □

Remark. *A:* $0 \neq \|r\| = \|b - Ax\| = \|b_2\|$.

What do orthogonal transformations look like? In general, these transformations are rotations and reflections.

e.g. $x \mapsto \alpha e_1 = Qx$ with $Q = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$.

Alternatively, $x \mapsto \alpha e_1 = x = 2\frac{\langle v,x\rangle}{\langle v,v\rangle}v$ where v is collinear to the difference $x = \alpha e_1$.

Givens rotations

Givens rotations are described by matrices of form

$$Q_{k,l} := \begin{bmatrix} I & & & & \\ & c & & s & \\ & & I & & \\ & -s & & c & \\ & & & & I \end{bmatrix} \in \mathbb{R}^{m\times m}$$

with proper dimension of the unit matrix and $c^2 + s^2 = 1$. For $x \in \mathbb{R}^m$ it holds that

$$x \mapsto y = Q_{k,l}x \text{ with } y_i = (Q_{k,l}x)_i = \begin{cases} cx_k + sx_l & i = k \\ -sx_k + cx_l & i = l \\ x_i & \text{else} \end{cases}$$

Thus for $A = [a_1, \dots, a_n] \in \mathbb{R}^{m \times n}$, we get

$$Q_{k,l} A = [Q_{k,l} a_1, \dots, Q_{k,l} a_n]$$

Hence, only rows k and l of matrix A will be modified.

Question: How to choose c and s to eliminate a component x_l of x? Because $Q_{k,l}$ operates on the (k, l)-layer, we consider without loss of generality $m = 2$. By $x_k^2 + x_l^2 \neq 0$ and $s^2 + c^2 = 1$ it follows that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} x_k \\ x_l \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}$$

$$\iff r = \pm \sqrt{x_k^2 + x_l^2}, c = \frac{x_k}{r}, s = \frac{x_l}{r}$$

Computing x_k^2 might gives us an exponent overflow.

$$\tau := \frac{x_k}{x_l}, s := \frac{1}{\sqrt{1 + \tau^2}}, c := s\tau \text{ if } |x_l| > |x_k|$$

and accordingly,

$$\tau := \frac{x_l}{x_k}, c = \frac{1}{\sqrt{1 + \tau^2}}, s := c\tau \text{ if } |x_k| \geq |x_l|$$

Now transform A iteratively into triangular form.

For example, consider

$$A = \begin{bmatrix} A & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \xrightarrow{(5,4)} \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \end{bmatrix} \xrightarrow{(4,3)} \dots$$

$$\xrightarrow{(2,1)} \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{bmatrix} \xrightarrow{(5,4)} \dots \xrightarrow{(4,3)} \dots \xrightarrow{(5,4)} \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Remark (Computing time). *Computing time for a dense matrix $A \in \mathbb{R}^{m \times n}$:*

1. $\sim \frac{n^2}{2}$ *square roots and* $\frac{4}{3} n^3$ *multiplications, if* $m \approx n$.
2. $m \cdot n$ *are square root and* $2mn^2$ *multiplications if* $m \ggg n$.

This is an alternative for Gaussian elimination. The algorithm is *more stable*, but *more expensive*.

But it can be advantageous for special matrix, e.g. Hessenberg matrices, which are "almost triangular" (so the first subdiagonal also contains non-zero entries). Here, we need $n - 1$ Givens rotations if we want to achieve upper triangular form.

Householder reflections

↓ *This lecture took place on 2018/10/25.*

For given $0 \neq v \in \mathbb{R}^m$, we define the matrix $Q \in \mathbb{R}^{m \times m}$.

$$Q = I - 2\frac{vv^T}{v^T v}$$

They describe reflections on the mirror plane, which are vertical to v. The following properties hold:

1. symmetry: $Q = Q^T$
2. orthogonality: $QQ^T = I$
3. $Q^2 = \text{id}$, thus Q is *involutional*
4. $\text{spec}(Q) = \{\pm 1\}$, -1 simple, 1 $(n-1)$-times.

For $y \in \mathbb{R}^n$,

$$y \mapsto Qy = \left(I - 2\frac{vv^T}{v^T v}\right) y = y - 2 \cdot \frac{\langle v, y \rangle}{\langle v, v \rangle}$$

Question: How can we achieve that Q maps vector y to $\alpha \cdot e_1$? Hence,

$$\alpha \cdot e_1 = y - 2\frac{\langle v, y \rangle}{\langle v, v \rangle} \cdot v \in \text{span}(e_1) = \alpha(\{e_1\})$$

Consider that

$$|\alpha| = \|\alpha e_1\| = \left\| y - 2\frac{\langle v, y \rangle}{\langle v, v \rangle} v \right\|_2 = \|Qy\|_2 = \|y\|_2$$

and $y \in \text{span}(y - \alpha \cdot e_1)$. This way we can retrieve Q by,

$$v \coloneqq y - \alpha e_1 \qquad \alpha = I \|y\|_2$$

To avoid elimination (or subtraction) in $v = (y_1 - \alpha, y_2, \dots, y_n)^T$, choose $\alpha \coloneqq -\text{sign}(y_1) \cdot \|y\|_2$. Now,

$$\langle v, v \rangle = \langle y - \alpha e_1, y - \alpha e_1 \rangle = \|y\|^2 - 2\alpha \langle e_1, y \rangle + \alpha^2 = 2\alpha^2 - 2\alpha \langle e_1, y \rangle = -2\alpha(y_1 - \alpha)$$

Qy for $y \in \mathbb{R}^m$ can be determined by

$$Qy = y - 2\frac{\langle v, y \rangle}{2(y_1 - \alpha)} \cdot v$$

In a first step choose Q_1 (where $A \in \mathbb{R}^{m \times n}$):

$$[a_1 \dots a_n] = A \to A' = Q_1 \cdot A = \begin{bmatrix} \alpha_1 & \\ 0 & \\ \vdots & a'_2 \dots a'_n \\ 0 & \end{bmatrix}$$

where $Q_1 = I - 2\frac{v_1 v_1^T}{v_1^T v_1}$ with $v_1 := a_1 - \alpha_1 e_1$ and $\alpha_1 := -\operatorname{sign}(a_{11}) \cdot \|a_1\|$.

After the k-th step, we get

$$A^{(k)} = \begin{bmatrix} * & & & * \\ & \ddots & & \\ & & * & \\ 0 & & & T^{(k+1)} \end{bmatrix}$$

with $T^{(k+1)} \in \mathbb{R}^{m-k \times n-k}$. Now we create orthogonal matrix

$$Q_{k+1} := \begin{bmatrix} I & 0 \\ 0 & \tilde{Q}_{k+1} \end{bmatrix}$$

where $\tilde{Q}_{k+1}$ modifies matrix $T^{(k+1)}$. The next subcolumn will be eliminated.

After $p = \min(m, n)$ steps, we retrieve an upper triangular matrix

$$R = Q_p \dots Q_1 \cdot A$$
$$\iff A = QR \text{ where } Q = Q_1 \cdot \dots \cdot Q_p$$

Based on Theorem 2.6, we retrieve the following process:

1. $A = QR$ (QR decomposition using Householder or Givens)
2. $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = Q^t b$ with $b_1 \in \mathbb{R}^n$ and $b_2 \in \mathbb{R}^{m-n}$ (transformation of b)
3. $Rx = b_1$ (solving by backwards substitution)

Remark (Implementation). *Usually, we store the Householder vectors $v_1, \dots, v_p$ in the lower half of A and the diagonal elements $v_{1i} = \alpha_i$ with $i = 1, \dots, p$ in a separate vector.*

Remark (Computing time).

- $\sim 2n^2 m$ *multiplications if* $m \gg n$
- $\frac{2}{3}n^3$ *if* $m \approx n$

It is important to recognize that this process is numerically stable!

3.3 Singular value decomposition

Literature:

- Gene H. Golub and Charles F. Van Loan: "Matrix computations"

If $A \in \mathbb{R}^{m \times n}$ has non-full rank n, then the linear equation system of $\min \|b - Ax\|$ is not unique, because $\forall \tilde{x} : \|b - A\tilde{x}\| = \|b - A(\tilde{x} + v)\| \; \forall v \in \ker(A)$.

Are there more decompositions of the matrix?

Recognize that the QR decomposition always exists anyways.

Theorem 3.7. *Let $A \in \mathbb{R}^{m\times n}$. There $\exists U \in \mathbb{R}^{m\times m}, V \in \mathbb{R}^{n\times n} : U^T U = I, V^T V = I$ and $A = U\Sigma V^T$ with $\Sigma \in \mathbb{R}^{n\times n}$ where $\Sigma = \operatorname{diag}(\sigma_1, \dots, \sigma_p), p = \min(m, n), V = [v_1, \dots, v_n]$ and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$.*

Remark 3.8. *σ_i are called "singular values" of A. u_i and v_i are called left- and right-singular vectors of A.*

↓ *This lecture took place on 2018/10/30.*

Proof. Let $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ such that $\|x\|_2 = 1 = \|y\|_2$ and $Ax = \sigma y$ where $\sigma = \|A\|_2$. Recognize that x, y exist because $\|A\|_2 = \sup_{\|\tilde{x}\|=1} \|A\tilde{x}\|_2 = \max_{\|\tilde{x}\|_2=1} \|A\tilde{x}\|_2$.

Hence, there exists x with $\|x\|_2 = 1$ and $\|Ax\|_2 = \|A\|_2$. Followingly, we define $y := \frac{Ax}{\|A\|_2}$ and it is true that $\sigma y = Ax$.

$$\|y\|_2 = \frac{\|Ax\|_2}{\|A\|_2} = 1$$

We construct orthonormal bases $\{x, v_2, \dots, v_n\}$ and $\{y, u_2, \dots, u_m\}$. The corresponding matrices $U = \begin{bmatrix} y & u_2 & \dots & u_m \end{bmatrix}$ and $V = \begin{bmatrix} x & v_2 & \dots & v_n \end{bmatrix}$ are orthogonal. $V^T V = I, U^T U = I$. Let $U_2 := (u_2, \dots, u_m)$ and $V_2 := (v_2, \dots, v_n)$. Furthermore,

$$U^T A V = \begin{bmatrix} y^T \\ U_2^T \end{bmatrix} A \begin{bmatrix} X & V_2 \end{bmatrix} = \begin{bmatrix} y^T A x & y^T A V_2 \\ U_2^T A x & U_2^T A V_2 \end{bmatrix} = \begin{bmatrix} \sigma & w^T \\ 0 & B \end{bmatrix} := A_1$$

where $w \in \mathbb{R}^{n-1}$ and $B \in \mathbb{R}^{m-1\times n-1}$ because $\left\| A_1 \left(\begin{bmatrix} \sigma \\ w \end{bmatrix} \right) \right\|_2^2 \geq (\sigma^2 + w^T w)$ it holds that $\|A_1\|_2^2 \geq (\sigma^2 + w^T W)$. On the other hand, $\sigma^2 = \|A\|_2^2 = \|A_1\|_2^2$. Thus $w = 0$. The statement then follows inductively. □

Properties of the singular value decomposition

Corollary 3.9. *If $A = U\Sigma V^T \in \mathbb{R}^{m\times n}$ and $m \geq n$ is the singular value decomposition of A, then $Av_i = \sigma_i u_i$ and $A^T u_i = \delta_i v_i$ for all $i = 1, \dots, n$.*

Proof. Consider columnwise the equality $AV = U\Sigma$ and accordingly, $A^T U = V\Sigma^T$. □

By the corollary, it immediately follows that

$$A^T A v_i = \sigma_i^2 v_i \qquad A A^T u_i = \sigma_i^2 u_i$$

hence the squares of singular values are the eigenvalues of the matrices $A^T A$, and accordingly AA^T.

Illustrative interpretation: The singular values of a matrix A give the length of the semi-axes of the ellipsis defined by $E = \{Ax \mid \|x\|_2 = 1\}$. The directions of semi-axes are denotes by u_i.

$$\|A\|_2^2 = \sup_{x \neq 0} \frac{\langle Ax, Ax \rangle}{\langle x, x \rangle} = \sup_{x \neq 0} \frac{\langle A^T Ax, x \rangle}{\langle x, x \rangle} = \lambda_{\text{max}}(A^T A)$$

Corollary 3.10. *Let $A \in \mathbb{R}^{m \times n}$. Then,*

$$\|A\|_2 = \sigma_1 \qquad \|A\|_F = \sqrt{\text{trace}(A^T A)} = \sqrt{\sigma_1^2 + \cdots + \sigma_p^2}$$

Proof. Consider that for spectral norm and Frobenius norm, it holds that $\|A\| = \|U^T AV\| = \|\Sigma\|$. □

Corollary 3.11. *If A has exactly r positive singular values, then $rK(A) = r$ and* $\ker(A) = \text{span}\{v_{r+1}, \dots, v_n\}$ *and* $\text{range}(A) = \text{span}\{u_1, \dots, u_r\}$.

Proof. Because U and V are regular, it holds that $\text{rank}(A) = \text{rank}(U\Sigma V^T) = \text{rank}(\Sigma) = r$.

The statements regarding the kernel and image of A result by Corollary 3.9. □

Corollary 3.12. *Let $A \in \mathbb{R}^{m \times n}$ and* $\text{rank}(A) = r$. *Then $A = \sum_{i=1}^r \sigma_i u_i v_i^T$.*

Proof. Again, consider the matrix notation of the singular value decomposition:

$$A = (U\Sigma)V^T = \begin{bmatrix} \sigma_1 u_1 & \sigma_2 u_2 & \dots & \sigma_r u_r & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix} = \sum_{r=1}^r \sigma_i u_i v_i^T$$

□

Theorem 3.13 (Eckart-Young-Mirsky)**.** *If $k < r = \text{rank}(A)$ and $A_k = \sum_{i=1}^k \sigma_i u_i v_i^T$, then* $\min_{\text{rank}(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}$.

Proof. Because $U^T A_k V = \text{diag}(\sigma_1, \dots, \sigma_k, 0, \dots, 0)$, $\text{rank}(A_k) = k$. Furthermore,

$$\|U^T(A - A_k)V\|_2 = \text{diag}(0, \dots, 0, \sigma_{k+1}, \dots, \sigma_p)$$

and therefore $\|A - A_k\|_2 = \sigma_{k+1}$. Now let $B \in \mathbb{R}^{m \times n}$ with $\text{rank}(B) = k$ arbitrary. Then we can find an orthonormal basis $x_1, \dots, x_{n-k}$ such that $\text{kernel}(B) = \text{span}\{x_1, \dots, x_{n-k}\}$. Furthermore it holds that $\text{span}\{x_1, \dots, x_{n-k}\} \cap \text{span}\{v_1, \dots, v_{k+1}\} \neq \{0\}$. Let z with $\|z\|_2 = 1$ of this intersection. Then $Bz = 0$ and $Az = A\sum_{i=1}^{k+1} \alpha_i v_i = \sum_{i=1}^{k+1} \sigma_i \alpha_i u_i$. Because $\|z\|_2 = 1$, $1 = \|z\|_2^2 = \left\|\sum_{i=1}^{k+1} \alpha_i v_i\right\|_2^2 = \sum_{i=1}^{k+1} \alpha_i^2$. Followingly,

$$\|A - B\|_2^2 \geq \|(A - B)z\|_2^2 = \|Az\|_2^2 = \sum_{i=1}^{k+1} \sigma_i^2 \alpha_i^2 \geq \sigma_{k+1}^2 \underbrace{\sum_{i=1}^{k+1} \alpha_i^2}_{=1} = \sigma_{k+1}^2$$

□

We go back to the topic of $\min_x \|b - Ax\|_2$.

Consider the equation $b \cdot Ax = d$ and we followingly want to minimize the norm of the residue $\|d\|_2$. Left-sided multiplication with U^T gives

$$U^T(b - Ax) = U^T d \implies U^T b - U^T A V V^T x = U^T d = \tilde{d}$$

Because of orthogonality of U, $\left\|\tilde{d}\right\|_2 = \|d\|_2$ and thus we can minimize $\left\|\tilde{d}\right\|_2$. For this we introduce auxiliary vectors $\tilde{x} = V^T x \in \mathbb{R}^n$ and $\tilde{b} = U^T b \in \mathbb{R}^m$. Because of the singular value decomposition, we retrieve the special form of $U^T b - U^T A V V^T x = U^T d = \tilde{d}$

$$\begin{aligned} -\sigma_i \tilde{x}_i + \tilde{b}_i &= \tilde{d}_i & i &= 1, \dots, r \\ \tilde{b}_i &= \tilde{d}_i & i &= r+1, \dots, m \end{aligned} \tag{3}$$

The last $m - r$ components are independent of $\tilde{x}_i$. The term $\|\tilde{a}\|_2^2 = \sum_{i=1}^m \tilde{d}_i^2$ will become minimal if $\tilde{d}_i = 0$ for $i = 1, \dots, r$. In this case it holds that

$$\min \left\|\tilde{d}\right\|_2^2 = \sum_{i=r+1}^m \tilde{d}_i^2 = \sum_{i=r+1}^m \tilde{b}_i^2 = \sum_{i=r+1}^m (u_i^T b)^2$$

The first r unknowns are given by

$$\tilde{x}_i = \frac{\tilde{b}_i}{\sigma_i} \qquad i = 1, \dots, r$$

because of equations 3 and simultaneously the last $n - r$ unknowns can be chosen freely. The solution vector x therefore has the following representation:

$$x = V\tilde{x} = \sum_{i=1}^n \tilde{x}_i v_i = \sum_{i=1}^r \frac{\tilde{b}_i}{\sigma_i} v_i + \sum_{i=r+1}^n \tilde{x}_i v_i = \sum_{i=1}^r \frac{u_i^T b}{\sigma_i} v_i + \sum_{i=r+1}^n \tilde{x}_i v_i$$

with the free parameters $\tilde{x}_{r+1}, \dots, \tilde{x}_n$. If matrix A does not have full column rank, then the generic solution is representable as sum of a particulate solution in the subspace of r right-singular vectors v_i to *positive* singular values σ_i and an arbitrary vector of kernel(A).

Often, the specific solution x^* with minimal Euclidean norm is used. Because of orthogonality, it is characterized by $\tilde{x}_i = 0$ for $i = r+1, \dots, n$ and we get

$$x^* = \sum_{i=1}^r \frac{u_i^T b}{\sigma_i} v_i \qquad \|x^*\|_2 = \min_{\|b - Ax\|_2^2} \|x\|_2$$

with $\|b - Ax\|_2^2 = \sum_{i=r+1}^n (u_i^T b)^2$.

Generalized inverse

Using the singular value decomposition, we can also give the inverse of a regular matrix $A \in \mathbb{R}^{n \times n}$, because $A^{-1} = (U\Sigma V^T)^{-1} = V\Sigma^{-1}U^T$. For arbitrary matrices

$A \in \mathbb{R}^{m \times n}$ with $\operatorname{rank}(A) = r$, we can define the *generalized inverse* by

$$A^\dagger = [v_1 \quad \dots \quad v_r] \operatorname{diag}\left(\frac{1}{\sigma_1}, \dots, \frac{1}{\sigma_r}\right) \begin{bmatrix} u_1^T \\ \vdots \\ u_r^T \end{bmatrix}$$

This gives the so-called *Moore-Penrose inverse*.

4 Interpolation

Problem setting: For an unknown function $f : \mathbb{R} \to \mathbb{R}$, construct from given data $f^{(j)}(t_i)$ for $i = 0, \dots, n$ and $j = 0, \dots, c_i$ a (efficiently computable) function φ which approximates f. We require the following properties:

Interpolation property at points t_i, φ and f correspond.

$$\varphi^{(j)}(t_i) = f^{(j)}(t_i) \qquad \forall i, j$$

We also call $f^{(j)}(t_i)$ *supporting points of the function*

Approximation property in a (yet unknown) function space it holds that $\|\varphi - f\| \approx 0$

4.1 Polynomial interpolation

First, we assume that only function values $f_i := f(t_i)$ for $t = 0, \dots, n$ at pairwise different points $t_0, \dots, t_n$ are given.

Goal: Find a polynomial $P \in \mathcal{P}_n$ of degree $\deg(P) \leq n$.

$$P(t) = a_n t^n + a_{n-1} t^{n-1} + \dots + a_n t + a_0$$

with $a_0, \dots, a_n \in \mathbb{R}$ such that f is interpolated at $t_0, \dots, t_n$, thus $P(t_i) = f_i$ for $i = 0, \dots, n$.

Uniqueness and condition

The values $a_0, \dots, a_n$ are uniquely determined by $P(t_i) = f_i$ for $i = 0, \dots, n$, because if $P, Q \in \mathcal{P}_n$ satisfy $P(t_i) = f_i$ then,

$$P(t_i) = Q(t_i) \text{ for } i = 0, \dots, n$$

and thus $R := P - Q \in \mathbb{P}_n$ for some polynomial with $n + 1$ zeros. Thus R is a zero polynomial, thus $P = Q$.

Now consider the map

$$\mathcal{P}_n \to \mathbb{R}^{n+1}, p \mapsto (p(t_0), \dots, p(t_n))$$

The map is linear and by property $\dim(\mathcal{P}_n) = n + 1 = \dim(\mathbb{R}^{n+1})$ with injectivity, surjectivity follows (e.g. by the rank theorem).

↓ *This lecture took place on 2018/11/06.*

Theorem 4.1. *For $n+1$ supporting points (Z_i, f_i) for $i = 0, \dots, n$ with pairwise different points $t_0, \dots, t_n$, there exists a unique interpolation polynomial P of degree n such that $P(t_i) = f_i \forall i$.*

This unique polynomial P is called interpolation polynomial of *f for points $p_0, \dots, p_n$ with $p = P(f \mid t_0, \dots, t_n)$.*

A proof for Theorem 4.1 is not provided.

To compute p, we need a basis of $\mathcal{P}_n$. If P is given in *coefficient representation* $(a_n t^n + \dots + a_1 t + a_0)$ with respect to monomial basis $\{1, t, \dots, t^n\}$, by $p(t_i) = f_i$ we get the linear equation system

$$\underbrace{\begin{bmatrix} t_i^j \end{bmatrix}}_{=:v_n} \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ \vdots \\ f_n \end{bmatrix} \qquad A = \begin{bmatrix} 1 & t_0^1 & t_0^2 & \dots \\ 1 & t_1^1 & t_1^2 & \dots \\ \vdots & & & \ddots \end{bmatrix}$$

V_n is called *Vandermonte matrix* and it holds:

Theorem 4.2. *Let $V_n \in \mathbb{R}^{n+1 \times n+1}$ be Vandermonte's matrix. Then*

$$\det(V_n) = \prod_{i=0}^{n} \prod_{j=i+1}^{n} (t_j - t_i)$$

Remark 4.3. *Obviously:* $\det(V_n) \neq 0 \iff t_i \neq t_j \forall i = j$

Alternative basis: Consider the Lagrange basis $L_0, \dots, L_n$ where these are the uniquely determined interpolation polynomials $L_i \in \mathcal{P}_n$ with $L_i(t_j) = \delta_{ij}$.

This leads to an explicit representation:

$$L_i(t) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{t - t_j}{t_i - t_j}$$

The interpolation polynomial P for arbitrary supporting points $f_0, \dots, f_n$ can be constructed by superposition of polynomials.

$$P(t) = \sum_{i=0}^{n} f_i L_i(t)$$

Thus,

$$P(t_j) = \sum_{i=0}^{n} f_i \cdot L_i(t_j) = \sum_{i=0}^{n} f_i \cdot \delta_{ij} = f_j$$

Remark 4.4 (Alternative characterization)**.** *The Lagrange polynomials create an orthonormal basis of $\mathcal{P}_n$ with respect to scalar product $\langle P, Q \rangle = \sum_{i=0}^{n} P(t_i) Q(t_i)$. Because $\langle P, L_i \rangle = P(t_i)$,*

$$P = \sum_{i=0}^{n} \langle P, L_i \rangle \cdot L_i = \sum_{i=0}^{n} P(t_i) \cdot L_i = \sum_{i=0}^{n} f_i \cdot L_i$$

Theorem 4.5. *Let $a \leq t_0 < \dots < t_n \leq b$. Let $L_{i,n}$ be the Lagrange polynomials. Then the absolute conditioning number κ_{abs} of polynomial interpolation $\Phi = P(\cdot \mid t_0, \dots, t_n) \in C[a,b] \to \mathcal{P}_n$ with respect to supremum norm is the so-called* Lebesgue constant $\kappa_{\text{abs}} := \Lambda_n = \max_{t \in [a,b]} \sum \left| L_{i,n}(t) \right|$ *for points $t_0, \dots, t_n$.*

Proof. Polynomial interpolation is linear

$$\implies \Phi'(f)(g) = \Phi(g)$$

Thus, show: $\|\Phi'\| = \Lambda_n$.

$$\begin{aligned}
\forall f \in C[a,b] : |\varphi(f)(t)| &= \left| \sum f(t_i) L_{i,n}(t) \right| \\
&\leq \sum |f(t_i)| \cdot \left| L_{i,n}(t) \right| \\
&\leq \|f\|_\infty \cdot \max_{t \in [a,b]} \sum \left| L_{i,n}(t) \right| \\
\implies \|\Phi\|_\infty = \kappa_{\text{abs}} &\leq \Lambda_n
\end{aligned}$$

In the other direction: We construct $g \in C[a,b]$:

$$|\Phi(g(\tau))| = \|g\|_\infty \cdot \max_{t \in [a,b]} \sum_{i=0}^{n} \left| L_{i,n}(t) \right|$$

Let τ be the specific point which takes this maximum.

Let $g \in C[a,b] : \|g\|_\infty = 1$ and $g(t_i) = \text{sign}(L_{i,n}(\tau))$. g solves the interpolation problem $(t_i, \text{sign}(L_{i,n}(t_i)))$.

$$\implies |\Phi(g)(\tau)|] = \sum \left| L_{i,n}(\tau) \right| = \|g\|_\infty \cdot \max_{t \in [a,b]} \sum_{i=0}^{n} \left| L_{i,n}(t) \right|$$

$$\implies \kappa_{\text{abs}} = \Lambda_n$$

□

n	Λ_n for equidistant points	Λ_n for Chebyshev points
5	3.11	2.104
10	29.9	2.48
15	512.05	2.727
20	10986.53	2.9

Chebyshev points:

$$t_i = \cos\left(\frac{2i+1}{2n+2} \cdot \pi \right)$$

Polynomial evaluation by Aitken-Neville

Motivation: Often we are interested in $P(t)$ for a few t and not in t itself.

Lemma 4.6 (Aitken's lemma)**.** *For the interpolation polynomial $P = P(f \mid t_0, \dots, t_n)$, the following recursion formula holds:*

$$P(f \mid t_0, \dots, t_n) = \frac{(t_0 - t)P(f \mid t_1, \dots, t_n) - (t_n - t)P(f \mid t_0, \dots, t_{n-1})}{t_0 - t_n}$$

Proof. Let us denote the function of the right with φ. φ is a polynomial of degree n. Thus $\varphi \in \mathcal{P}_n$. Furthermore,

$$\phi(t_i) = \frac{(t_0 - t_i) \cdot f_i - (t_n - t_i) f_i}{t_0 - t_n} = f_i \forall i \in \{1, \dots, n-1\}$$

For $i = 0$ or $i = n$:

$$\varphi(t_0) = -\frac{t_n - t_0}{t_0 - t_n} f(t_0)$$
$$\varphi(t_n) = \frac{t_0 - t_n}{t_0 - t_n} f(t_n)$$

The interpolation polynomial is unique. Thus we can conclude the claim. □

Aitken's lemma provides a recursive method for evaluation of $P(f \mid t_0 \dots t_n)$ at point t.

Notation: $P(f(t_i)) = f_i$, $P_{i,k} \coloneqq P(f, t_{i-k} \dots t_i)(t), i \geq k$.

Neville's scheme: where $P_{i0} = f_i$ and

	P_{i0}	P_{i1}	P_{i2}	...	P_{in}
t_0	P_{00}				
t_1	P_{10}	P_{11}			
t_n	P_{n0}	P_{n1}	...		P_{nn}

$$P_{ik} = \frac{t - t_{i-k}}{t_i - t_{i-k}} P_{i,k-1} + \frac{t_i - t}{t_i - t_{i-k}} P_{i-1,k-1} = P_{i,k-1} + \frac{t - t_i}{t_i - t_{i-k}} (P_{i,k-1} - P_{i-1,k-1})$$

How many operations are necessary?

$k = 1$**:** $6n$

$k = 2$**:** $6(n-1)$

$k = n$**:** 6

In total, we get

$$6 \cdot \sum_{i=1}^{n} i = 6 \frac{n(n-1)}{2} = \mathcal{O}(n^2)$$

If there is one additional supporting point, the computing time is increased by $\mathcal{O}(n)$.

Now we allow, that points repeat, thus $a \leq t_0 \leq \dots \leq t_n \leq b$.

$$\triangle := \{t_i\}_{i=0}^n$$

If the derivatives $f'(t_i), \dots f^{(k)}(t_i)$ are given, t_i is expected to occur $k+1$ times.

$$d_i := \max\left\{j \mid t_i = t_{i-j}\right\}$$

For example:

t_i	t_0	$= t_1$	$< t_2$	$= t_3$	$= t_4$	$< t_5$	$< t_6$
d_i	0	1	0	1	2	0	0

Once we define $\forall i : \mu_i : C^n[a,b] \to \mathbb{R}$, $\mu_i(f) = f^{(d_i)}(t_i)$, we can define the goal of Hermitian interpolation:

We construct a polynomial $P \in \mathcal{P}_n$ satisfying $\mu_i(P) = \mu_i(f) \forall i = 0, \dots, n$.

↓ *This lecture took place on 2018/11/08.*

Remark (revision of Hermitian interpolation). *$i = 0, \dots, n$ with $\mu_i : C^n([a,b]) \to \mathbb{R}^n$ with $\mu_i(f) := f^{(d_i)}(t_i)$*

The solution $p = P(f(t_0, \dots, t_n)) \in \mathcal{P}_n$ is called Hermitian interpolant *of f at $t_0, \dots, t_n$.*

Theorem 4.7. *For every function $f \in \mathcal{C}^n([a,b])$ and every monotonic sequence $a \leq t_0 \leq t_1 \leq \dots \leq t_n \leq b$ of (not necessarily different) points there exists exactly one polynomial $P \in \mathcal{P}_n$ such that $\forall i \in \{0, \dots, n\} : \mu_i(P) = \mu_i(f)$.*

Proof. A map $\mu : \mathcal{P}_n \to \mathbb{R}^{n+1}$ with $P \mapsto (\mu_0 P, \dots, \mu_n P)$ is linear and injective. By $\mu(P) = 0$, P has at least $n+1$ roots (with multiplicities); so its the zero polynomial. Because $\dim(\mathcal{P}_n) = \dim(\mathbb{R}^{n+1}) = n+1$, existence follows. □

If all points are pairwise different, we retrieve the Lagrange interpolant

$$P(f \mid t_0, \dots, t_n) = \sum_{i=0}^n f(t_i) L_i$$

However, if $t_0 = t_1 = \dots = t_n$, the interpolant is the aborted Taylor series around $t = t_0$

$$P(f \mid t_0, \dots, t_n)(t) = \sum_{j=0}^n \frac{(t-t_0)^j}{j!} f^{(j)}(t_0)$$

is also called *Taylor interpolant.*

Lemma 4.8. *Assuming $t_i \neq t_j$, the Hermitian interpolation polynomial $P = P(f(t_0, \dots, t_n))$ satisfies*

$$P = \frac{(t_i - t)P(f \mid t_0, \dots \hat{t}_i, \dots, t_n) - (t_j - t)P(f \mid t_0, \dots, \hat{t}_j, \dots, t_n)}{t_i - t_j}$$

where $\hat{t}_i$ means that the i-th point is skipped.

Proof. We prove the interpolation properties by insertion of the definition.

If many evaluations of polynomials are required, it is advantageous to determine polynomial P explicitly. Therefore, choose the Newton basis $w_0, \dots, w_n$ of $\mathcal{P}_n : w_0(t) := 1$ and $w_i(t) := \prod_{j=0}^{i-1}(t - t_j), i \geq 1$. □

Definition 4.9. *The leading coefficient a_n of the interpolation polynomial $P(f \mid t_0, \dots, t_n) = a_n t^n + a_{n-1}t^{n-1} + \dots + a_0$ of f for points $t_0 \leq t_1 \leq \dots \leq t_n$ is called n-th* divided difference *of f at $t_0, \dots, t_n$ and is denoted as $[t_0, \dots, t_n]f := a_n$.*

Theorem 4.10. *For every function $f \in \mathcal{C}^n([a, b])$ and points $t_0, \dots, t_n$, $P := \sum_{i=0}^{n}[t_0, \dots, t_i]f \cdot w_i$ defines the interpolation polynomial $P(f \mid t_0, \dots, t_n)$ of f at $t_0, \dots, t_n$. If $f \in \mathcal{C}^{n+1}([a, b])$, then $f(t) = P(t) + [t_0, \dots, t_n, t]f \cdot w_{n+1}(t)$.*

Proof. Proof by induction over n.

For $n = 0$, we get value $[t_0]f = f(t_0)$ as leading coefficient (at points t_0). The constant interpolation polynomial P satisfies $p := [t_0]f \cdot w_0 = f(t_0) \cdot 1$. This is shown by the first statement for $n = 0$. To determine $[t_0, t]f$, compute the leading coefficients of the interpolation polynomial at points t_0 and t. For this polynomial it holds true that

$$\begin{aligned} P(f \mid t_0, t) &= a + a_1 t =: p(t) \\ \implies f(t_0) &= p(t_0) = a_0 + a_1 t_0 \\ f(t) &= p(t) = a_0 + a_1 t \\ \implies a_1 &= [t_0, t]f = \frac{f(t) - f(t_0)}{t - t_0} \end{aligned}$$

This implies

$$P(t) = [t_0]f \cdot w_0 + [t_0, t]f \cdot w_1(t) = f(t_0) \cdot 1 + \frac{f(t) - f(t_0)}{t - t_0}(t - t_0) = f(t)$$

So the statement for case $n = 0$ follows.

Now let $n > 0$ and

$$P_{n-1} := P(f \mid t_0, \dots, t_{n-1}) = \sum_{i=0}^{n-1}[t_0, \dots, t_i]f \cdot w_i]$$

the interpolation polynomial of f at $t_0, \dots, t_{n-1}$. For $P_n = P(f(t_0, \dots, t_n))$, we have

$$P_n(t) = [t_0, \dots, t_n]f \cdot t^n + a_{n-1}t^{n-1} + \dots + a_0 = [t_0, \dots, t_n]f \cdot w_n(t) + Q_{n-1}(t)$$

with some polynomial $Q_{n-1} \in \mathcal{P}_{n-1}$. Because $Q_{n-1} = P_n - [t_0, \dots, t_n]f \cdot w_n$, Q_{n-1} satisfies the interpolation requirements for $t_0, \dots, t_{n-1}$. Therefore

$$Q_{n-1} = P_{n-1} = \sum_{i=0}^{n-1}[t_0, \dots, t_i]f \cdot w_i$$

This show the first statement. Especially, $P_n + [t_0, \dots, t_n, t]f \cdot w_{n+1}$ interpolates the function f at points $t_0, \dots, t_n, t$. So the second statement follows. □

The following properties are given:

Theorem 4.11. *Dividing differences $[t_0, \dots, t_n]f$ satisfying (assuming $f \in C^n([a,b])$)*

1. $[t_0, \dots, t_n]P = 0$ *for all* $P \in \mathcal{P}_{n-1}$
2. *For collapsing points* $t_0 = \dots = t_n$, *we have* $[t_0, \dots, t_n]f = \frac{f^{(n)}}{n!}$
3. *For* $i < j$, *we have the recursive formula* $[t_i, \dots, t_j]f = \frac{[t_{i+1}, \dots, t_j]f - [t_i, \dots, t_{j-1}]f}{t_j - t_i}$

↓ *This lecture took place on 2018/11/13.*

Theorem 4.12 (Hermite-Gnocci Formula)**.** *For the n-th dividing difference of an n-times differentiable function $f \in \mathcal{C}^n$, we have*

$$\sum^n := \left\{ s = (s_0, \dots, s_n) \in \mathbb{R}^{n+1} \mid \sum s_i = 1, s_i \geq 0 \right\}$$
$$[t_0, \dots, t_n]f = \int_{\sum^n} f^{(n)} \left(\sum_{i=0}^n s_i t_i \right) ds$$

No proof is given of this theorem.

Approximation error

Theorem 4.13. *If $f \in \mathcal{C}^{n+1}$, for the interpolation polynomial $P(f(t_0, \dots, t_n))$ and $t_i, t \in [a,b]$ we have:*

$$f(t) - P(f \mid t_0, \dots, t_n)(t) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \cdot w_{n+1}(t) \text{ with } \xi = \xi(t) \in (a,b)$$

Proof. Let $t \in [a,b]$. For $t = t_i$, both sides are zero. Thus, we assume $t \neq t_i \forall i$.

For $\alpha \in \mathbb{R}$ we define that $g(\tau) := f(\tau) - P(f \mid t_0, \dots, t_n)(\tau) - \alpha w_{n+1}(\tau)$. g is $(n+1)$-times continuous differentiable and has roots $t_0, \dots, t_n$. Choose

$$\alpha = \frac{f(t) - P(f \mid t_0, \dots, t_n)(t)}{(t - t_0) \cdot \dots \cdot (t - t_n)}$$

Thus, α is a root of g. This implies there are $n+2$ roots.

By Rolle's Theorem, g' has $n+1$ roots and consider this iteratively, we have exactly one root $\xi \in [a,b]$ for $g^{(n+1)}$. For ξ, we have:

$$\begin{aligned} 0 &= g^{(n+1)}(\xi) \\ &= f^{(n+1)}(\xi) - \frac{f(t) - P(f \mid t_0, \dots, t_n)(t)}{(t - t_0) \dots (t - t_n)} \cdot (n+1)! \end{aligned}$$

Because P is a polynomial of degree $n+1$ and w_{n+1} is a polynomial with leading coefficient 1.

If we resolve this by $P(f \mid \dots)(t)$ shows our claim. □

Remark 4.14. *In case of Taylor interpolation* $(t_0 = \cdots = t_n)$ *we have the Lagrange remainder term of the Taylor expansion*

$$f(t) - P(f \mid t_0 \dots t_n)(t) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(t - t_0)^{n+1}$$

Consider the following class of functions:

$$\mathcal{F} := \left\{ f \in \mathcal{C}^{n+1}[a,b] \mid \sup_{x \in [a,b]} \left| f^{(n+1)}(x) \right| \leq M(n+1)! \right\}$$

for $M > 0$ constant, the approximation error depends on the choice of point t with respect to $w_{n+1}(t) = (t - t_0) \cdot \dots \cdot (t - t_0)$.

Remark 4.15 (An excursion: Numeric differentiation)**.** *Approach:*

1. *Choose points* $t_0, \dots, t_n$.
2. *Let* f *approximate this polynomial* $P \in \mathcal{P}_n$
3. *Differentiate* $P \to P'$
4. *Can be determined accurately such that the interpolation error can determine the accuracy of the approximation*

Consider $P(f \mid t_i)$ in Newton form:

$$P = f(t_0) + [t_0, t_1] f w_1 + \dots + [t_0 \dots t_n] f w_n$$

The approximation error is given by

$$r(t) = w_{n+1}(t) \cdot \frac{f^{(n+1)}(\xi(t))}{(n+1)!}$$

If $f \in \mathcal{C}^{n+2}$, we can differentiate both equation systems.

$$t = t_0 : P'(t_0) = [t_0, t_1] f + [t_0, t_1, t_2] r \cdot (t_0 - t_2) + \dots + [t_0 \dots t_n] f (t_0 - t_1) \dots (t_0 - t_{n-1})$$

$$r'(t_0) = (t_0 - t_n) \frac{f^{(n+1)}(\xi(t_0))}{(n+1)!} \qquad w_{n+1}^{(k)}(t_0) = 0 \forall k$$

Thus, we get:

$$f'(t_0) = P'(t_0) + r'(t_0) \qquad \text{(analogously for } t_j \neq t_0\text{)}$$

If the supporting points are distributed with distance $k > 0$ next to t_0, we get the classical differential quotient:

Example 4.16. Forward differential quotient *for the first derivative:* $t_1 := t_0 + k$

$$\begin{aligned} f'(t_0) &= P_1'(t_0) + r_1'(t_0) \\ &= [t_0, t_1] f + (t_0 - t_1) \cdot \frac{1}{2} f^n(\xi) \\ &= \frac{f(t_0 + h) - f(t_0)}{h} - \frac{n}{2} \cdot f''(\xi) \end{aligned}$$

The resulting error is in $\mathcal{O}(h)$ *for* $h \to 0$

Central differential quotient $t_0 = t_1 - k, t_1, t_2 = t_1 + k$.

The three supporting points define a polynomial of second degree, we can use to determine $f'(t_2)$.

$$\implies f'(t_1) = \underbrace{P'(t_1)}_{\deg=2} + r_2(t_1) = [t_0, t_1]f + [t_0, t_1, t_2]f \cdot \underbrace{(t_1 - t - 0)}_{=h} + (t_1 - t_0)(t_1 - t_2) \cdot \frac{1}{6} \cdot f'''(\xi)$$

$$= \frac{f(t_1) \cdot f(t_0)}{h} + \frac{1}{2n}\left(f(t_2) - 2f(t_2) + f(t_0)\right)$$

$$= \frac{1}{2h}(f(t_1 + h) - f(t_1 - h)) - \frac{h^2}{6} f'''(\xi)$$

where

$$[t_0, t_1, t_2]f = \frac{[t_1, t_2]f - [t_0, t_1]f}{t_2 - t_0} = \frac{1}{2h}\left(\frac{f(t_2) - f(t_1)}{h} - \frac{f(t_1) - f(t_0)}{h}\right)$$

The error is in $\mathcal{O}(h^2)$ for $h \to 0$.

Thus, symmetric differential quotients usually give a better approximation then the one-sided quotient.

Minmax property of the Chebyshev polynomial

(see page 214 in the book)

Our goal is to show that $\max_{t \in [a,b]} |w_{n+1}(t)|$ is minimized using Chebyshev points. We consider the *Chebyshev polynomials*

$$T_n = \cos(n \cdot \arccos(x)) \text{ for } x \in [-1, 1)$$

For $x \in \mathbb{R}$, we define recursively,

$$T_k(x) = 2x \cdot T_{k-1}(x) \cdot T_{k-2}(x) \qquad T_0(x) = 1 \qquad T_1(x) = x$$

- The Chebyshev polynomials have integral coefficients.
- The largest coefficient of T_n is $a_n = 2^{n-1}$
- T_n is even if n is even, otherwise it is odd
- $T_n(1) = 1, T_n(-1) = (-1)^n$
- $|T_n(x)| \leq 1 \forall x \in [-1, 1]$
- $|T_n(x)|$ takes the value 1 at the so-called *Chebyshev x-coordinates* (dt. "Tschebyscheff Abszissen") $\tilde{x}_k = \cos(\frac{k\pi}{2})$

$$\implies |T_n(x)| = 1 \iff x = \tilde{x}_k = \cos(\frac{k\pi}{2}) \text{ for } k = 1, \dots, n$$

- T_k create an orthogonal basis over $[-1, 1]$ to the weight function $w(x) = \frac{1}{\sqrt{1-x^2}}$

- Global representation:

$$T_k(x) = \frac{1}{2}\left((x+\sqrt{x^2-1})^k + (x-\sqrt{x^2-1})^k\right) \qquad \text{for } x \in \mathbb{R}$$

In terms of the approximation error, we look for $q \in \mathcal{P}_{n+1}$ of degree $n+1$ with leading coefficient 1 and the smallest supremum norm over some interval $[a,b]$.

$$\implies \max_{t\in[a,b]} |q(t)| \to \min$$

$\implies$ reduce the problem to interval $[-1,1]$.

We use the following map:

$$x : [a,b] \to [-1,1] \qquad t \mapsto x(t) = 2\frac{t-a}{b-a} - 1$$

Inverse function:

$$t(x) = \frac{1-x}{2}a + \frac{1+x}{2}b$$

If $q \in \mathcal{P}_{n+1}$ with $\deg(q) = n+1$ and leading coefficient 1 is the solution of the min-max problem

$$\max_{x\in[-1,1]} |q(x)| \to \min$$

then $\hat{q}(t) = q(t(x))$ is the solution of the original problem with leading coefficient $\frac{2^{n+1}}{(b-a)^{n+1}}$.

Theorem 4.17. *Every polynomial $q_n \in \mathcal{P}_n$ with leading non-zero coefficient takes up in interval $[-1,1]$ a value with absolute value $\geq \frac{|a_n|}{2^{n-1}}$.*

In particular, the Chebyshev polynomials $T_n(x)$ are minimal with respect to maximum norm $\|f\|_\infty = \max_{x\in[-1,1]} |f(x)|$ and the polynomials of degree n with leading coefficient 2^{n-1}.

Proof. Let $q_n \in \mathcal{P}_n$ with leading coefficient $a_n = 2^{n-1}$. Remark: $|q_n(x)| < 1 \forall x \in [-1,1]$.

$T_n - q_n$ has degree $\leq n-1$. At the Chebyshev x-coordinates, we have $T_n(\tilde{x}_{2k}) = 1$ and $q_n(\tilde{x}_{2k}) < 1$. $T_n(\tilde{x}_{2k+1}) = -1$ and $q_n(\tilde{x}_{2k+1}) > -1$.

$T_n - q_n$ at the $n+1$ Chebyshev x-coordinates is alternatingly positive and negative, thus there are $\geq n$ roots.

But $\deg \leq n-1 \implies T_n - q_n \equiv 0$ is a contradiction.

$$\implies \forall q_n \in \mathcal{P}_n \text{ with } a_n = 2^{n-1}$$
$$\exists x \in [-1,1] : |q_n(x)| \geq 1$$

For $q_n \in \mathcal{P}_n$ with $a_n \neq 0$, the following claim follows: $\tilde{q}_n = q_n \cdot \frac{2^{n-1}}{a_n}$ is a polynomial with leading coefficient 2^{n-1}. □

Remark 4.18. *When minimizing the approximation error of the polynomial interpolation, we look for the points $t_0, \dots, t_n \in [a, b]$ that solve the min-max problem* $\max_{t\in[a,b]} |w(t)| = \max |(t - t_0) \cdot \dots \cdot (t - t_n)| \to \min$.

We look for the normed polynomial $w_n(\cdot) \in \mathcal{P}_n$ with real roots $t_0, \dots, t_n$: $\max_{t\in[a,b]} |w(t)| \to \min$.

$\implies$ *for $[a, b]$, this is the $n + 1$-th Chebyshev polynomial with roots $t_i = \cos\left(\frac{2i+1}{2n+2} \cdot \pi\right)$ with $i = 0, \dots, n$.*

↓ *This lecture took place on 2018/11/15.*

4.2 Trigonometric interpolation

Motivation: The interpolation of periodic processes (e.g. waves) it is useful to use periodic functions. Thus for this section, we assume $f(x) = f(x \cdot 2\pi)$.

Idea: Consider interpolation polynomials on the complex unit circle by variable transformation $z : [0, 2\pi] \to \mathbb{C}, t \mapsto e^{it}$.

Consider polynomial P_n,

$$\begin{aligned} P_n(z(t)) &= \sum_{k=0}^{n-1} c_k (z(t))^k \\ &= \sum_{k=0}^{n-1} c_k \cdot e^{itk} =: p(t) \end{aligned}$$

We call P *trigonometric polynomial* and consider the following problem:

Given $t_j = 2\pi \frac{j}{n}$ and points $f_j \in \mathbb{C} : 0 \le j < n$. Find c_k with $0 \le k < n$: $\sum_{k=0}^{n-1} c_k \cdot e^{2\pi i k \frac{j}{n}} = f_j$ with $j = 0, \dots n - 1$.

In matrix representation, we get $M \cdot c := [c_0 \quad \dots \quad c_{n-1}] = [f_0 \quad \dots \quad f_{n-1}]$ with $M \in \mathbb{C}^{n\times n}$,

$$M = (m_{kj})_{k,j=0}^{n-1} \qquad m_{kj} = e^{2\pi i k \frac{j}{n}}$$

Pay attention that a different indexing is used here.

The interpolation polynomial has a unique solution if M is invertible and

$$M^{-1} = \frac{1}{n} \cdot M^* = \frac{1}{n} \cdot \overline{M}^t$$

Recognize that $M^*_{kj} = \overline{M}_{jk} = e^{-2\pi i j \frac{k}{n}}$.

Now consider the entries $M * M$:

$$
\begin{aligned}
(M^* M)_{k,j} &= \sum_{l=0}^{n-1} \overline{M}_{lk} \cdot M_{lj} \\
&= \sum_{l=0}^{n-1} e^{-2\pi i l \frac{k}{n}} \cdot e^{2\pi i l \frac{j}{n}} \\
&= \sum_{l=0}^{n-1} e^{-2\pi i l \frac{k-j}{n}} \\
&= \begin{cases} \sum_{l=0}^{n-1} 1 & \text{if } k = j \\ \sum_{l=0}^{n-1} \left(e^{-2\pi i \frac{k-j}{n}}\right)^l & \text{else} \end{cases} \qquad = \frac{1 - e^{-2\pi i (k-j)}}{1 - e^{-2\pi i \frac{k-j}{n}}}
\end{aligned}
$$

which is a geometric sum.

For the enumerator, we have:

$$
\begin{aligned}
\exp(-2\pi i(k-j)) = \exp(-\pi i)^{2(k-j)} \qquad &= 1 - ((-1)^2)^{k-j} = 0 \\
\implies M * M = n \cdot I &
\end{aligned}
$$

The unique solution of the interpolation problem is given by

$$
c = \frac{1}{n} \cdot M^* \cdot f \qquad \text{or} \qquad c_k = \sum_{j=0}^{n-1} f_j \cdot e^{-2\pi i \frac{j-k}{n} \frac{1}{n}}
$$

This process is called discrete Fourier Transform (dFT)

The inverse Fourier transform corresponds to the polynomial evaluation $f_i = \sum \dots$.

Fast Fourier transform

Disadvantage of discrete Fourier transform: The computational requirements $\mathcal{O}(n^2)$ are quite high. The computation can be sped up with fast Fourier Transform.

In the following we assume that n is even. For the unit root $w_n^k := \exp(-2\pi i \frac{k}{n})$, we have $w_{2n}^{2k} = w_n^k = w_{\frac{n}{2}}^{\frac{k}{2}}$. Decompose the components of the Discrete Fourier Transform in its even and odd parts:

$$
\begin{aligned}
n \cdot c_k &= \sum_{j \text{ even}} w_n^{kj} f_j + \sum_{j \text{ odd}} w_n^{kj} f_j \\
&= \sum_{l=0}^{\frac{n}{2}-1} w_n^{k2l} f_{2l} + \sum_{l=0}^{\frac{n}{2}-1} w_n^{k(2t+1)} f_{2l+1} \\
&= \sum_{l=0}^{\frac{n}{2}-1} w_{\frac{n}{2}}^{kl} f_{2l} + w_n^k \sum_{l=0}^{\frac{n}{2}-1} w_{\frac{n}{2}}^{kl} f_{2l+1} \\
&=: \tilde{c}_k + w_n^k \tilde{\tilde{c}}_k
\end{aligned}
$$

For $k \leq \frac{n}{2} - 1$, $\tilde{c}_k$ give the Discrete Fourier Transform for the $\frac{n}{2}$ values $f_0, f_2, \dots f_{n-2}$. $\tilde{\tilde{c}}_k$ give the Discrete Fourier Transform for the $\frac{n}{2}$ values $f_1, f_3, \dots, f_{n-1}$.

For those k we can c_k compute by two DFT by length $\frac{n}{2}$. The remaining c_k ($k > \frac{n}{2} - 1$) can alse be expressed with $\tilde{c}_k$ and $\tilde{\tilde{c}}_k$, because for $k \geq \frac{n}{2}$, and accordingly $k = \tilde{k} + \frac{n}{2}$, with $0 \leq \tilde{k} \leq \frac{n}{2} - 1$, we have:

$$w_n^{2lk} = w_n^{2l\tilde{k}+ln} = w_n^{2l\tilde{k}} \cdot (\underbrace{w_n^n}_{=1})^l$$

$$\begin{aligned} w_n^k &= w_n^{\tilde{k}+\frac{n}{2}} \\ &= w_n^{\tilde{k}} \cdot w_n^{\frac{n}{2}} \\ &= w_n^{\tilde{k}} \cdot e^{-\pi i} \\ &= -w_n^{\tilde{k}} \end{aligned}$$

So for $k = \tilde{k} + \frac{n}{2} \geq \frac{n}{2} - 1$:

$$nc_k = \tilde{c}_k - w_n^{\tilde{k}} \tilde{\tilde{c}}_k$$

If the unit roots are precomputed, the computational effort is given by $A(n)$ for some DFT of length n by 2 DFTs of length $\frac{n}{2}$ and n additions and $\frac{n}{2}$ multiplications:

$$A(n) = 2A(\frac{n}{2}) + \frac{3}{2}n$$

If $\frac{n}{2}$ is even again, we can solve the DFT by two recursions. For $n = 2^n$:

$$A(n) = 2n \cdot \log_2(n)$$

Algorithm 4. Input: $f_0, \dots, f_{n-1}$, $n = 2^m$

1. *If* $n = 1$: $c_0 = f_0$ *end*
2. $(a_0, \dots, a_{\frac{n}{2}}) = (f_0, f_2, \dots, f_{n-2})$
3. $(b_0, \dots, b_{\frac{n}{2}}) = (f_1, f_3, \dots, f_{n-1})$
4. *for* $k = 0, \dots, \frac{n}{2} - 1$
   (a) $c_k = a_k + w_n^k b_k$
   (b) $c_{k+\frac{n}{2}} = a_k - w_n^k b_k$

Output: $c_0, \dots, c_{n-1}$

↓ *This lecture took place on 2018/11/20.*

Problem setting: $f_j = f(t_j) \in \mathbb{R} \forall j$. Can we create a pure interpolation polynomial?

We restrict ourself to $n = 2p + 1$ and reenumerate the coefficients of the complex interpolation polynomial from $c_0, \dots, c_{n-1}$ to $c_0, \dots, c_p, c_{-p}, \dots, c_{-1}$. Recognize that

$$e^{2\pi i \frac{n-1}{n}} = e^{-2\pi i \frac{1}{n}} \qquad \dots \qquad e^{2\pi i \frac{p+1}{n}} = e^{-2\pi i \frac{p}{n}}$$

Thus, the interpolation polynomial can be written as

$$f(t_j) = \sum_{k=-p}^{p} c_k \cdot e^{ikt_j}$$

Because $f(t_j) \in \mathbb{R}$ it follows that $f(t_j) = \overline{f(t_j)}$ and thus

$$\sum_{k=-p}^{p} c_k \cdot e^{ikt_j} = \sum_{k=-p}^{p} \overline{c_k} e^{-ikt_j} \implies \overline{c_k} = c_{-k} \text{ for } k = 1, \dots, p \text{ and } c_0 \in \mathbb{R}$$

If we write $c_k = c_k^r + ic_k^i$, we get

$$\begin{aligned} f(t_j) &= c_0 + \sum_{k=1}^{p} (c_k e^{ikt_j} + \overline{c}_k e^{-ikt_j}) \\ &= c_0 + \sum_{k=1}^{p} \left[(c_k^r + ic_k^i)(\cos(kt_j) + i \cdot \sin(kt_j)) + (c_k^r - ic_k^i)(\cos(kt_j) + i \sin(kt_j)) \right] \\ &= c_0 + \sum_{k=1}^{p} (2c_k^r \cos(kt_j) - 2c_k^i \sin(kt_j)) \end{aligned}$$

Therefore, the unique interpolation polynomial is a linear combination of $\{1, \cos(kt), \sin(kt) : k = 1 : \frac{n-1}{2}\}$ and the coefficients can be computed by complex Fourier transform.

4.3 Splines

Motivation: For high polynomial degrees, usually high oscillations occur. This is inappropriate for interpolation.

Idea: Split up the interval in smaller intervals, solve the problem locally.

Problem: In general, only $\mathcal{C}^0$, but not smooth.

Spline functions satisfy the following three conditions:

- Piecewise polynomials of highest degree k
- p-times continuously differentiable on the entire interval
- The curvature is minimized

In practice, *cubic splines* will occur most often.

Problem setting: Let $n + 1$ pairwise different points $a = t_0 < t_1 < \dots < t_b = b$ with function values $y_0, \dots, y_n$ be given. Find a function $g : [a, b]$ that satisfies:

1. $s(t_i) = y_i \forall i$
2. $s \in \mathcal{C}^1([a,b])$
3. $s \in \mathcal{C}^k([t_i, t_{i+1}])$
   For the derivative there exist left- and rightsided limits in the points t_i, that must not necessarily correspond.
4. s minimizes the functional $J[s] := \frac{1}{2}\int_a^b (s'')^2\,dt$

Theorem 4.19.

$$\exists! s : s \textit{ satisfies the four previous properties}$$

Proof. First, we assume the existence of s and then we derive necessary optimality conditions that lead to a unique solvable linear equation system.

Let s be a function that satisfies 1–4, let s_1 be a function satisfying 1–3.

$$\implies J[s] \leq J[s_1]$$

So we write $s_1(t) = s(t) + \varepsilon k(t)$ with parameter $\varepsilon \in \mathbb{R}$. Because $\varepsilon^h = s_1 - s$, h satisfies 2 and 3, $h(t_i) = 0 \forall i$.

Every h that satisfies this condition, is considered *reliable*, because it gives some s_1 with condition 1–3.

Now consider $J[s] := J[s_1] = \frac{1}{2} \cdot \int_a^b (s''(t) + \varepsilon h''(t))^2\,dt$ with reliable h.

Therefore $J(\varepsilon)$ is a quadratic polynomial in ε, that takes up its minimum in $\varepsilon = 0$.

Thus our necessary optimality condition is $J'(0) = \frac{dJ}{d\varepsilon}\Big|_{\varepsilon=0} = 0$ for every reliable function h. Furthermore it is true that

$$J'(0) = \int_a^b s''(t) \cdot h''(t)\,dt$$

Two times partial integration over the subintervals (t_i, t_{i+1}) gives

$$\begin{aligned}\int_{t_i}^{t_{i+1}} s''(t)h''(t)\,dt &= s''(t)h'(t)\Big|_{t_i}^{t_{i+1}} - \int_{t_i}^{t_{i+1}} s^{(3)}(t) \cdot h'(t)\,dt \\ &= s''(t)h'(t)\Big|_{t_i}^{t_{i+1}} - s^{(3)}(t) \cdot h(t)\Big|_{t_i}^{t_{i+1}} \\ &+ \int_{t_i}^{t_{i+1}} s^{(k)}(t)h(t)\,dt\end{aligned}$$

with $h(t_i) = 0 \forall i$.

After summing up all subsums, we get

$$\begin{aligned}0 = J'(0) = -s''(t_0^+)h'(t_0) + s''(t_1^-)h'(t_1) + s''(t_2^-) - s''(t_2^+)h'(t_2) + \\ \cdots + s''(t_n^-)h'(t_n) + \int_a^b s^{(k)}(t)h(t)\,dt\end{aligned}$$

We want to achieve that $s^{(k)}(t) = 0 \forall t$. Thus for $\xi : s^{(k)}(\xi) \neq 0$ construct $h : h(\xi) \neq 0$ and $ht \neq 0$

By $J'(0) = 0$ follows that $s^{(k)}(t) = 0 \forall t \neq t_i \forall i$. In the other case, if $s^{(k)}(\xi) > 0$ for $\xi \in (t_i, t_{i+1})$, we construct some reliable h with $h(x)$ only in some neighborhood of ξ.

$$h(x) = \begin{cases} \exp\left(\frac{\delta^2}{x^2-\delta^2}\right) & |x| < \delta \\ 0 & \text{otherwise} \end{cases} \qquad \text{shifted by } \xi$$

$g^{(k)}$ is continuos, so also in some neighborhood of $\xi + 0$, so the integral > 0. This gives a contradiction. Furthermore it follows that $s''(t_0^+) = s''(t_0^-) = 0$ and $s''(t_i^-) = s''(t_i^+) \forall i = 1, \dots, n-1$. Thus we get $0 = J'(0)$ like above. Otherwise we create an appropriate h-function.

By the first conclusion, we get that s can be represented as polynomial of third degree in every interval $[t_i, t_{i+1}]$. By property 1, 2 and the second conclusion we get

$$s_i(t_i) = s_{i-1}(t_i) \forall i = 1, \dots, n-1 \qquad s_i'(t_i) = s_{i-1}'(t_i) \qquad s_i''(t_i) = s_{i-1}''(t_i)$$
$$s_0'(t_0) = s_{n-1}'(t_n) = 0$$

The spline function is a cubic, piecewise polynomial in $C^2([a, b])$.

It remains to show the uniqueness of s.

In every subinterval there are four freedoms to uniquely set s_i. Two conditions result by property $s_i(t_i) = y_i$, $s_i(t_{i+1}) = y_{i+1}$. For the remaining two conditions we define $c_i := s_i'(t_i)$.

We now transform the interval $[t_i, t_{i+1}]$ onto $[0, 1)$ using $x := \frac{t-t_i}{\tau_i}$, $\tau_i = t_{i+1} - t_i$ and therefore s_i on $q_i(x) = s_i(t) = s_i(t_i + \tau_i x)$.

Also q_i are cubic polynomials and we get $q_i(0) = y_i$, $q^{(1)} = y_{i+1}$, $q_i'(0) = \tau_i c_i$, $q_i'(1) = \tau_i c_{i+1}$. This is a Hermitian interpolation problem with solution

$$q_i(x) = (1-x')^2 y_i + x^2 y_i t + 1 + x(1-x)[(1-x)(2y_i + \tau_i c_i) + x(2y_{i+1} - \tau_i c_{i+1})]$$

By this interpolation the continuity of s and s is guaranteed.

$$q_i''(0) = 6(y_{i+1} - y_i) - 2\tau_i(2c_i + c_{i+1}) \qquad q_i''(1) = -6(y_{i+1} - y_i) + 2\tau_i(c_i + 2c_{i+1})$$

Because $s''(t) = \frac{g_i''(x)}{\tau_i^2}$, we get the conditions:

$$\frac{2c_0 + c_1}{\tau_0} = \frac{3(y_1 - y_0)}{\tau_0^2}$$
$$\frac{c_{n-1} + 2c_n}{\tau_{n-1}} = \frac{3(y_n - y_{n-1})}{\tau_{n-1}^2}$$
$$\frac{c_{i-1} + 2c_i}{\tau_{i-1}} + \frac{2c_i + c_{i+1}}{\tau_i} = \frac{3(y_i - y_{i-1})}{\tau_{i-1}^2} + \frac{3(y_{i+1} - y_i)}{\tau_i^2} \forall i \neq 0, \dots, n$$

Thus we retrieve a linear equation system:

$$\begin{bmatrix} \frac{2}{\tau_0} & \frac{1}{\tau_0} & \\ \frac{1}{\tau_0} & \frac{2}{\tau_0} + \frac{2}{\tau_1} & \ddots \\ 0 & \frac{1}{\tau_1} & \ddots \\ & 0 & \ddots \end{bmatrix}$$

We want to show that this linear equation system is invertible. It is strictly diagonal dominant.

Theorem (Gershgorin theorem). *For $B \in \mathbb{R}^{n \times n}$,*

$$\operatorname{spec} B \subseteq \bigcup_{k=1}^{n} \left\{ z \in \mathbb{C} \mid |z - b_{kk}| \leq \sum_{l \neq k} |b_{kl}| \right\}$$

For some arbitrary row, this circle lies strictly in the positive complex plane.

$$\implies 0 \notin \operatorname{spec} A \iff A \text{ regular} \implies \exists! [c_0, \dots, c_m]^t : A \cdot c = \text{right-hand side}$$

Thus, a unique spline function s_i created by piecewise polynomials on $[t_i, t_{i+1}]$ exists. □

↓ *This lecture took place on 2018/11/22.*

$$s''(t_0) = 0 \qquad s''(t_n) = 0$$

Theorem (Gershgorin circle theorem). *Let $B \in \mathbb{R}^{n \times n}$.*

$$\sigma(b) = \bigcup_{k=1}^{n} \left\{ z \in \mathbb{C} \mid |z - b_{kk}| \leq \sum_{l \neq k} |b_{kl}| \right\}$$

Applied to A, we can conclude that all eigenvalues of A lie in $\mathbb{C}_{>0}$.

$$\implies 0 \notin \operatorname{spec}(A) \iff A \textit{ is invertible} \implies \exists! : [c_0, \dots, c_n]^T : Ac = \textit{ right-hand side}$$

Thus, the existence of a unique spline function s follows which is realized by piecewise cubic polynomials on $[t_i, t_{i+1}]$.

Remark. *The function s is called* natural spline interpolant *and the marginal conditions, created by minimizing the functional $y[s]$, are called* natural marginal conditions.

Alternatively, instead we can use $s''(t_0) = s''(t_n) = 0$:

Complete spline $s'(t_0) = y_0', s'(t_n) = y_n'$

Periodic spline $s'(t_0) = s'(t_n), s''(t_0) = s''(t_n)$

Spline spaces and B-splines

This is an extended spline concept of arbitrary order k.

Definition. *Let $\Delta := \{t_0, \dots, t_n\}$ is a grid of pairwise different points $a = t_0 < t_1 < \dots < t_n = b$. A spline of degree $k-1$ (of order k) with respect to Δ is a function $s \in C^{k-2}([a, b])$, that corresponds with a polynomial $s_i \in \mathcal{P}_{k-1}$ in every interval $[t_i, t_{i+1}]$ with $i = 0, \dots, n-1$.*

The space of splines of degree $k-1$ with respect to Δ is denoted with $S_{k,\Delta}$.

Remark. *$S_{k,\Delta}$ is a real vector space and it is true that $\mathcal{P}_{k-1} \subset S_{k,\Delta}$. Furthermore the aborted powers of degree $k-1$*

$$(t-t_i)_+^{k-1} = \begin{cases} (t-t_i)^{k-1} & \text{if } t \geq t_i \\ 0 & \text{if } t < t_i \end{cases}$$

are contained in $S_{k,\Delta}$.

Theorem 4.20. *The monomials and aborted powers create a basis*

$$B := \{1, t, \dots, t^{k-1}, (t-t_1)_+^{k-1}, \dots, (t-t_{n-1})_+^{k-1}\}$$

of the spline space $S_{k,\Delta}$. Especially, it holds that

$$\dim(S_{k,\Delta}) = k + n - 1$$

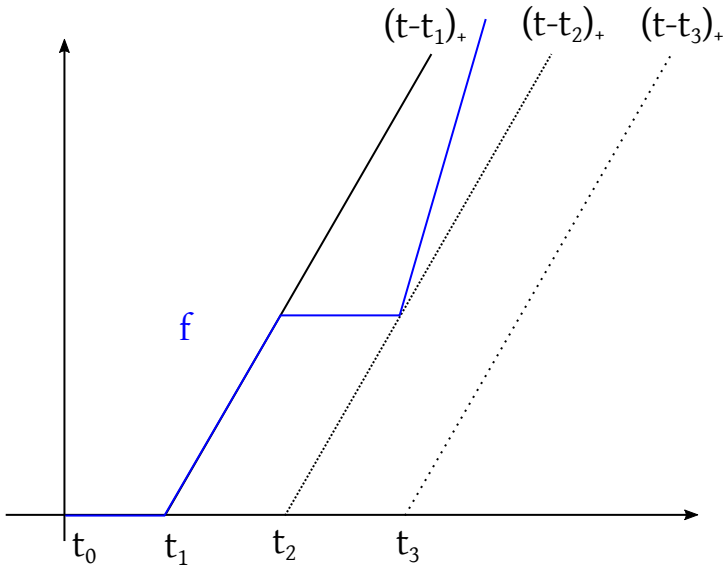

Figure 3: Linear combination of aborted powers of order 2

The basis B is not appropriate to represent a spline $s \in S_{k,\Delta}$, because

- the support of monomials t^i is entire $\mathbb{R}$. $\mathbb{R} \quad \{0\}$.
- for points with $t_i \approx t_{i+1}$, the aborted powers are *almost* linear dependent.

$$s(t) = \sum_{i=0}^{k-1} a_i t^i + \sum_{i=1}^{n-1} c_i (t-t_i)_+^{k-1}$$

Definition 4.21. *Let $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_d$ is an arbitrary sequence of points. Then the B-splines $N_{i,k}(t)$ of order k for $k = 1, \dots, d$ and $i = 1, \dots, d-k$ are recursively defined by*

$$N_{i,1}(t) := \chi_{[\tau_i, \tau_{i+1})}(t) = \begin{cases} 1 & \text{if } \tau_i \leq t < \tau_{i+1} \\ 0 & \text{else} \end{cases}$$

$$N_{i,k}(t) = \frac{t - \tau_i}{\tau_{i+k-1} - \tau_i} N_{i,k-1}(t) + \frac{\tau_{i+k} - t}{\tau_{i+k} - \tau_{i+1}} N_{i+1,k-1}(t)$$

Remark. *The characteristic function χ vanishes for collapsing points. In the recursion, expressions of form $\frac{0}{0} := 0$ are defined. Thus, B-splines are also given for collapsing points.*

Example 4.22 ($k = 2$)**.**

$$\begin{aligned} N_{i,2}(t) &= \frac{t - \tau_i}{\tau_{i+1} - \tau_i} N_{i,1}(t) + \frac{\tau_{i+2} - t}{\tau_{i+2} - \tau_{i+1}} N_{i+1,1}(t) \\ &= \frac{t - \tau_i}{\tau_{i+1} - \tau_i} \chi_{[\tau_i, \tau_{i+1})} + \frac{\tau_{i+2} - t}{\tau_{i+2} - \tau_{i+1}} \chi_{[\tau_{i+1}, \tau_{i+2})} \end{aligned}$$

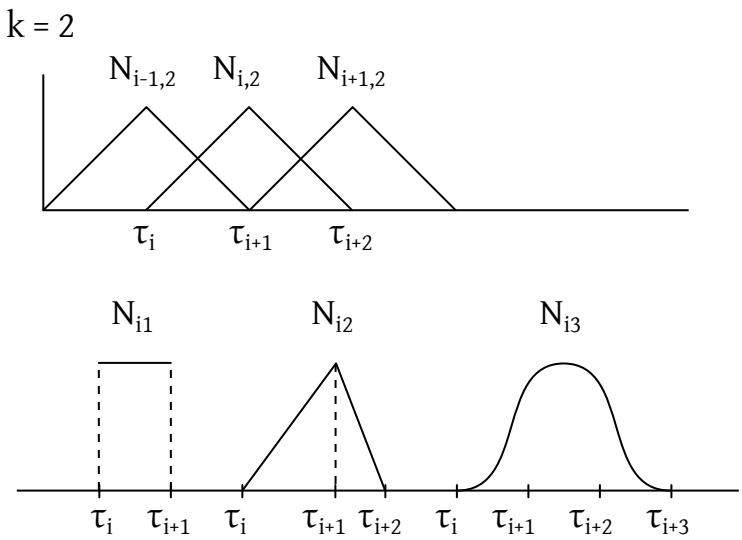

Figure 4: B-splines of order $k = 2$ or $k = 1, 2, 3$

For the B-splines, as defined above, the following properties hold:

1. $\text{supp}(N_{i,k}) \subset [\tau_i, \tau_{i+k}]$ (*local support*)
2. $N_{i,k}(t) \geq 0 \forall t \in \mathbb{R}$ (non-negative)
3. $N_{i,k}$ is a piecewise polynomials of degree $k-1$ with respect to intervals $[\tau_i, \tau_{i+1}]$.

↓ *This lecture took place on 2018/11/27.*

Alternatively we can write B-splines as divided differences for the aborted power $f_t(s) = (s-t)_t^{k-1}$.

Lemma 4.23. *If $\tau_i < \tau_{i+k}$, then B-spline $N_{i,k}$ satisfies $N_{i,k} = (\tau_{i+k} - \tau_i) \cdot [\tau_i \quad \dots \quad \tau_{i+k}] \, (-t)_t^{k-1}$.*

If τ_j is a point with multiplicity M, hence $\tau_j = \tau_{j+M-1}$, then $N_{i,k}$ at point τ_j is a least $(k-1-n)$ times differentiable. For $M < k-1$. For $N_{i,k}$, it is true that

$$N'_{i,k}(t) = (k-1)\left(\frac{N_{i,k-1}(t)}{\tau_{i+k-1} - \tau_i} - \frac{N_{i+1,k}(t)}{\tau_{i+k} - \tau_{i+1}}\right)$$

A proof for Theorem 4.23 is not provided.

Goal: Characterize ρ_k by B-splines...

Therefore introduce *extended points sequence* $T = \{t_j\}_{j=1}^{\alpha+k}$ where the boundary points $a = t_0$ and $b = t_n$ are each counted k times.

$$\triangle : a = t_0 < t_1 < \dots < t_n = b$$

$$T : \tau_1 = \dots = t_k < t_{k+1} < \dots < \tau_{d+1} = \dots = \tau_{d+k} = b$$

with $d = n - 1 + k = \dim(S_{k,0})$.

It can be shown that the B-splines $N_{i,k}$, for $i = 1, \dots, k$, corresponding to extended points T create a basis of $S_{k,\delta}$.

Example 4.24.

$$t_0 < t_1 < t_2 < t_3$$

$$\tau_1 = \tau_2 = t_0 \qquad \tau_3 = t_1 \qquad \tau_4 = t_2 \qquad \tau_5 = \tau_6 = t_4$$

$$N_{1,2} = \frac{\tau_3 - t}{\tau_3 - \tau_2} \cdot X_{[\tau_2, \tau_3]}$$

TODO image splines

Thus, every spline $s \in S_{k,\delta}$ has a unique representation as linear representation $s = \sum_{i=1}^{d} \alpha_i N_{i,k}$. α_i is called De-Boor points *of s.*

Special case $k = 2$ (interpolation of f: $\alpha_i = f(t_{i-1})$.

5 Computational quadrature

Goal: Numerical computation (approximation of Riemann integral)

$$I(f) \coloneqq I_a^b(f) = \int_a^b f \, dt$$

with some piecewise continuous function f.

Numerical integration is primarily solved for solving the initial value problem $y'(t) = f(t), y(a) = 0, t \in [a, b] \implies y(b) = I(f)$.

5.1 Quadrature formulas

Assume $b > a$. The definite integral $I_a^b : C[a,b] \to \mathbb{R}$ with $f \mapsto \int_a^b f\,dt$ is a *positive* linear form, thus

1. I is linear, $I(\alpha f + \beta g) = \alpha I(f) + \beta I(g)$
2. I is positive: $f \geq 0 \implies I(f) \geq 0$
3. *additivity:* $I_a^b(f) = I_a^c(f) + I_c^b(f)$

Question: Condition of I? We choose the L^1-norm:

$$|f|_1 := \int_a^b |f(t)|\,dt = I_a^b(|f|)$$

Lemma 5.1. *The absolute and relative condition of quadrature problem (I, f), $I(f) = \int_a^b f$ with respect to L^1-norm $|\cdot|_1$: $\kappa_{abs} = 1, \kappa_{rel} = \frac{I(|f|)}{|I(f)|}$*

Proof. We investigate: $\left\|I(\tilde{f}) - I(f)\right\| \dot{\leq} \kappa_{\text{abs}} \cdot \left|\tilde{f} - f\right|$.

For every deviation $S_f \in C[a,b]$ we get:

$$\begin{aligned}\left\|I_a^b(f + S_f) - I(f)\right\|_1 &= \left\|I(f + S_f - f)\right\|_1 = \left\|I(S_f)\right\|_1 = \left|\int_a^b S_f\right| \\ &\leq \int_a^b |S_f| = \left\|I(|S_f|)\right\| \implies \kappa_{\text{abs}} = 1\end{aligned}$$

Especially for positive deviation, we get equality. We investigate:

$$\frac{\left\|I(\tilde{f}) - I(f)\right\|}{\|I(f)\|} \dot{\leq} \kappa_{\text{rel}} \frac{\left\|\tilde{f} - f\right\|}{\|f\|}$$

We know:

$$\frac{\left\|I(\tilde{f}) - I(f)\right\|}{\|I(f)\|} \dot{\leq} 1 \cdot \frac{\left\|\tilde{f} - f\right\|}{\|I(f)\|} \cdot \frac{\overbrace{|f|}^{=I(|f|)}}{|f|} \implies \kappa_{\text{rel}} = \frac{I(|f|)}{|I(f)|}$$

□

Remark. *For strongly oscillating functions, relative conditions can become very large.*

Goal: Numerical quadrature shall be used to create a positive linear form on $C[a,b]$.

$$\hat{I} : f \mapsto \hat{I}(f)$$

We want to create $\hat{I}$ und want to approximate I in the best possible way $\implies I(f) \approx \hat{I}(f)$.

TODO image Riemann

$\to$ Riemann sums or trapezoidal sums

Example 5.2 (Trapezoidal sums). *Decompose* $[a, b]$ *in* n *subintervals* $[t_{i-1}, t_i]$. $h_i := t_i - t_{i-1}$.

Approximate $\int f$ *by the sum of trapezoidal areas* $T^{(n)} = \sum_{i=1}^{n} T^{(i)}$, $T^{(i)} = \frac{h_i}{2}(f(t_{i-1}) + f(t_i))$ *Thus,* $T^{(n)}$ *is a positive linear form.*

Now we consider the Riemann lower/upper sum: $R_u^{(n)} \leq T^{(n)} \leq R_0^{(n)}$.

For $n \to \infty$ *and* f *continuous,* R_u *and* R_o *converge, hence* T *converges as well.*

Definition 5.3. $\hat{I}$ *is called* quadrature formula *for computation of* $I(f)$ *if:* $\hat{I}(f) = (b-a) \cdot \sum_{i=0}^{n} \lambda_i f(t_i)$ *with points* $t_0, \dots, t_n$ *and weights* $\lambda_0, \dots, \lambda_n : \sum \lambda_i = 1$.

Be aware that condition $\sum \lambda_i = 1$ requires that constant functions are integrated accurately.

Furthermore: $\hat{I}$ is positive iff $\lambda_i \geq 0 \forall i$.

Using $\sum |\lambda_i| \geq 1$ we can specify how much the positivity property is violated.

5.2 Newton-Cotes formula

Idea: Approximate f by some piecewise linear function, that is integrated accurately, thus $\hat{I}(f) = I(\hat{f})$.

For $t_0, \dots, t_n$ use the interpolation polynomial $\hat{f}(t) = P(f \mid t_0, \dots, t_n) = \sum_{i=0}^{n} f(t_i) \cdot L_{i,n}(t)$ with Lagrange polynomials $L_{i,n}(t) = \prod_{\substack{j \neq i \\ j=0}} \frac{t-t_j}{t_i-t_j}$.

This gives the quadrature formulas $\hat{I}(f) = I(P(f \mid t_0 \dots t_n)) = (b-a) \cdot \sum \lambda_{i,n} f(t_i)$ with $\lambda_{i,n} = \frac{1}{(b-a)} \cdot \int_a^b L_{i,n}(t)\, dt$.

Lemma 5.4. *For* $n+1$ *pairwise different points* $t_0, \dots, t_n$ *there is exactly one quadrature formula* $\hat{I}(f) = (b-a) \sum \lambda_i f(t_i)$, *that are exact for all* $P \in \mathcal{P}_n$.

Proof. Let $L_{i,n}$ be Lagrange polynomials for points t_i. Necessarily,

$$\hat{I}(L_{i,n}) = I(L_{i,n}) = (b-a) \cdot \sum \lambda_j \underbrace{L_{i,n}(t_j)}_{=\delta_{i,j}} = (b-a) \cdot \lambda_i \cdot L_{i,n}(t_j)$$

$$\implies \lambda_i = \frac{I(L_{i,n})}{b-a}$$

□

For equivalent points $h_i = h = \frac{b-a}{n}$ and $t_i = a + i \cdot n$ we retrieve the *Newton-Cotes formulas.*

The corresponding Newton-Cotes weights are given by

$$\lambda_{i,n} = \frac{1}{b-a} \int_a^b \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{t-t_j}{t_i-t_j} = \frac{1}{n} \int_0^n \prod \frac{s-j}{i-j}\, ds$$

Substitute $s = \frac{t-a}{h}$. The weights are independent of interval boundaries. They only must be computed once.

n	$\lambda_{0,n} \dots \lambda_{n,n}$	Error	$\tau \in [a,b]$
1	$\frac{1}{2}, \frac{1}{2}$	$\frac{h^3}{12} f''(\tau)$	trapezoidal rule
2	$\frac{1}{6}, \frac{4}{6}, \frac{1}{6}$	$\frac{h^5}{90} f^{(4)}(\tau)$	Simpson rule, (dt. "Keplersche Faßregel")
3	$\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8}$	$\frac{3 \cdot h^5}{80} \cdot f^{(4)}(\tau)$	Newton's $\frac{3}{8}$ rule
4	$\frac{7}{90}, \frac{32}{90}, \frac{12}{90}, \frac{32}{90}, \frac{7}{90}$	$\frac{8 \cdot h^7}{945} \cdot f^{(6)}(\tau)$	Milne rule

Table 3: Newton-Cotes weights $\lambda_{i,n}$ for $n = 1, \dots, 4$

For orders 1 to 7, the weights are positive. For $n = 8$ and higher, negative weights can occur. This is mostly irrelevant in practice.

Lemma 5.5. *For $g, h \in C[a,b]$, $g \geq 0$ or $g \leq 0$, $\forall t \in [a,b]$ we have:*

$$\int g \cdot h = h(\tau) \cdot \int g \, dt$$

for some $\tau \in [a,b]$.

Proof. Without loss of generality $g \geq 0$

$$\min_{t \in [a,b]} h(t) \cdot \int_a^b g(s) \, ds \leq \int h \cdot g \leq \max h(t) \int g$$

$$F(t) := \int h \cdot g \, ds - h(t) \int g \, ds$$

$$F(a) \geq 0, F(b) \leq 0$$

$$\text{IVT} \implies \exists t : F(t) = 0 \implies \int h \cdot g = h(t) \cdot \int g$$

□

Lemma. *Let $f \in \mathcal{C}^2[a,b]$. For the approximation error the trapezoidal rule $T = \frac{b-a}{2}(f(a) + f(b))$ with $h = b - a$ we have:*

$$T = \int f = \frac{h^3}{12} f''(\tau)$$

for some $\tau \in [a,b]$.

↓ *This lecture took place on 2018/11/29.*

Lemma 5.6. *The map, given by n-th divided difference of a function $f \in \mathcal{C}^1([a,b])$, $g : \mathbb{R}^{n+1} \to \mathbb{R}$ with $g(t_0, \dots, t_n) = [t_0, \dots, t_n]f$ is continuous in its arguments t_i. Furthermore there exists some $\xi \in [t_0, t_n]$ for all points $t_0 \leq \dots \leq t_n$ such that*

$$[t_0, \dots, t_n]f = \frac{f^{(n)}(\xi)}{n!}$$

Lemma 5.7. *Let $f \in \mathcal{C}^2([a,b])$. The approximation error of the trapezoidal rule $T = \frac{b-a}{2}(f(a) - f(b))$ with step size $h := b - a$ it holds that*

$$T - \int_a^b f\,dt = \frac{h^3}{12} f''(\tau)$$

for some $\tau \in [a,b]$.

Proof. By Theorem 4.10, we know that

$$f(t) = P(t) + [a,b,t]f \cdot \underbrace{(t-a)}_{\geq 0}\underbrace{(t-b)}_{\geq 0}$$

and by Lemma 5.6 furthermore

$$[a,b,t]f = \frac{f^k(\xi)}{2}$$

for some $\chi \in [a,b]$ depending on t. For the square rule (see extended Mean Value Theorem), this means

$$\begin{aligned}\int_a^b f\,dt &= \int_a^b P(t)\,dt + \int_a^b [a,b,t]f \cdot \underbrace{(t-a)}_{\geq 0}\underbrace{(t-b)}_{\leq 0} \\ &= T + \frac{f''(\tau)}{2} \underbrace{\int_a^b (t-a)(t-b)}_{= -\frac{(b-a)^3}{6} \text{ for some } \tau \in [a,b]}\end{aligned}$$

for some $\tau \in [a,b]$. Therefore, in total,

$$T = \int_a^b f\,dt = \frac{h^3}{12} f''(\tau)$$

□

Lemma 5.8 (Simpson's rule, (dt. "Keplersche Fassregel")).

$$S = \frac{b-a}{6}(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b))$$

is exact for polynomials $Q \in \mathcal{P}_3$. For $f \in \mathcal{C}^4([a,b])$, the approximation error can be expressed with step size $\frac{b-a}{2}$ as

$$S = \int_a^b f\,dt = \frac{f^{(4)}(\tau)}{90} h^5$$

for some $\tau \in [a,b]$.

Proof. Let $Q \in \mathcal{P}_3$. By Theorem 4.10, it follows by $P(Q \mid a, \frac{a+b}{2}, b)$

$$Q(t) = P(t) + \gamma \underbrace{\gamma(t-a)\left(t - \frac{a+b}{2}\right)(t-b)}_{w_3(t)}$$

with some constant $\gamma = \frac{Q'''(\xi(t))}{6} \in \mathbb{R}$. For the integral, we have:

$$\int_a^b Q(t)\,dt = \int_a^b P(t)\,dt + \gamma \underbrace{\int_a^b w_3(t)\,dt}_{=0}$$

provides the first statement.

For $f \in \mathcal{C}^4([a,b])$, we create the Hermite interpolant $Q = P_4(f \mid a, \frac{a+b}{2}, \frac{a+b}{2}, b)$. By the remainder term formula, we get:

$$f(t) = Q(t) + [a, \frac{a+b}{2}, \frac{a+b}{2}, b, t]f \underbrace{(t-a)\left(t - \frac{a+b}{2}\right)^2 (t-b)}_{=w_4(t)}$$

□

↓ *This lecture took place on 2018/12/04.*

5.3 Gauss–Christoffel Quadrature Formulas

Our goal is to compute "optimal" points t_i and weights λ_i such that $I(P) = \hat{I}(P)$ for $P \in \mathcal{P}_N$ with $N \to$ max. Now we consider the general integral of form $I(f) := \int_a^b w(t)f(t)\,dt$ with some positive weight functions $w(t) > 0$, $t \in (a,b)$. We assume that norms $\|P\| = \sqrt{\langle P, P\rangle_w} = \left(\int_a^b w(t)P(t)\,dt\right)^{\frac{1}{2}} < \infty$ are well-defined for all $P \in \mathcal{P}_k$ and all $k \in \mathbb{N}$. For $a = -\infty$ and $b = \infty$, we require that moments μ_k are bounded:

$$\mu_k := \int_{-\infty}^{\infty} t^k \cdot w(t)\,dt$$

Remark 5.9. *TODO There must be number 5.9 somewhere here.*

For the condition analysis, use weighted L^1-norm

$$\|f\|_{1,W} = \int_a^b w(t)f(t)\,dt =: I(|f|)$$

This is the analogous result for Lemma 5.1.

Typical weight functions are:

- $w(t) = 1$ on $[-1,1]$
- $w(t) = \frac{1}{\sqrt{1-t^2}}$ on $[-1,1]$
- $w(t) = e^{-t}$ on $[0,\infty)$
- $w(t) = e^{-t^2}$ on $(-\infty,\infty)$

Construction: For given n, construct quadratic formulas of form $\hat{I}_n(f) := \sum_{i=0}^{n} \lambda_{i,n} f(\tau_{i,n})$ with $n+1$ points $\tau_{0,n}, \dots, \tau_{n,n}$ and weights $\lambda_{0,n}, \dots, \lambda_{n,n}$ such that $\hat{I}_n(P) = I(P)$ for all $P \in \mathcal{P}_N$. We have $2n+2$ free parameters for a polynomial with $N+1$ coefficients. We guess $N \leq 2n+1$ (but consider the non-linearity of the problem!). It is possible to prove that $N \leq 2n+1$.

Lemma 5.10. *If $\hat{I}_n$ is exact for all polynomials $P \in \mathcal{P}_{2n+1}$, then P_{n+1} is defined by*

$$P_{n+1} := (t - \tau_{0,n}) \dots (t - \tau_{n,n})$$

orthogonal to arbitrary $P_j \in \mathcal{P}_j$, $j < n+1$ with respect to the scalar product weighted by w.

Proof. For $P_j \in \mathcal{P}_j$ with $j < n+1$, we have $P_{n+1} P_j \in \mathcal{P}_{2n+1}$. Thus,

$$\begin{aligned}
\langle P_j, P_{n+1} \rangle &= \int_a^b w(t) P_j(t) P_{n+1}(t)\, dt \\
&= \hat{I}_n(P_j P_{n+1}) = \sum_{i=0}^{n} \lambda_{i,n} P_j(\tau_{i,n}) \underbrace{P_{n+1}(\tau_{i,n})}_{=0} = 0
\end{aligned}$$

□

Theorem 5.11. *For every weighted scalar product $\langle f, g \rangle_w := \int_a^b w(t) f(t) g(t)\, dt$ there exists a uniquely determined orthogonal polynomial $P_k \in \mathcal{P}_k$ with leading coefficient 1. They satisfy the so-called* three-term recursion

$$P_k(t) = (t + a_k) P_{k-1}(t) + b_k P_{k-2}(t) \qquad \textit{for } k = 1, 2, \dots$$

with the initial values $P_{-1} = 0$, $P_0 = 1$ and the coefficients

$$a_k = -\frac{\langle t P_{k-1}, P_{k-1} \rangle_w}{\langle P_{k-1}, P_{k-1} \rangle_w} \qquad b_k = -\frac{\langle P_{k-1}, P_{k-1} \rangle_w}{\langle P_{k-2}, P_{k-2} \rangle_w}$$

Proof. For $k = 0$, there exists a polynomial $p_0 \equiv 1 \in \mathcal{P}_0$ with leading coefficient 1. Now let $P_0, \dots, P_{k-1}$ be pairwise orthogonal polynomials $P_j \in \mathcal{P}_j$ of degree j with leading coefficient 1. Let $P_k \in \mathcal{P}_k$ be a normed polynomial of degree k. The $P_k - tP_{k-1} \in \mathcal{P}_{k-1}$ and because $P_0, \dots, P_{k-1}$ define an orthonormal basis of $\mathcal{P}_{k-1}$, we have

$$P_k - tP_{k-1} = \sum_{j=0}^{k-1} c_j P_j \qquad c_j = \frac{\langle P_k - tP_{k-1}, P_j \rangle_w}{\langle P_j, P_j \rangle_W}$$

By the requirement $\langle P_k, P_j \rangle_w = 0 \forall j = 0, \dots, k-1$, we get

$$c_j = -\frac{\langle tP_{k-1}, P_j \rangle_w}{\langle P_j, P_j \rangle_w} = -\frac{\langle P_{k-1}, tP_j \rangle}{\langle P_j, P_j \rangle}$$

This implies $c_0 = \dots = c_{k-3} = 0$ (orthogonality of P_{k-1} to P_j, $j < k-1$). Furthermore,

$$\begin{aligned}
c_{k-1} &= -\frac{\langle tP_{k-1}, P_{k-1} \rangle_v}{\langle P_{k-1}, P_{k-1} \rangle_w} \\
c_{k-2} &= -\frac{\langle P_{k-1}, tP_{k-2} \rangle_w}{\langle P_{k-2}, P_{k-2} \rangle_w} = -\frac{\langle P_{k-1}, P_{k-1} \rangle_w}{\langle P_{k-2}, P_{k-2} \rangle_w}
\end{aligned}$$

(because $tP_{k-2} \in \mathcal{P}_{k-1}$ and $P_0, \dots, P_{k-1}$ orthonormal basis). P_k follows:

$$P_k = (t + C_{k-1})P_{k-1} + c_{k-2}P_{k-2} = (t + a_k)P_{k-1} + b_k P_{k-2}$$

□

By the two previous results, we conclude that the points must be the roots of the uniquely determined orthogonal polynomial P_{n+1} (for appropriate weight w). To integrate polynomials of degree n exactly, we can use the fact that the weights satisfy

$$\lambda_{i,n} = \frac{1}{b-a} \int_a^b L_{i,n}(t)\, dt$$

with Lagrange polynomials $L_{i,n}$.

Lemma 5.12. *Let $\tau_{0,n}, \dots, \tau_{n,n}$ be the roots of the $(n+1)$-th orthogonal polynomial P_{n+1}. Then every quadrature formula $\hat{I}_n(f) = \sum_{i=0}^n \lambda_i f(\tau_{i,n})$ satisfies:*

$$\hat{I}_n \text{ is exact on } \mathcal{P}_n \iff \hat{I}_n \text{ exact on } \mathcal{P}_{2n+1}$$

Proof. Direction $\Longleftarrow$ is immediate. Direction $\Longrightarrow$:

Let $\hat{I}_n$ be exact on $\mathcal{P}_n$ and $P \in \mathcal{P}_{2n+1}$. Then polynomials $Q, R \in \mathcal{P}_n$ exist such that $P = QP_{n+1} + R$. Thus, it follows that

$$\int_a^b w\, P = \int_a^b wQP_{n+1} + \int_a^b Rw = 0 + \hat{I}_n(R)$$

Additionally, it holds that

$$I(P) = \int_a^b wP = \hat{I}_n(R)$$

$$= \sum_{i=0}^n \lambda_{i,n} R(\tau_{i,n}) = \sum_{i=0}^n \lambda_{i,n} \left(R(\tau_{i,n}) + (P_{n+1} \cdot Q)(\tau_{i,n})\right) = \hat{I}(P)$$

□

Theorem 5.13. *There exists uniquely determined points $\tau_{0,n}, \dots, \tau_{n,n}$ and weights $\lambda_{0,n}, \dots, \lambda_{n,n}$ such that for the quadrature formula*

$$\hat{I}_n(f) = \sum_{i=0}^n \lambda_{i,n} \cdot f(\tau_{i,n})$$

it holds that

$$\hat{I}_n(P) = \int_a^b wP \forall P \in \mathcal{P}_{2n+1}$$

The points $\tau_{i,n}$ are the roots of the $(n+1)$-th orthogonal polynomial P_{n+1} (with leading coefficient 1) with respect to weight function w and the weights are

$$\lambda_{i,n} := \frac{1}{b-a} \int_a^b L_{i,n}(t)\, dt$$

with Lagrange polynomials $L_{i,n}(\tau_{j,n}) = \delta_{i,j}$. Furthermore it holds that $\lambda_{i,n} > 0$, hence $\hat{I}_n$ is the positive linear form and

$$\lambda_{i,n} = \frac{1}{P'_{n+1}(\tau_{i,n})P_n(\tau_{i,n})} \langle P_n, P_n \rangle_w$$

Proof. Show: positivity / representation of $\lambda_{i,n}$

Choose $Q \in \mathcal{P}_{2n+1}$ such that $Q(\tau_{i,n}) = 0$ for $i \neq k$ and $Q(\tau_{k,n}) \neq 0$ for some point $\tau_{k,n}$. Then

$$\int_a^b wQ = \lambda_{k,n} Q(\tau_{k,n})$$

$$\implies \lambda_{k,n} = \frac{1}{Q(\tau_{k,n})} \int_a^b wQ$$

Especially, this holds for

$$Q(t) := \left(\frac{P_{n+1}(t)}{(t - \tau_{k,n})} \right)^2 = \left(\prod_{\substack{i=0 \\ l \neq k}}^{n} (t - \tau_{i,n}) \right)^2$$

It holds that $Q \in \mathcal{P}_{2n}$ and $Q(\tau_{k,n}) = P'_{n+1}(\tau_{k,n})^2$. Thus we get

$$\lambda_{k,n} = \frac{1}{Q(\tau_{k,n})} \int_a^b wQ = \int_a^b w \left(\frac{P_{n+1}(t)}{P'_{n+1}(\tau_{k,n})(t - \tau_{k,n})} \right)^2 dt > 0$$

and therefore the positivity of the weights. For the representation of the weights, we let

$$Q(t) := \frac{P_{n+1}(t)}{t - \tau_{k,n}} P_n(t)$$

Furthermore $Q \in \mathcal{P}_{2n}$ and it follows that

$$\lambda_{k,n} = \frac{1}{P'_{n+1}(\tau_{k,n})P_n(\tau_{k,n})} \int_a^b w(t) \frac{P_{n+1}(t)}{t - \tau_{k,n}} P_n(t)\, dt$$

The leading coefficient of $\frac{P_{n+1}(t)}{t - \tau_{k,n}}$ is 1, such that

$$\frac{P_{n+1}(t)}{t - \tau_{k,n}} = P_n(t) + Q_{n-1}(t) \text{ for some } Q_{n-1} \in \mathcal{P}_{n-1}$$

Because Q_{n-1} is orthogonal to P_n,

$$\lambda_{k,n} = \frac{1}{P'_{n+1}(\tau_{k,n})P_n(\tau_{k,n})} \langle P_n, P_n \rangle_w$$

□

Remark. *The quadrature formulas $\hat{I}_n$ resulting from the theorem are called Gauss-Christoffel formulas of the weight function w.*

Error estimate

Theorem 5.14. *The approximation error of the Gauss-Christoffel quadrature can be expressed for every function $f \in \mathcal{C}^{2n+2}([a,b])$ as*

$$\int_a^b wf\,dt - \hat{I}_n(f) = \frac{f^{(2n+2)}(\tau)}{(2n+2)!}\left\langle P_{n+1}, P_{n+1}\right\rangle_w \text{ for some } \tau \in [a,b]$$

Proof. Let $P \in \mathcal{P}_{2n+1}$ be the Hermitian interpolant of f at $2n+2$ points $\tau_{0,n}, \tau_{0,n}, \dots, \tau_{n,n}\tau_{n,n}$. Then

$$f(t) = P(t) + [\tau_{0,n}, \tau_{0,n}, \dots, \tau_{n,n}, \tau_{n,n}, t]f\underbrace{(t-\tau_{0,n})^2 \dots (t-\tau_{n,n})^2}_{(P_{n+1}(t))^2 \geq 0}$$

Because $\hat{I}_n$ integrates the function P exactly, we get

$$\begin{aligned}\int_a^b wf\,dt &= \int_a^b wP\,dt + \frac{f^{(2n+2)}(\tau)}{(2n+2)!}\int_a^b wP_{n+1}^2\,dt \\ &= \sum_{i=0}^n \lambda_{i,n}P(\tau_{i,n}) + \frac{f^{(2n+2)}(\tau)}{(2n+2)!}\left\langle P_{n+1}, P_{n+1}\right\rangle_w\end{aligned}$$

by Hermitian interpolant

$$\begin{aligned}&= \sum_{i=0}^n \lambda_{i,n}f(\tau_{i,n}) + \frac{f^{(2n+2)}(\tau)}{(2n+2)!}\left\langle P_{n+1}, P_{n+1}\right\rangle_w \\ &= \hat{I}_n(f) + \frac{f^{(2n+2)}(\tau)}{(2n+2)!}\left\langle P_{n+1}, P_{n+1}\right\rangle_w\end{aligned}$$

□

Example 5.15 (Gauss-Chebyshev quadrature). *For $[-1,1]$, $w(t) = \frac{1}{\sqrt{1-t^2}}$ and the Chebyshev polynomials T_k it holds that*

$$\int_{-1}^1 \frac{T_k(t)T_j(t)}{\sqrt{1-t^2}}\,dt = \begin{cases}\pi & if k = j = 0 \\ \pi/2 & if k = j > 0 \\ 0 & if k \neq j\end{cases}$$

Thus $P_n(t) = 2^{1-n}T_n(t)$ are the orthogonal polynomials with leading coefficient 1. The roots of P_{n+1} are the Chebyshev points

$$\tau_{i,n} = \cos\left(\frac{2i+1}{2n+2}\pi\right) \quad i = 0, \dots, n$$

For $n > 0$, we get the weights

$$\lambda_{i,n} = \frac{1}{2^{-n}T'_{k+1}(\tau_{i,n})2^{1-n}T_n(\tau_{i,n})}\int_{-1}^1 \frac{2^{2-2n}T_n^2}{\sqrt{1-t^2}}\,dt = \frac{2\pi/2}{T'_{n+1}(\tau_{i,n})T_n(\tau_{i,n})} = \frac{\pi}{n+1}$$

Implies the Gauss-Chebyshev quadrature of form

$$\hat{I}_n(f) = \frac{\pi}{n+1} \sum_{i=0}^{n} f(\tau_{i,n}) \text{ with } \tau_{i,n} = \cos\left(\frac{2i+1}{2n+2}\pi\right)$$

with approximation error

$$\int_{-1}^{1} \frac{f(t)}{\sqrt{1-t^2}}\, dt - \hat{I}_n(f) = \frac{\pi}{2^{2n+1}(2n+2)!} f^{(2n+2)}(\tau) \text{ for } \tau \in [-1,1]$$

↓ *This lecture took place on 2018/12/06.*

Common orthogonal systems are:

1. Chebyshev polynomials T_n for $w(t) = \frac{1}{\sqrt{1-t^2}}$ on $[-1,1]$
2. Laguerre polynomials L_n for $w(t) = w^{-t}$ on $[0,\infty)$
3. Hermitian polynomials H_n for $w(t) = e^{-t^2}$ on $(-\infty,\infty)$
4. Legendre polynomials P_n for $w(t) = 1$ on $[-1,1]$

Computation of roots and weights

Disadvantages of Gauss-Christoffel quadrature:

1. coefficients/points depend on the integration interval
2. for every n we retrieve different coefficients/points

The first disadvantage can be fixed by the affine linear transformation

$$t = \frac{b-a}{2}x + \frac{a+b}{2}$$

$$\implies I(f) = \int_a^b f(t)\, dt = \frac{b-a}{2} \int_{-1}^{1} \underbrace{f\left(\frac{b-a}{2}x + \frac{a+b}{2}\right)}_{\tilde{f}(x)} dx$$

$$\implies \hat{I}(f) = \frac{b-a}{2} \sum_{i=0}^{n} \lambda_{i,n} f\left(\frac{b-a}{2}\tau_{i,n} + \frac{a+b}{2}\right)$$

If we want to approximate an integral $\int_a^b f(t)\, dt$, we can write it as

$$\int_a^b \underbrace{\frac{f(t)}{w(t)}}_{\tilde{f}(t)} \cdot w(t)\, dt$$

In practice, points $\tau_{i,n}$ and weights $\lambda_{i,n}$ will be determined for the orthogonal polynomials based on the three-term recursion.

By Theorem 5.11, we know that

$$P_k(t) = (t - \beta_k)P_{k-1}(t) - \gamma_k^2 P_{k-2}(t)$$

$$\text{with} \qquad \beta_k = \frac{\langle tP_{k-1}, P_{k-1}\rangle_w}{\langle P_{k-1}, P_{k-1}\rangle_w} \qquad \gamma_k^2 = \frac{\langle P_{k-1}, P_{k-1}\rangle_w}{\langle P_{k-2}, P_{k-2}\rangle_w}$$

If the orthogonal polynomials are given by this recursion, we can define the linear equation system $T_p = tp + r$ with

$$T := \begin{bmatrix} \beta_1 & 1 & & & \\ \gamma_2^2 & \beta_2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_n^2 & \beta_n & 1 \\ & & & \gamma_{n+1}^2 & \beta_{n+1} \end{bmatrix} \qquad p := \begin{bmatrix} P_0(t) \\ \vdots \\ P_n(t) \end{bmatrix} \qquad r := \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -P_{n+1}(t) \end{bmatrix}$$

Then it holds that $P_{n+1}(t) = 0 \iff T_p = t_p$, so the roots of P_{n+1} are the eigenvalues of T and for every eigenvalue τ, we have a corresponding eigenvector $P(\tau)$.

All roots are real.

Question: Can we write it as symmetric eigenvalue problem?

Trial: Scaling of eigenvalue problem with some diagonal matrix $D = \operatorname{diag}(d_0, \dots, d_n)$ such that $\hat{T}\hat{p} = t\hat{p}$ with $\hat{p} = Dp$ and $\hat{T} = DTD^{-1}$.

For the orthogonal scaling applied to $A \in \mathbb{R}^{(n+1)\times(n+1)}$ it is true that $A \mapsto \hat{A} := DAD^{-1}$. $\hat{a_{ij}} = \frac{di}{dj} a_{ij}$.

For symmetry of $\hat{T}$ we require that

$$\gamma_{i+1}^2 \frac{d_i}{d_{i-1}} = \frac{d_{i-1}}{d_i} \implies d_i^2 = \frac{d_{i-1}^2}{\gamma_{i+1}^2}$$

We can achieve this by

$$d_0 := 1 \qquad d_i = \frac{1}{\gamma_2 \cdots \gamma_{i+1}} \qquad i = 1, \dots, n$$

This way we retrieve the symmetric tridiagonal matrix:

$$T := \begin{bmatrix} \beta_1 & \gamma_2 & & & \\ \gamma_2 & \beta_2 & \gamma_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_n & \beta_n & \gamma_{n+1} \\ & & & \gamma_{n+1} & \beta_{n+1} \end{bmatrix} = \hat{T}^T$$

The integration weights can be determined by the corresponding eigenvalue problem ($\to$ eigenvalues).

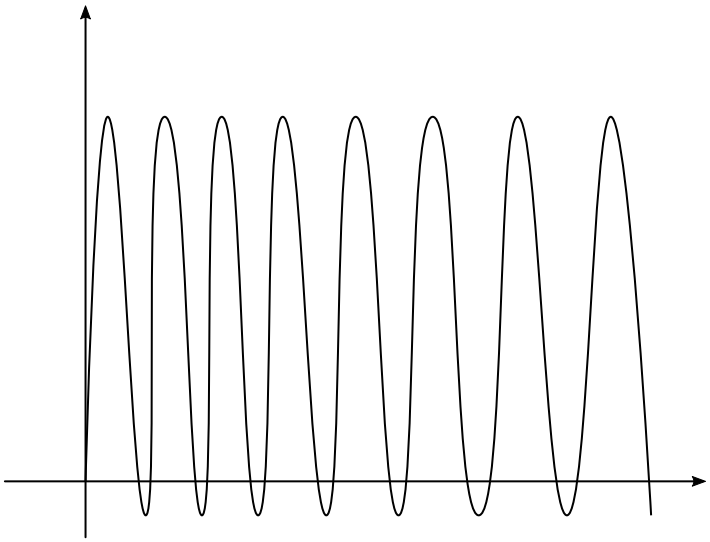

Figure 5: Motivational example for adaptive quadrature

5.4 Adaptive Quadrature

Use a problem-adjusted grid with *adaptive step size* $\delta_j = t_{j+1} \cdot t_j$ with respect to the desired tolerance.

Idea: use an *adaptive quadrature* which is based on the a-posteriori estimate of the integration error

$$\varepsilon_a^b(f) \coloneqq \hat{I}_a^b(f) - I_a^b(f)$$

Processes of adaptive quadrature are successive applications of a cycle consisting of steps.

$$\underbrace{\text{solution}}_{\hat{I}_a^b(f)} \implies \underbrace{\text{estimate}}_{\substack{\text{error estimate}\\ \text{by a quadrature}\\ \text{formula of higher order}}} \implies \text{marking} \implies \underbrace{\text{refinement}}_{\text{bijection}}$$

↓ *This lecture took place on 2018/12/11.*

Solution Computation of approximation value $\hat{I}^{(l)}(f)$, $l \in \mathbb{N}_0$ using a quadratur formula, eg, trapezoidal rule with respect to some given decomposition,

$$\triangle_l \coloneqq \left\{a = t_0^{(l)} < t_1^{(l)} < \cdots < t_{n_l}^{(l)} = b\right\}$$

of the integration domain $[a, b]$ in its subintervals $\triangle_{l,i} \coloneqq [t_i^{(l)}, t_{i+1}^{(l)}]$ of length $\left|\triangle_{l,i}\right| = t_{i+1}^{(l)} - t_i^{(l)}$ with $0 \leq i \leq n_l - 1$. In the following we consider the quadrature formula as *basis procedure* and l as the step of the refinement process.

estimate consists of estimate of the integration error

$$\varepsilon^{(l)}(f) := \hat{I}^{(l)}(f) - I(f)$$

based on a posteriori data (with desirably little computational requirements).

marking marks—based on the error estimate—the intervals $\triangle_{l,i}$, that should be refined for some more precise approximation. For $l \geq 1$ and $i \in \{0, \dots, m_l - 1\}$, we call $\triangle_{l,i}$ some interval of step $s(\triangle_{l,i}) = l$, if $\triangle_{l,i}$ is created by bisection of some interval from $\triangle_{l-1}^M$. Otherwise $\triangle_{l,i}$ corresponds to some interval of decomposition $\triangle_{l-1}$ and therefore $s(\triangle_{l,i}) \geq l-1$. We denote the set of intervals marked for refinement with $\triangle_l^M$.

refinement uses some bisection of $\triangle_{l,i} = \begin{bmatrix} t_i^{(l)} & \dots & t_{i+1}^{(l)} \end{bmatrix}$ in two subintervals

$$\triangle_{l,i} := \begin{bmatrix} t_i^{(l)} & \frac{t_i^{(l)}+t_{i+1}^{(l)}}{2} \end{bmatrix} \triangle_{l,i}^+ := \begin{bmatrix} \frac{t_{i+1}^{(l)}+t_i^{(l)}}{2} & t_{i+1}^{(l)} \end{bmatrix}$$

For the estimate of the integration error $\varepsilon^{(l)}(f)$ look for some simple-to-compute and localizable estimate η_l of form $\eta_l = \sum_{i=0}^{n_l-1} \eta_{\triangle_{l,i}}$ where $\eta_{\triangle_{l,i}}$ with $0 \leq i \leq n_l - 1$ represents estimates of integration errors $\varepsilon_{\triangle_{l,i}}(f)$ with respect to subintervals $\triangle_{l,i}$. We call estimator η_l *reliable*, if the constant $\Gamma > 0$ exists, such that $\left|\varepsilon^{(l)}(f)\right| \leq \Gamma_{\eta_l}$. We call it *efficient* if $\gamma\eta_l \leq \left|\varepsilon^{(l)}(f)\right|$ with some constant $\gamma > 0$.

For some given tolerance we want to abort the adaptive refinement process, if $\eta_l \leq$ tolerance. Reliable estimators also guarantee a uniform magnitude for the integration error. Efficient estimators ensure that integration errors are not overestimated too much. In the perfect case, we have $\gamma_l\Gamma \approx 1$.

Idea: Construct an estimator by using two quadrature formulas of different order, eg. Simpson's rule $\hat{I}(f)$ for basis processes.

$$\text{trapezoidal rule} \equiv \hat{I}(f)$$

Therefore we assume that

$$\left|\tilde{\varepsilon}_{\triangle_{l,i}}^{(l)}(f)\right| \leq q \left|\varepsilon_{\triangle_{l,i}}^{(l)}(f)\right| \qquad 0 \leq q < 1$$

Correspondingly we define

$$\eta_{\triangle_{l,i}} := \left|\hat{I}_{\triangle_{l,i}}^{(l)}(f) - \tilde{I}_{\triangle_{l,i}}^{(l)}(f)\right|$$

as estimator of the local integration error $\varepsilon_{\triangle_{l,i}}^{(l)}(f)$. By the triangle inequality, we get

$$\begin{aligned}
\left|\varepsilon_{\triangle_{l,i}}^{(l)}\right| &= \left|\hat{I}_{\triangle_{l,i}}^{(l)}(f) - I(f)\right| \\
&= \left|\hat{I}_{\triangle_{l,i}}^{(l)}(f) - \tilde{I}_{\triangle_{l,i}}^{(l)}(f)\right| + \left|\tilde{I}_{\triangle_{l,i}}^{(l)}(f) - I(f)\right| \\
&\leq \eta_{\triangle_{l,i}} + \left|\tilde{\varepsilon}_{\triangle_{l,i}}^{(l)}(f)\right| \\
&\leq \eta_{\triangle_{l,i}} + q\left|\varepsilon_{\triangle_{l,i}}^{(l)}(f)\right|
\end{aligned}$$

At the same time we have,

$$\left|\varepsilon^{(l)}_{\triangle_{l,i}}(f)\right| \geq \eta_{\triangle_{l,i}} - \left|\widetilde{\varepsilon}^{(l)}_{\triangle_{l,i}}(f)\right|$$

Therefore $\eta_{\varepsilon_{l,i}}$ is a reliable ($\Gamma = \frac{1}{1-q}$) and efficient ($\gamma = \frac{1}{1+q}$) estimator at some subinterval $\triangle_{l,i}$. For marking in the adaptive cycle we assume that the local estimators $\eta_{\triangle_{l,i}}$ with constants $\kappa > 1$ and $c > 0$ behave like $\eta_{\triangle_{l,i}} \doteq c\left|\triangle_{l,i}\right|^{\kappa}$. We denote $\triangle_{l-1,j_i}$ and $j_i \in \{0, \dots, n_l - 1\}$ the interval at step $\leq l-1$ which contains $\triangle_{l,i}$. If $\triangle_{l-1,j_i} \in \triangle^M_{l-1}$, we have $\left|\triangle_{l-1,j_i}\right| = 2\left|\triangle_{l,i}\right|$ and it follows that

$$\eta_{\triangle_{l-1,j_i}} \doteq 2^{\kappa}\eta_{\triangle_{l,i}}$$

Recognize that

$$2^{-\kappa} \doteq \frac{\eta_{\triangle_{l,i}}}{\eta_{\triangle_{l-1,j_i}}}$$

Because $\left|\triangle_{l,i}\right|^I = \frac{\triangle l,i}{2}$, we get

$$\eta_{\triangle^{\pm}_{l,i}} \doteq C2^{-\kappa}\left|\triangle_{l,i}\right|^{\kappa} \doteq \frac{\eta^2_{\triangle_{l,i}}}{\eta_{\triangle_{l-1,j_i}}} =: \overline{\eta}_{\triangle_{l,i}}$$

To process step marking, we use the maximum $m_{\eta_l} := \max_{s(\triangle_{l,i})=l} \overline{\eta_{\triangle_{l,i}}}$ of local estimators $\eta_{\triangle_{l,i}}$ with $s(\triangle_{l,i}) = l$. For some given $\theta \in (0,1)$ we mark all subintervals $\triangle_{l,i}$ for refinement for which $\eta_{\triangle_{l,i}} \geq \theta m_{\eta_l}$. As termination condition we compare $\widehat{I}^{(l)}_{[a,b]}(f)$ with $\widehat{I}^{(l-1)}_{[a,b]}(f)$.

$$\varepsilon^{(l)}_{\text{rel}} := \frac{\left|\widehat{I}^{(l)}_{[a,b]}(f) - \widehat{I}^{(l-1)}_{[a,b]}(f)\right|}{\left|\widehat{I}^{(l)}_{[a,b]}(f)\right|} \leq \text{ tolerance}$$

Algorithm 5.

1. *Input:* $f, \triangle_0, \theta \in (0,1]$ *and tolerance* > 0
2. *for* $l = 0, 1, \dots$
   - (a) *Determine* $\widehat{I}^{(l)}_{l,i}(f), \widehat{I}^{(l)}_{\triangle_{l,i}}(f), \eta_{\triangle_{l,i}}$ *and* $\widehat{I}^{(l)}_{[a,b]}(f)$
   - (b) *if* $l = 1$
     - i. *Determine* $\varepsilon^{(l)}_{rel}$
     - ii. *if* $\varepsilon^{(l)}_{rel} \leq$ *tolerance*
       - A. *Terminate with solution* $I^{(l)}_{[a,b]}(f)$
   - (c) *for* $i = 0, 1, \dots, m_l - 1$
     - i. *Determine* $\eta_{\widehat{\triangle_{l,i}}}$ *for* $\triangle_{l,i}$
   - (d) *Compute* m_{η_l}
   - (e) *Create* $\triangle_{l+1}$ *by bisection of all* $\triangle_{l,i}$ *for which* $\eta_{\triangle_{l,i}} \geq \theta_{\eta_l}$

5.5 Multidimensional quadrature

Goal: Overview how to generalize quadrature formulas over domains $D \subset \mathbb{R}^n$.
Idea: Reduce to integration of standard domains

Transformation of integrals

Reduce from $\int_a^b$ to $\int_c^d$ by expressing a coordinate transformation by some bijective, continously differentiable map $\Psi : [c, d] \to [a, b]$. Variable transformation $x = \Psi(t)$ gives

$$\int_a^b f(x)\, dx = \int_c^d f(\Psi(t))\Psi'(t)\, dt$$

For the special case of affine maps $\Psi(t) = \alpha t + \beta$ with $\alpha, \beta \in \mathbb{R}$, the parameters are distinctly given by $\Psi(c) = a$ and $\Psi(d) = b$.

Thus, intervals are mapped the intervals for $\alpha \neq 0$. Furthermore it holds that $p(x)$ is a polynomial of degree n in x, then $p(\Psi(t))$ a polynomial of degree n in t.

The accuracy of the quadrature formula is not affected.

In $\mathbb{R}^n$, we use an analogous approach. We integrate over some domain B_1 using some quadrature formula for B_2. Affine maps have the form

$$\Psi : B_2 \to B_1 \qquad \Psi(t) = At + c : A \in \mathbb{R}^{n \times n}, c \in \mathbb{R}^n$$

and are uniquely defined by predefining the transformation of $n+1$ points ($n(n+1)$ equations for $n^2 + n$ parameters in A and c). The transformation formula is then given by

$$\int_{B_1} f(x)\, dx = \int_{B_2} f(\Psi(t)) \cdot |\det(\nabla\Psi(t))|\ dt$$

where $\nabla\Psi(t)$ denotes the Jacobian matrix of Ψ. In the affine case, we have $\nabla\Psi(t) = A$

Integration over standard square

We consider $f : Q \to \mathbb{R}$ and $Q = [0, 1] \times [0, 1] \subset \mathbb{R}^2$

$$\iint_Q f(x, y) d(x, y) = \int_0^1 \int_0^1 f(x, y)\, dx\, dy$$

The integration boundaries for x depend not on y and we can insert separate quadrature formulas for x and y.

$$\begin{aligned}
\int_0^1 \int_0^1 f(x,y)\,dx\,dy &= \int_0^1 F(y)\,dy \\
&\approx \sum_{j=0}^{n} \lambda_j F(y_j) \\
&= \sum_{j=0}^{n} \lambda_j \int_0^1 f(x,y_j)\,dx \\
&\approx \sum_{j=0}^{n} \lambda_j \sum_{k=0}^{n} \lambda_k f(x_k,y_j) \\
&= \sum_{j,k=0}^{n} \lambda_j \lambda_k \cdot f(x_k,y_j)
\end{aligned}$$

with points and weights (y_j^κ, λ_j), and accordingly $(x_\kappa, \lambda_\kappa)$.

If the quadrature formulas are exactly of degree m, then functions of form $x^k y^j$, $0 \leq k, j \leq m$ are integrated accurately. The approximation error can be derived by the error representation of one-dimensional quadrature formulas.

↓ *This lecture took place on 2018/12/13.*

Integration over default triangle

We consider $f : T \to \mathbb{R}, T = \{(x,y) \mid 0 \leq x \leq 1, 0 \leq y \leq 1-x\} \subset \mathbb{R}^2$.

$$\iint_T f(x,y)\,d(x,y) = \int_0^1 \int_0^{1\cdot x} f(x,y)\,dy\,dx$$

Here, integration boundaries are not independent in x and y (and therefore the supporting points). Thus consider the quadrature formulas of form

$$\sum_{j=0}^{n} \lambda_j f(x_j,y_j) \approx \iint_T f(x,y) d(x,y)$$

We require that polynomials of form $x^k y^j$, $0 \leq k+j \leq m$, are integrated accurately. for $m \geq 2$ e.g. the polynomials $1, x, y, x^2, xy, y^2$ (those create an orthogonalbasis wrt. scalar product $\langle f, g\rangle = \iint_T f(x,y)g(x,y)\,d(x,y)$).

So, in $\mathbb{R}^n$ there exists for all m some quadrature formula of accuracy m, that requires only $\binom{m+n}{n}$ supporting points.

These are not unique.

(x_j, x_j)	λ_j	accurately
$(\frac{1}{3}, \frac{1}{3})$	$\frac{1}{2}$	1
$(0,0), (1,0), (0,1)$	$\frac{1}{6}$	1
$(\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, 0), (0, \frac{1}{2})$	$\frac{1}{6}$	2
$(\frac{1}{6}, \frac{1}{6}), (\frac{2}{3}, \frac{1}{6}), (\frac{1}{6}, \frac{2}{3})$	$\frac{1}{6}$	2

Table 4: The first line corresponds to the Gauss quadrature on T

5.6 Higher dimensional integrals

Question: How do we compute higher dimensional integrals $\int_Q f(x)\,dx$ for, for example, $Q = \prod_{i=1}^d [a_i, b_i]$.

$$\text{Fubini's Theorem} \implies \int_Q f(x)\,dx = \int_{a_d}^{b_d} \dots \int_{a_1}^{b_1} f(x_1, \dots, x_d)\,dx_1 \dots dx_d$$

Use one-dimensional quadrature formulas: Consider e.g. $Q = [0,1]^d$ and f arbitrary. For $d = 1$ we get by trapeziodal sum the approximation

$$\int_0^1 f(x)\,dx \approx \sum_{i=0}^n \lambda_i f\left(\frac{i}{n}\right)$$

with $\lambda_0 = \lambda_n = \frac{1}{2n}$ and $\lambda_i = \frac{1}{\underbrace{m}_{k}}$ for $i \neq 0, n$.

In the multidimensional case we analogously get

$$\int_a f(x)\,dx = \int_0^1 \dots \int_0^1 f(x_1, \dots, x_d)\,dx_1 \dots dx_d \approx \sum_{j_1=0}^n \dots \sum_{j_d=0}^n \lambda_{j_1} \dots \lambda_{j_d} f\left(\frac{j_1}{m}, \dots, \frac{j_d}{m}\right)$$

The number of evaluations is $N = (n+1)^d$. For $d = 1$ and $f \in \mathcal{C}^2([0,1])$ we know that the integration error behaves like $\mathcal{O}(\frac{1}{n^2})$. For $d \geq 2$, this value does not improve (consider the special case $f(x_1, \dots, x_d) = g(x_1)$). The approximation error of the quadrature formula above behaves like $\mathcal{O}(\frac{1}{n^2}) = \mathcal{N}^{-\frac{2}{d}}$.

For large d the error margin decreases. To halve the approximation error, we need to apply $2^{\frac{d}{2}}$ more function evaluations (*curse of dimensionality*).

Remark. *The following paragraph is not relevant for exams.*

Monte-Carlo integration

In the following, we denote the Lebesgue measure on $\mathbb{R}^d$ with γ. For $Q := \prod_{i=1}^d [a_i, b_i]$ we have $\gamma(Q) = \prod_{i=1}^d (b_i - a_i)$. For some integration domain B with $0 < \gamma(B) < \infty$, we consider the probability space $(B, d(B), \mu)$ with Lebesgue-measurable subsets of B and the probability measure $\mu = \frac{\gamma}{\gamma(B)}$. For some function $f \in \mathcal{C}(B)$ we have

$$\int_B f\,d\gamma(x) = \gamma(B) \cdot \int_B f(x)\,d\mu(x) = \gamma(B) \cdot \mathcal{E}(f(X))$$

where X is a uniform random variable distributed among B.

Index