NumCSE - Complete (Ch. 6-7)

Additional files	
Class	NumCSE
Date	@Nov 17, 2020
PPT	
Topic	Lecture Document Ch. 6-7
Type	Book

0. Table of Contents

6. Approximation of Functions in 1D

6.1 Introduction

Approximation of functions: Generic view

- Given: a *function* $f : D \subset \mathbb{R}^n \to \mathbb{R}^d$, $n, d \in \mathbb{N}$, often in procedural form, e.g. for $n = d = 1$, as `double f(double)`
- Goal: find a simple function $\tilde{f} : D \to \mathbb{R}^d$ such that the approximation error $f - \tilde{f}$ is small.

We define an abstract concept for the sake of clarity: When in this chapter we talk about an ***"approximation scheme"*** we refer to a mapping $A : X \to V$, where X and V are spaces of functions $I \to \mathbb{K}$, $I \subset \mathbb{R}$ an interval.

Examples are:

- $X = C^k(I)$, the spaces of functions $I \to \mathbb{K}$ that are k times continuously differentiable.
- $V = \mathcal{P}_m(I)$, the space of polynomials of degree $\leq m$.
- $V = \mathcal{S}_{d,\mathcal{M}}$, the space of splines of degree d on the knot set $\mathcal{M} \subset I$.

- $V = \mathcal{P}_{2n}^T$, the space of trigonometric polynomials of degree $2n$.

Every interpolation scheme spawns a corresponding approximation scheme:

Interpolation scheme + sampling → approximation scheme

$$f : I \subset \mathbb{R} \to \mathbb{K} \to_{sampling} (t_i, y_i := f(t_i))_{i=0}^{m} \to_{interpolation} \tilde{f} := I_{\mathcal{T}} y \ \ (\tilde{f}(t_i) = y_i).$$

6.2 Approximation by Global Polynomials

The local approximation of sufficiently smooth functions by polynomials is a key idea in calculus, which manifest itself in the importance of approximation by ***Taylor polynomials***: For $f \in C^k(I)$, $k \in \mathbb{N}$, $I \subset \mathbb{R}$ an interval, we approximate

$$f(t) \simeq f(t_0) + f'(t_0)(t - t_0) + \frac{f^{(2)}(t_0)}{2}(t - t_0)^2 + \cdots + \frac{f^{(k)}(t_0)}{k!}(t - t_0)^k, \text{ for some } t_0 \in I.$$

The taylor polynomial T_k of degree k approximates f in a neighbourhood $J \subset I$ of t_0. This can be quantified by the ***remainder formulas***

$$f(t) - T_k(t) = \int_{t_0}^{t} f^{(k+1)}(\tau)\frac{(t-\tau)^k}{k!}\mathrm{d}\tau = f^{(k+1)}(\zeta)\frac{(t-t_0)^{k+1}}{(k+1)!}, \, \zeta = \zeta(t, t_0) \in]\min(t, t_0), \, \max(t, t_0)[,$$

which show that for $f \in C^{k+1}(I)$ the Taylor polynomial T_k is pointwise close to $f \in C^{k+1}(I)$, if the interval I is small and f^{k+1} is bouned pointwise.

6.2.1 Polynomial Approximation: Theory

Sloppily speaking, the Taylor polynomials provide *uniform* approximation of a smooth function f in small intervals, provided that its derivatives do not blow up "too fast".

The question is, whether polynomials still offer uniform approximation on arbitrary bounded closed intervals and for functions that are merely continuous, but not any smoother. The answer is *Yes* and this profound result is known as the ***Weierstrass Approximation Theorem.***

Theorem: Uniform Approximation by Polynomials

For $f \in C^0([0, 1])$, define the n-th ***Bernstein approximation*** as

$$p_n(t) = \sum_{j=0}^{n} f\left(\frac{j}{n}\right)\binom{n}{j} t^j (1-t)^{n-j}, \, p_n \in \mathcal{P}_n.$$

It satisfies $||f - p_n||_\infty \to 0$ for $n \to \infty$. If $f \in C^m([0, 1])$, then even $||f^{(k)} - p_n^{(k)}||_\infty \to 0$ for $n \to \infty$ and all $0 \leq k \leq m$.

In the equation above, the function f is approximated by a linear combination of ***Bernstein polynomials*** of degree n

$$B_j^n(t) := \binom{n}{j} t^j (1-t)^{n-j}, \, B_j^n \in \mathcal{P}_n.$$

Since f is *uniformly continuous* on $[0, 1]$, given $\epsilon > 0$, we can choose $\delta > 0$ independently of t such that $|f(s) - f(t)| < \epsilon$, if $|s - t| < \delta$. Then, if we choose $n > (\epsilon\delta^2)^{-1}$, we can bound

$$|f(t) - p_n(t)| \leq (||f||_\infty + 1)\epsilon \ \ \forall t \in [0, 1].$$

This means that p_n is arbitrarily close to f for sufficiently large n.

Definition: Size of Best Approximation Error

Let $|| \cdot ||$ be a semi-norm on a space X of functions $I \to \mathbb{K}$, $I \subset \mathbb{R}$ an interval. The size of the ***best approximation error*** of $f \in X$ in the space $\mathcal{P}_k$ of polynomials of degree $\leq k$ with respect to $|| \cdot ||$ is

$$\mathrm{dist}_{||\cdot||}(f, \mathcal{P}_k) := \inf_{p \in \mathcal{P}_k} ||f - p||.$$

Theorem: If $f \in C^r([-1,\,1])$ (r times continuously differnetiable), $r \in \mathbb{N}$, then for any polynomials degree $n \geq r$,

$$\inf_{p\in\mathcal{P}_n} ||f-p||_{L^\infty([-1,\,1])} \leq \left(1+\frac{\pi^2}{2}\right)^r \frac{(n-r)!}{n!}\left|\left|f^{(r)}\right|\right|_{L^\infty([-1,\,1])},$$

where

$$\left|\left|f^{(r)}\right|\right|_{L^\infty([-1,\,1])} := \max_{x\in[-1,\,1]}|f^{(r)}(x)|.$$

Assume that an interval $[a,\,b] \subset \mathbb{R},\ a < b$, and a polynomial approximation scheme $\hat{A}: C^0([-1,\,1]) \to \mathcal{P}_n$ are given. Based on the *affine linear mapping*

$$\phi: [-1,\,1] \to [a,\,b],\ \phi(\hat{t}) := a + \frac{1}{2}(\hat{t}+1)(b-a),\ -1 \leq \hat{t} \leq 1,$$

we can introduce the ***affine pullback*** of functions:

$$\phi^*: C^0([a,\,b]) \to C^0([-1,\,1]),\ \phi^*(f)(\hat{t}) := (f\circ\phi)(\hat{t}) := f(\phi(\hat{t})),\ -1 \leq \hat{t} \leq 1.$$

Lemma: If $\phi^*: C^0([a,\,b]) \to C^0([-1,\,1])$ is an affine pullback, then $\phi^*: \mathcal{P}_n \to \mathcal{P}_n$ is a *bijective* linear mapping for any $n \in \mathbb{N}_0$.

The lemma tells us that the spaces of polynomials of some maximal degree are *invariant under affine pullback*. Thus, we can define a ***polynomial approximation scheme*** A on $C^0([a,\,b])$ by

$$A: C^0([a,\,b]) \to \mathcal{P}_n,\ A := (\phi^*)^{-1} \circ \hat{A} \circ \phi^*,$$

whenever $\hat{A}$ is a polynomial approximation scheme on $[-1,\,1]$.

Lemma: For every $f \in C^0([a,\,b])$ we have

$$||f||_{L^\infty([a,\,b])} = ||\phi^* f||_{L^\infty([-1,\,1])},\ ||f||_{L^2([a,\,b])} = \sqrt{|b-a|}\cdot||\phi^*||_{L^2([-1,\,1])}.$$

6.2.2 Error Estimates for Polynomial Interpolation

In Section 5.2.2 we introduced the Lagrangian polynomial intterpolation operator $I_\mathcal{T}: \mathbb{K}^{n+1} \to \mathcal{P}_n$ belonging to a node set $\mathcal{T} = \{t_j\}_{j=0}^n$. It introduces an approximation scheme on $C^0(I),\ I \subset \mathbb{R}$ an interval, if $\mathcal{T} \subset I$.

Definition: Lagrangian Approximation Scheme

Given an interval $I \subset \mathbb{R},\ n \in \mathbb{N}$, a node set $\mathcal{T} = \{t_0, \ldots, t_n\} \subset I$, the ***Lagrangian interpolation polynomial approximation scheme*** $L_\mathcal{T}: C^0(I) \to \mathcal{P}_n$ is defined by

$$L_\mathcal{T}(f) := I_\mathcal{T}(y) \in \mathcal{P}_n \text{ with } y := [f(t_0), \ldots, f(t_n)]^T \in \mathbb{R}^{n+1},\ I_\mathcal{T}(y)(t_j) = (y)_j,\ j = 0, \ldots, n.$$

Our goal in this section will be to estimate the norm of the ***interpolation error*** $||f - I_\mathcal{T} f||$ (for relevant norm on $C(I)$).

6.2.2.1 Convergence of Interpolation Errors

Definition: Types of asymptotic convergence of approximation schemes

Writing $T(n)$ for the bound of the norm of the interpolation error we distinguish the following ***types of asymptotic behavior:***

- $\exists p > 0: T(n) \leq n^{-p}$: *algebraic convergence,* with rate $p > 0$,
- $\exists 0 < q < 1: T(n) \leq q^n$: *exponential convergence*

The bounds are assumed to be sharp in the sense, that no bounds with larger rate p or smaller q can be found.

6.2.2.2 Interpolnds of Finite Smoothness

Now we aim to establish bounds for the supremum norm of the interpolation error of Lagrangian interpolation similar to the result of Jackson's best approximation theorem.

Theorem: L^∞ **polynomial best approximation estimate**

If $f \in C^r([-1,\,1])$ (r times continuously differentiable), $r \in \mathbb{N}$, then, for any polynomial degree $n \geq r$,

$$\inf_{p \in \mathcal{P}_n} ||f-p||_{L^\infty([-1,1])} \leq \left(1+\frac{\pi^2}{2}\right)^r \frac{(n-r)!}{n!} \left|\left|f^{(r)}\right|\right|_{L^\infty([-1,1])},$$

where

$$\left|\left|f^{(r)}\right|\right|_{L^\infty([-1,1])} := \max_{x\in[-1,1]} |f^{(r)}(x)|.$$

Theorem: Representation of interpolant error

We consider $f \in C^{n+1}(I)$ and the Lagrangian interpolation approximation scheme for a node set $\mathcal{T} := \{t_0, ..., t_n\} \subset I$. Then for every $t \in I$ there exists a $\tau_t \in]\min\{t, t_0, ..., t_n\}, \max\{t, t_0, ..., t_n\}[$ such that

$$f(t) - L_{\mathcal{T}}(f)(t) = \frac{f^{(n+1)}(\tau_t)}{(n+1)!} \cdot \prod_{j=0}^{n}(t-t_j).$$

Lemma: For $f \in C^{n+1}(I)$ let $I_{\mathcal{T}} \in \mathcal{P}_n$ stand for the unique Lagrange interpolant of f in the node set $\mathcal{T} := \{t_0, ..., t_n\} \subset I$. Then for all $t \in I$ the interpolation error is

$$f(t) - I_{\mathcal{T}}(f)(t) = \int_0^1 \int_0^{\tau_1} \cdots \int_0^{\tau_{n-1}} \int_0^{\tau_n} f^{(n+1)}(t_0 + \tau_1(t_1-t_0)) + \cdots + \tau_n(t_n - t_{n-1}) + \tau(t-t_n) \mathrm{d}\tau \mathrm{d}\tau_n ... \mathrm{d}\tau_1 \cdot \prod_{j=}^{r}$$

6.2.2.3 Analytic Interpolands

Definition: Real-analytic functions

A function $f \in C^\infty(I)$ defined on an open interval $I \subset \mathbb{R}$ is called ***real-analytic*** on I, if it possesses a convergent Taylor series at every point $t_0 \in I$:

$$\forall t_0 \in I : \exists \rho = \rho(t_0) > 0 : f(t) = \sum_{k=0}^{\infty} \frac{f^{(k)}(t_0)}{k!}(t-t_0)^k, \; \forall t \in I : |t-t_0| < \rho(t_0).$$

$\rho(t_0)$ is called ***radius of convergence*** of the Taylor series.

We may say that an analytic function locally agrees with a "polynomial of degree ∞", because this is exactly what a convergent ***power series***

$$f(t) = \sum_{k=0}^{\infty} a_k(t-t_0), \; a_k \in \mathbb{R},$$

represents.

A first glimpse of the relevance of analyticity for polynomial approximation: Let $I \subset \mathbb{R}$ be a closed interval and f real-analytic on I. We add a stronger assumption: There There is a $t_0 \in I$ and $\rho > 0$ sucht that

- The Taylor seriers of f at t_0

  $$f \sum_{k=0}^{\infty} a_k(t-t_0)^k, \; a_k \in \mathbb{R},$$

  has radius of convergence ρ,
- I is contained int the ρ-ball around t_0

  $$I \subset \{t \in \mathbb{R} : |t-t_0| < \rho\}.$$

We approximate f by its ***truncated Taylor series***

$$p_n(t) := \sum_{k=0}^{n} a_k(t-t_0)^k \in \mathcal{P}_n.$$

Since I is closed we can find $0 < r < \rho$ such that

$$I \subset \{t \in \mathbb{R} : |t - t_0| \leq r\}.$$

The convergence theory of power series ensures that for any $r < R < \rho$

$$\sum_{k=0}^{\infty} |a_k| R^k =: C \leq \infty.$$

Combining the three statements from above: for arbitrary $t \in I$ we arrive at

$$\Rightarrow ||f - p_n||_{L^\infty(I)} \leq Cq^{n+1} \;\; \forall n \in \mathbb{N} \text{ with } q := \frac{r}{R} < 1.$$

This confirms ***exponential convergence*** of the approximation error incurred by truncating the Taylor series.

Definition: Analyticity of a complex valued function

Let $D \subset \mathbb{C}$ be an open set in the complex plane. A function $f : D \to \mathbb{C}$ is called ***complex-analytic/holomorphic*** in D, for every point $z \in D$ one can find $\rho(z) > 0$ and $a_k \in \mathbb{C}$, $k \in \mathbb{N}_0$, such that

$$f(w) = \sum_{k=0}^{\infty} a_k (w - z)^k \;\; \forall w \in D : |z - w| < \rho(z).$$

Theorem: Residue theorem

Let $D \subset \mathbb{C}$ be an open set, $G \subset D$ a closed set contained in D, $\gamma := \psi D$ its boundary, and Π a finite set contained in the interior of G. Then each function f that is analytic in $D \setminus \Pi$ holds

$$\frac{1}{2\pi i} \int_\gamma f(z) \mathrm{d}z = \sum_{p \in \Pi} \mathrm{res}_p f,$$

where $\mathrm{res}_p f$ is the ***residual*** of f in $p \in \mathbb{C}$.

6.2.3 Chebychev Interpolation

As pointed out, when we build approximation schemes from interpolation schemes, we have the extra freedom to choose the sampling points (= interpolation nodes). Now, based on the insight into the structure of the interpolation errors, we seek to choose "optimal" sampling points. They will give rise to the so-called ***Chebychev polynomial*** approximation schemes, also known as ***Chebychev interpolation.***

6.2.3.1 Motivation and Definition

We are given the following setting:

- Without loss of generality: $I = [-1, 1]$
- Interpoland: $f : I \to \mathbb{R}$ at least continuous, $f \in C^0(I)$
- Set of interpolation nodes: $\mathcal{T} := \{-1 \leq t_0 < t_1 < \cdots < t_n \leq 1\}$, $n \in \mathbb{N}$

We recall the following theorem: $||f - L_\mathcal{T} f||_{L^\infty(I)} \leq \frac{1}{(n+1)!} \cdot ||f^{(n+1)}||_{L^\infty(I)} \cdot ||w||_{L^\infty(I)}$, with *nodal polynomial* $w(t) := (t - t_0) \cdot (t - t_1) \cdots (t - t_n)$.

Optimal choice of interpolation nodes independent of interpoland

The idea is as follows: choose nodes $t_0, \ldots, t_n$ such that $||w||_{L^\infty(I)}$ is minimal! This is equivalent to finding a polynomial $q \in \mathcal{P}_{n+1}$

- with leading coefficient $= 1$
- such that it minimizes the norm $||q||_{L^\infty(I)}$.

→ Then choose nodes $t_0, \ldots, t_n$ as ***zeros*** of q.

Definition: Chebychev polynomials

The n^{th} ***Chebyshev polynomial*** is defined as

$$T_n(t) := \cos(n \cdot \arccos(t)), \quad -1 \leq t \leq 1, \, n \in \mathbb{N}.$$

Theorem: The function T_n defined above satisfies the ***3-term recursion***

$$T_{n+1}(t) = 2t \cdot T_n(t) - T_{n-1}(t), \quad T_0 = 1, \, T_1 = t, \, n \in \mathbb{N}.$$

This theorem implies the following statements:

- $T_n \in \mathcal{P}_n$
- Their leading coefficients are equal to 2^{n-1}
- The T_n are linearly independent
- $\{T_j\}_{j=0}^n$ is a basis of $\mathcal{P}_n = \mathrm{Span}\{T_0, \ldots, T_n\}, \, n \in \mathbb{N}_0$

```
// Computes the values of the Chebychev polynomials at points passed
// in x using the 3-term recursion.
// The values T_k(x_j) are returned in row k+1 of V
void chebpolmult(const int d, const RowVectorXd &x, MatrixXd &V) {
  const unsigned n = x.size();
  V = MatrixXd::Ones(d+1, n); //T_0 = 1
  V.block(1, 0, 1, n) = x;    //T_1(x) = x
  for(int k = 1; k < d; ++k) {
    const RowVectorXd p = V.block(k, 0, 1, n);      //p = T_k
    const RowVectorXd q = V.block(k-1, 0, 1, n):    //q = T_{k-1}
    V.block(k+1, 0, 1, n) = 2*x.cwiseProduct(p) - q //3-term recursion
  }
}
```

Furthermore note, that the *zeros* of T_n are given by

$$t_k = \cos\left(\frac{2k+1}{2n} \cdot \pi\right), \quad k = 0, \ldots, n-1.$$

Using the unqiue affine transformation we can transform the Chebychev nodes from $[-1, \, 1]$ to $[a, \, b]$ the following way:

$$\hat{t} \in [-1, \, 1] \to t := a + \frac{1}{2}(\hat{t} + 1)(b - a) \in [a, \, b].$$

The ***Chebychev nodes*** in the interval $I = [a, \, b]$ are given by

$$t_k := a + \frac{1}{2}(b - a) \cdot \left(\cos\left(\frac{2k+1}{2(n+1)} \cdot \pi\right) + 1\right), \quad k = 0, \ldots, n.$$

Parlance: When we use Chebychev nodes for polynomial interpolation we call the resulting Lagrangian approximation scheme ***Chebychev interpolation.***

6.2.3.2 Chebychev Interpolation Error Estimates

Note the following features of Chebychev interpolation on the interval $[-1, \, 1]$:

- Use of "optimal" interpolation nodes $\mathcal{T} = \{\hat{t} := \cos\left(\frac{2k+1}{2(n+1)} \cdot \pi\right), \, k = 0, \ldots, n\}$
- Corresponding to the nodal polynomial $w(t) = (t - t_0) \cdots (t - t_n) = 2^{-n} T_{n+1}(t)$, $||w||_{L^\infty(I)} = 2^{-n}$, with leading coefficient 1.

Theorem: Representation of interpolation error

We consider $f \in C^{n+1}(I)$ and the Lagrangian interpolation approximation scheme for a node set $\mathcal{T} := \{t_0, \ldots, t_n\} \subset I$. Then, for every $t \in I$ there exists $\tau_t \in]\min\{t, \, t_0, \ldots, t_n\}, \, \max\{t, \, t_0, \ldots, t_n\}[$ such that

$$f(t) - L_{\mathcal{T}}(f)(t) = \frac{f^{(n+1)}(\tau_t)}{(n+1)!} \cdot \prod_{j=0}^{n} (t - t_j).$$

By the above theorem we immediately get an interpolation error estimate for Chebychev interpolation of $f \in C^{n+1}([-1,\,1])$:

$$||f - I_{\mathcal{T}}(f)||_{L^\infty([-1,\,1])} \leq \frac{2^{-n}}{(n+1)!} \cdot \left|\left|f^{(n+1)}\right|\right|_{L^\infty([-1,\,1])}.$$

Estimates for the Chebychev interpolation error on $[a,\,b]$ are given as follows:

$$||f - I_{\mathcal{T}}(f)||_{L^\infty(I)} = \left|\left|\hat{f} - I_{\mathcal{T}}(\hat{f})\right|\right|_{L^\infty([-1,\,1])} \leq \frac{2^{-2n-1}}{(n+1)!}|I|^{n+1}\left|\left|f^{(n+1)}\right|\right|_{L^\infty(I)}.$$

Given the following task:

- Given: polynomial degree $n \in \mathbb{N}$ and a continuous function $f : [-1,\,1] \to \mathbb{R}$
- Sought: efficient representation/evaluation of Chebychev interpolant $p \in \mathcal{P}_n$, interpolant of degree $\leq n$ in Chebychev nodes on $[-1,\,1]$

More concretely, this boils down to a implementation of the following class:

```
class ChebInterp {
  private:
  // various internal data describing Chebychev interpolating polynomial p
  public:
  // constructor taking function f and degree n as arguments
  template <typename Function>
    PolyInterp(const Function &f, unsigned int n);
  // evaluation operator: y_j = p(x_j), j = 1,..., m
  template <typename Vector>
  void eval(const Vector &x, Vector &y) const;
}
```

The key idea is to internally represent p as a linear combination of Chebychev polynomials, a ***Chebychev expansion:***

$$p(t) = \sum_{j=0}^{n} \alpha_j T_j(t), \;\; t \in \mathbb{R},\, \alpha_j \in \mathbb{R},$$

where T_j is the Chebychev polynomial of degree j.

Let us now assume that the Chebychev expansion coefficients α_j are given and wonder, how we can efficiently compute $p(x)$ for some $x \in \mathbb{R}$:

Task: Given $n \in \mathbb{N},\, x \in \mathbb{R}$, and the Chebychev expansion coefficients $\alpha_j \in \mathbb{R},\, j = 0, \ldots, n$ compute $p(x)$ with

$$p(x) = \sum_{j=0}^{n} \alpha_j T_j(x), \;\; \alpha_j \in \mathbb{R}.$$

Idea: Use the 3-term recurrence $T_j(x) = 2xT_{j-1}(x) - T_{j-2}(x), \;\; j = 2,\,3, \ldots$, to design a recursive evaluation scheme. We recover the point value $p(x)$ as the point value of another polynomial of degree $n-1$ with known Chebychev expansion:

$$p(x) = \sum_{j=0}^{n-1} \tilde{\alpha}_j T_j(x) \text{ with } \tilde{\alpha}_j = \begin{cases} \alpha_j + 2x\alpha_{j+1}, & \text{if } j = n-1, \\ \alpha_j - \alpha_{j+2}, & \text{if } j = n-2, \\ \alpha_j, & \text{else.} \end{cases}$$

```
// Recursive evaluation of a polynomial p = sum_{j=1}^{n+1} a_jT_{j-1}
// at point x
// In: Vector of coefficients "a", evaluation point "x"
// Out: Value at point x
double recclenshaw(const VectorXd &a, const double x) {
  const VectorXd::Index n = a.size()-1;
  if(n == 0) return a(0);
  else if(n == 1) return (x * a(1) + a(0));
  else {
    VectorXd new_a(n);
    new_a << a.head(n-2), a(n-2)-a(n), a(n-1)+ 2*x*a(n);
    return recclenshaw(new_a, x); //recursion
```

```
  }
}
```

The above implementation yields a computational effort of $O(nm)$ for an evaluation at m points, $m, n \to \infty$.

```
// efficiently compute coefficients alpha_j in the Chebychev expansion
// p = sum_{j=0}^n alpha_jT_j of p in P_n based on values y_k in
// Chebychev nodes t_k
// In: values y_k passed in "y"
// Out: coefficients alpha_j
VectorXd chebexp(const VectorXd &y) {
  const int n = y.size() - 1; //degree of polynomial
  const std::complex<double> M_l(0, 1); //imaginary unit
  VectorXcd b(2 * (n+1));
  const std::complex<double> om = -M_l * (M_Pl * n) / ((double)(n+1));
  for(int j = 0; j <= n; ++j) {
    b(j) = std::exp(om * double(j)) * y(j);
    b(2 * n + 1 - j) = std::exp(om * double(2 * n + 1 - j)) * y(j);
  }
  Eigen::FFT<double> fft;
  VectorXcd c = fft.inv(b);
  VectorXd beta(c.size());
  const std::complex<double> sc = M_Pl_2/(n + 1) * M_l;
  for(unsigned j = 0; j < c.size(); ++j) {
    beta(j) = (std::exp(sc * double(-n+j)) * c[j]).real();
  }
  VectorXd alpha = 2 * beta.segment(n, n); alpha(0) = beta(n);
  return alpha;
}
```

6.5 Approximation by Trigonometric Polynomials

Now we address the approximation of a continuous 1-periodic function

$$f \in C^0(\mathbb{R}),\ f(t+1) = f(t)\ \ \forall t \in \mathbb{R}.$$

6.5.1 Approximation by Trigonometric Interpolation

- Idea: Adapt the policy of approximation by interpolation
- Here: Employ trigonometric interpolation from Section 5.6 into space $\mathcal{P}_{2n}^T$ of 1-periodic trigonometric polynomials

---

Recall: Trigonometric interpolation

Given nodes $t_0 < t_1 < \cdots < t_{2n}$, $t_k \in [0,\ 1[$, and values $y_k \in \mathbb{R}$, $k = 0, \ldots, 2n$, find

$$q \in \mathcal{P}_{2n}^T := \text{Span}\{t \to \cos(2\pi jt),\ t \to \sin(2\pi jt)\}_{j=0}^n,$$

with $q(t_k) = y_k$ for all $k = 0, \ldots, 2n$.

Terminology: $\mathcal{P}_{2n}^T =$ space of ***trigonometric polynomials*** of degree $2n$.

Trigonometric approximation of generic *1-periodic* continuous functions $\in C^0(\mathbb{R})$ in $\mathcal{P}_{2n}^T$ usually relies on ***equidistant*** interpolation nodes $t_k = \frac{k}{2n+1}$, $k = 0, \ldots, 2n$.

Notation: We denote the ***trigonometric interpolation operator*** in $2n+1$ equidistant nodes $t_k = \frac{k}{2n+1}$, $k = 0, \ldots, 2n$

$$T_n : C^0([0,\ 1[) \to \mathcal{P}_{2n}^T,\ \ T_n(f)(t_k) = f(t_k)\ \ \forall k \in \{0, \ldots, 2n\}.$$

6.5.2 Trigonometric Interpolation Error Estimates

We use the notation T_n for trigonometric interpolation in the $2n+1$ equidistant nodes $t_k := \frac{k}{2n+1}$. Our focus will be on the asymptotic behavior of

$$||f - T_n f||_{L^\infty([0,\ 1[)} \text{ and } ||f - T_n f||_{L^2([0,\ 1[)} \text{ as } n \to \infty,$$

for functions $f : [0,\ 1[\to \mathbb{C}$ with different ***smoothness properties.***

We know that the complex vector space of trigonometric polynomials $\mathcal{P}_{2n}^T$ is spanned by the $2n+1$ ***Fourier modes*** $t \to \exp(2\pi ikt)$ of "lowest frequency", that is, for $-n \le k \le n$

$$p \in \mathcal{P}_{2n}^T \Rightarrow p(t) = \sum_{k=-n}^{n} a_k \exp(-2\pi i t) \text{ for some } a_k \in \mathbb{C}.$$

Now let us make a connection: We learned that every function $f : [0,\, 1[\to \mathbb{C}$ with finite $L^2([0,\, 1])$-norm

$$||f||^2_{L^2([0,\,1])} := \int_0^1 |f(t)|^2 \, \mathrm{d}t < \infty,$$

can be expanded in a ***Fourier series***

$$f(t) = \sum_{k=-\infty}^{\infty} \hat{f}_k \exp(-2\pi i k t) \text{ in } L^2([0,\, 1]), \;\; \hat{f}_k := \int_0^1 f(t) \exp(2\pi i k t) \, \mathrm{d}t.$$

We add, that a limit in $L^2([0,\, 1])$ means that $\left|\left| f - \sum_{k=-M}^{M} \hat{f} \exp(-2\pi i k) \right|\right|_{L^2([0,\,1])} \to 0$ for $M \to \infty$. Also note the customary notation $\hat{f}_k$ for the ***Fourier coefficients.***

A fundamental result about functions given through Fourier series was the following fundamental ***isometry property*** of the mapping taking a function to the sequence of its Fourier coefficients.

Theorem: Isometry property of the Fourier transform

If the Fourier coefficients satisfy $\sum_{k\in\mathbb{Z}} |\hat{c}_j|^2 < \infty$, then the Fourier series

$$c(t) = \sum_{k\in\mathbb{Z}} \hat{c}_k \exp(-2\pi i k t)$$

yields a function $c \in L^2([0,\, 1])$ that satisfies

$$||c||^2_{L^2([0,\,1])} := \int_0^1 |c(t)|^2 \, \mathrm{d}t = \sum_{k\in\mathbb{Z}} |\hat{c}_j|^2.$$

We study the action of the trigonometric interpolation operator T_n on individual ***Fourier modes***

$$\mu_k(t) := \exp(-2\pi i k t), \;\; t \in \mathbb{R},\, k \in \mathbb{Z}.$$

Due to the 1-periodicity of $t \to \exp(-2\pi i t)$ we find for every node $t_j := \frac{j}{2n+1}$, $j = 0, ..., 2n$,

$$\mu_k(t_j) = \exp(-2\pi i k \frac{j}{2n+1}) = \exp(-2\pi i (k - l(2n+1)) \frac{j}{2n+1}) = \mu_{k-l(2n+1)}(t_j) \;\; \forall l \in \mathbb{Z}.$$

Since $T_n \mu_k = \mu_k$ for $k = -n, ..., n$, that is, for $u_k \in \mathcal{P}_{2n}^T$, the aliasing effect yields

$$T_n u_k = u_{\tilde{k}},\, \tilde{k} \in \{-n, ..., n\},\, k - \tilde{k} \in (2n+1)\mathbb{Z} \, [\tilde{k} := k \bmod (2n+1)].$$

6.5.3 Trigonometric Interpolation of Analytic Periodic Functions

---

Theorem: If $f : \mathbb{R} \to \mathbb{C}$ is 1-periodic and has an ***analytic extension*** to the strip

$$\overline{S} := \{z \in \mathbb{C} : -\eta \leq \mathrm{Im}(z) \leq \eta\}, \text{ for some } \eta > 0,$$

then its Fourier coefficients decay according to

$$|\hat{f}_j| \leq q^{|k|} \cdot ||f||_{L^\infty(\overline{S})} \;\; \forall k \in \mathbb{Z} \text{ with } q := \exp(-2\pi\eta) \in]0,\, 1[.$$

Theorem: Let $f : D \to \mathbb{C}$ be analytic in $D \subset \mathbb{C}$ and $U \subset D$ be simply connected and strictly contained in D. Then

$$\int_{\phi U} f(z) \, \mathrm{d}z = 0.$$

Theorem: If $f : \mathbb{R} \to \mathbb{C}$ is 1-periodic and possesses an ***analytic extension*** to the strip

$$\overline{S} := \{z \in \mathbb{C} : -\eta \leq \mathrm{Im}(z) \leq \eta\}, \text{ for some } \eta > 0,$$

then there is $C_\eta > 0$ depending only on η such that

$$||f - T_n f||_* \leq C_\eta e^{-\pi\eta n} ||f||_{L^\infty(\overline{S})}, \ \ n \in \mathbb{N}, \, (* = L^2(]0, 1[), \, L^\infty(]0, 1[)).$$

6.6 Approximation by Piecewise Polynomials

Recall some alternatives to interpolation by global polynomials discussed in Chapter 5:

- piecewise linear/quadratic interpolation
- cubic Hermite interpolation
- cubic spline interpolation

All these interpolation schemes rely on ***piecewise polynomials*** (of different global smoothness).

6.6.1 Piecewise Polynomial Lagrange Interpolation

Given an interval $[a, b] \subset \mathbb{R}$ endowed with ***mesh*** $\mathcal{M} := \{a = x_0 < x_1 < \cdots < x_{m-1} < x_m = b\}$.

General local Lagrange interpolation on a mesh (PPLIP)

1. Choose local degree $n_j \in \mathbb{N}_0$ for each cell of the mesh, $j = 1, ..., m$
2. Choose set of local interpoaltion nodes

$$\mathcal{T}^j := \{t_0^j, ..., t_{n_j}^j\} / subset I_j := [x_{j-1}, x_j], \, j = 1, ..., m,$$

   for each mesh cell/grid interval I_j.

3. Define ***piecewise polynomial*** interpolants $s : [x_0, x_m] \to \mathbb{K}$:

$$s_j := s_{|I_j} \in \mathcal{P}_{n_j} \text{ and } s_j(t_i^j) = f(t_i^j), \ \ i = 0, ..., n_j, \, j = 1, ..., m.$$

Corollary: The mapping $f \to s$ defines a *linear operator* $I_\mathcal{M} : C^0([a, b]) \to C^0_{\mathcal{M}, \mathrm{pw}}([a, b])$.

6.6.2 Cubic Hermite Interpolation: Error Estimates

Definition: Given $f \in C^1([a, b])$ and a mesh $\mathcal{M} := \{a = x_0 < x_1 < \cdots < x_{m-1} < x_m = b\}$ the ***piecewise cubic Hermite interpolant*** $s : [a, b] \to \mathbb{R}$ is defined as

$$s_{|[x_{j-1}, x_j]} \in \mathcal{P}_3, \ \ j = 1, ..., m, \ \ s(x_j) = f(x_j), \ \ s'(x_j) = f'(x_j), \ \ j = 0, ..., m.$$

Theorem: Let s be the cubic Hermite interpolant of $f \in C^4([a, b])$ on a mesh $\mathcal{M} := \{a = x_0 < x_1 < \cdots < x_{m-1} < x_m = b\}$. Then

$$||f - s||_{L^\infty([a, b])} \leq \frac{1}{4!} h_\mathcal{M}^4 \left|\left| f^{(4)} \right|\right|_{L^\infty([a, b])},$$

with the meshwidth $h_\mathcal{M} := \max_j |x_j - x_{j-1}|$.

6.6.3 Cubic Spline Interpolation: Error Estimates

We take $I = [-1, 1]$ and rely on an equidistant mesh (knot set) $\mathcal{M} := \{-1 + \frac{1}{2} \cdot j\}_{j=0}^n, \, n \in \mathbb{N} \to h = \frac{2}{n}$.

We study h-convergence of complete cubic spline interpolation, where the slopes at the endpoints of the interval are made to agree with the derivatives of the interpoland at these points. As *interpolands* we consider

$$f_1(t) = \frac{1}{1 + e^{-2t}} \in C^\infty(I), \, f_2(t) = \begin{cases} 0, & \text{if } t < -\frac{2}{5}, \\ \frac{1}{2}(1 + \cos(\pi(t - \frac{3}{5}))), & \text{if } -\frac{2}{5} < t < \frac{3}{5} \in C^1(I), \\ 1, & \text{otherwise} \end{cases}$$

We remark that there is the following theoretical result:

$$f \in C^4([t_0, t_n]), \, ||f - s||_{L^\infty([t_0, t_n])} \leq \frac{5}{384} h^4 \left|\left|f^{(4)}\right|\right|_{L^\infty([t_0, t_n])}.$$

6.7 Summary and Learning Outcomes

- You should be able to extract the asymptotic convergence of approximation errors from empriric data.
- You should know a few relevant norms and spaces of functions.
- You should be able to construct an approximation scheme on an arbitrary interval from an interpolation scheme on a fixed interval.
- You should recall bounds for the pointwise approximation error of polynomial interpolation for functions in C^r and analytic functions.
- You should be familiar with Chebychev interpolation: rationale, definition and algorithms.
- You should know about trigonometric interpolation and the behaviour of the associated pointwise approximation errors.
- You should be able to predict the convergence of piecewise polynomial interpolation in terms of meshwidth $h \to 0$.

7. Numerical Quadrature

7.1 Introduction

Numerical quadrature deals with the *approximate numerical evaluation of integrals* $\int_\Omega f(x) \, \mathrm{d}x$ for a given closed integration domain $\Omega \subset \mathbb{R}^d$. Thus, the underlying problem is the mapping

$$I : C^0(\Omega) \to \mathbb{R}, \, f \to \int_\Omega f(x) \, \mathrm{d}x,$$

with data space $X := C^0(\Omega)$ and result space $Y := \mathbb{R}$.

The integrand f, a continuous function $f : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ should not be thought of as given by an analytic expression, but as given in *procedural form*. In this chapter, the focus is on the special case $d = 1, \, \Omega = [a, \, b]$.

7.2 Quadrature Formulas - Quadrature Rules

Definition: An n-point ***quadrature formula/quadrature rule*** **(QR)** on $[a, \, b]$ provides an approximation of the value of an integral through *weighted sum* of point values of the integrand:

$$\int_a^b f(t) \, \mathrm{d}t \simeq Q_n(f) := \sum_{j=1}^n w_j^{(n)} f(c_j^{(n)}).$$

We use the following terminology:

- $w_j^{(n)}$: ***quadrature weights*** $\in \mathbb{R}$
- $c_j^{(n)}$: ***quadrature nodes*** $\in [a, \, b]$

```
template <class Function>
double quadformula(Function &f, const VectorXd &c, const VectorXd &w) {
  const std::size_t n = c.size();
  double l = 0;
  for(std::size_t i = 0; i < n; ++i) { l += w(i)*f(c(i)); }
  return l;
}
```

We learned that an approximation scheme for any interval could be obtained from an approximation scheme on a single *reference interval*, for example $[-1, \, 1]$ by means of ***affine pullback.*** A similar affine transformation technique makes it possible to derive quadrature formulas for an arbitrary interval from a single quadrature formula on a reference interval.

- Given: quadrature formula $(\hat{c}_j, \, \hat{w}_j)_{j=1}^n$ on *reference interval* $[-1, \, 1]$

- Idea: transformation formula for integrals

$$\int_a^b f(t)\,\mathrm{d}t = \frac{1}{2}(b-a)\int_{-1}^{1}\hat{f}(\tau)\,\mathrm{d}\tau,$$
$$\hat{f}(\tau) := f\Big(\frac{1}{2}(1-\tau)a + \frac{1}{2}(\tau+1)b\Big).$$

We therefore can state the ***quadrature formula*** for a given interval $[a,\, b],\ a, b \in \mathbb{R}$:

$$\int_a^b f(t)\,\mathrm{d}t \simeq \frac{1}{2}(b-a)\sum_{j=1}^{n}\hat{w}_j\hat{f}(\hat{c}_j) = \sum_{j=1}^{n} w_j f(c_j)$$
$$\text{with } \begin{cases} c_j = \frac{1}{2}(1-\hat{c}_j)a + \frac{1}{2}(1+\hat{c}_j)b, \\ w_j = \frac{1}{2}(b-a)\hat{w}_j. \end{cases}$$

Lemma: Every linear interpolation operator $I_{\mathcal{T}}$ spawns a quadrature formula.

Summing up, we have found

$$\text{interpolation schemes} \rightarrow \text{approximation schemes} \rightarrow \text{quadrature schmes}$$

In general, the quadrature formula will only provide an *approximation valueof the integral.* For a generic integrand we will encounter a non-vanishing ***quadrature error:***

$$E_n(f) := \Big|\int_a^b f(t)\,\mathrm{d}t - Q_n(f)\Big|.$$

As in the case of function approximation by interpolation, our focus will on the *asmyptotic behavior* of the quadrature error as a function of the number n of point evaluations of the integrand.

Therefore consider ***families of quadrature rules*** $\{Q_n\}_n$ described by

- ***quadrature weights*** $\{w_j^n,\, j = 1, ..., n\}_{n\in\mathbb{N}}$
- ***quadrature nodes*** $\{c_j^n,\, j = 1, ..., n\}_{n\in\mathbb{N}}$

We study the *asymptotic* behavior of the *quadrature error* $E(n)$ for $n \rightarrow \infty$.

Bounds for the maximum norm of the approximation error of an approximation scheme directly translate into estimates of the quadrature error of the induced quadrature scheme:

$$\Big|\int_a^b f(t)\,\mathrm{d}t - QA(f)\Big| \leq \int_a^b |f(t) - A(f)|\,\mathrm{d}t \leq |b-a| \cdot ||f - A(f)||_{L^\infty([a,\, b])}$$

7.3 Polynomial Quadrature Formulas

Now we specialize the general recipe for approximation schemes based on global polynomials, the Lagrange approximation scheme as introduced Section 6.2.

- Idea: replace integrand f with $p_{n-1} := I_{\mathcal{T}} \in \mathcal{P}_{n-1}$ = polynomial Lagrange interpoland of f for given node set $\mathcal{T} := \{t_0, ..., t_n\} \subset [a,\, b]$

$$\int_a^b f(t)\,\mathrm{d}t \simeq Q_n(f) := \int_a^b p_{n-1}(t)\,\mathrm{d}t.$$

The cardinal interpolants for Lagrange interpolation are the Lagrange polynomials

$$L_i(t) := \prod_{j=0,\, j\neq i}^{n-1}\frac{t-t_j}{t_i-t_j},\ i = 0, ..., n-1 \Rightarrow p_{n-1}(t) = \sum_{i=0}^{n-1} f(t_i)L_i(t).$$

Then we can conclude the n-point ***quadrature formula:***

$$\int_a^b p_{n-1}(t)\,\mathrm{d}t = \sum_{i=0}^{n-1} f(t_i)\int_a^b L_i(t)\,\mathrm{d}t \Rightarrow \begin{cases}\text{nodes } c_i = t_{i-1},\\ \text{weights } w_i := \int_a^b L_{i-1}(t)\,\mathrm{d}t.\end{cases}$$

We use the L^∞-bound for Lagrange interpolation

$$||f - L_{\mathcal{T}} f||_{L^\infty(I)} \leq \frac{||f^{(n+1)}||_{L^\infty(I)}}{(n+1)!}\max_{t\in I}|(t-t_0)\cdots(t-t_n)|,$$

to conclude for any n-point quadrature rule based on polynomial interpolation:

$$f \in C^n([a,\,b]) \Rightarrow \left|\int_a^b f(t)\,\mathrm{d}t - Q_n(f)\right| \leq \frac{1}{n!}(b-a)^{n+1}\left|\left|f^{(n)}\right|\right|_{L^\infty([a,\,b])}.$$

7.4 Gauss Quadrature

7.4.1 Order of a Quadrature Rule

Wo can we gauge the "quality" of an n-point quadrature formula Q_n without testing it for specific integrands? The next definition gives an answer.

Definition: The ***order*** of quadrature rule $q_n : C^0([a,\,b]) \to \mathbb{R}$ is defined as

$$\mathbf{order}(Q_n) := \max\{m \in \mathbb{N}_0 : Q_n(p) = \int_a^b p(t)\,\mathrm{d}t \;\; \forall p \in \mathcal{P}_m\} + 1,$$

that is, the *maximal* degree $+1$ of polynomials for which the quadrature rule is guaranteed to be exact.

Corollary: An affine transformation of a quadrature rule does not change its order.

Further, by construction, all polynomial n-point quadrature rules possess order at least n.

Theorem: Sufficient order conditions for quadrature rules

An n-point quadrature rules on $[a,\,b]$

$$Q_n(f) := \sum_{j=1}^{n} w_j f(c_j), \quad f \in C^0([a,\,b]),$$

with nodes $t_j \in [a,\,b]$ and weights $w_j \in \mathbb{R}$, $j = 1, ..., n$, has ***order*** $\geq n$, **if and only if**

$$w_j = \int_a^b L_{j-1}(t)\,\mathrm{d}t, \;\; j = 1, ..., n,$$

where L_k, $k = 0, ..., n-1$, is the k-th ***Lagrange polynomial*** associated with the ordered node set $\{c_1,\, c_2, ...,\, c_n\}$.

The above theorem provides a concrete formula for quadrature weights, which gurantee order n for an n-point quadrature formula. Yet evaluating integrals of Lagrange polynomials may be cumbersome. Here we give a general recipe for finding the weights w_j according to the above theorem without dealing with Lagrange polynomials.

- Given: arbitrary nodes $c_1, ..., c_n$ for n-point local quadrature formula on $[a,\,b]$

Now if $p_1, ..., p_n$ is *any basis* of $\mathcal{P}_{n-1}$, then, thanks to the linearity of the integral and quadrature formulas,

$$Q_n(p_j) = \int_a^b p_j(t)\,\mathrm{d}t \;\; \forall j = 1, ..., n \iff Q_n \text{ has order} \geq n.$$

We can form the following linear systems of equations:

$$\begin{bmatrix} p_1(c_1) & \cdots & p_1(c_n) \\ \vdots & & \vdots \\ p_n(c_1) & \cdots & p_n(c_n)\end{bmatrix} \cdot \begin{bmatrix} w_1 \\ \vdots \\ w_n\end{bmatrix} = \begin{bmatrix}\int_a^b p_1(t)\,\mathrm{d}t \\ \vdots \\ \int_a^b p_n(t)\,\mathrm{d}t\end{bmatrix}.$$

7.4.2 Maximal-Order Quadrature Rules

A natural question is whether an n-point quadrature formula can achieve an order $> n$. A negative result limits the maximal order that can be achieved:

Theorem: The maximal order of an n-point quadrature rule is $2n$.

If we want to construct n-point quadrature rules with maximal order $2n$, we first need to search for necessary conditions that have to be met by the nodes.

Optimist's assumption: $\exists$ *family* of n-point quadrature formulas on $[-1,\ 1]$

$$Q_n(f) := \sum_{j=1}^{n} w_j^{(n)} f(c_j^{(n)}) \simeq \int_{-1}^{1} f(t)\ \mathrm{d}t,\ \ w_j^{(n)} \in \mathbb{R},\ n \in \mathbb{N},\ \text{of order } 2n \iff \text{ exact for polynomials } \in \mathcal{P}_{2n-1}.$$

We define $\bar{P}_n(t) := (t - c_1^{(n)}) \cdots (t - c_n^{(n)}),\ t \in \mathbb{R} \Rightarrow \bar{P} \in \mathcal{P}_n$.

By assumption on the order of Q_n we know that for any $q \in \mathcal{P}_n$:

$$\int_{-1}^{1} q(t)\bar{P}_n(t)\ \mathrm{d}t = \sum_{j=1}^{n} w_j^{(n)} q(c_j^{(n)}) \underbrace{\bar{P}_n(c_j^{n})}_{=0} = 0.$$

We conclude $L^2([-1,\ 1])$-***orthogonality:*** $\int_{-1}^{1} q(t)\bar{P}_n(t)\ \mathrm{d}t = 0\ \ \forall q \in \mathcal{P}_{n-1}$. We can also give algebraic arguments for existence and uniqueness of $\bar{P}_n$. Switching to a monomial representation of $\bar{P}_n := t^n + \alpha_{n-1}t^{n-1} + \cdots + \alpha_1 t + \alpha_0$, by linearity of the integral, the concluded orthogonality is equivalent to

$$\sum_{j=0}^{n-1} \alpha_j \int_{-1}^{1} t^j t^l\ \mathrm{d}t = -\int_{-1}^{1} t^n t^l\ \mathrm{d}t,\ \ l = 0, ..., n-1.$$

This is a linear system of equations $A[\alpha_j]_{j=0}^{n-1} = b$ with a symmetric, positive definite coefficient matrix $A \in \mathbb{R}^{n,n}$. Hence, A is regular and the coefficients α_j are uniquely determined. Thus, there is omly one n-point quadrature rule of order $2n$.

$\Rightarrow$ The nodes of an n-point quadrature formula of ***order*** $2n$, if it exists, must coincide with the unique zeros of the polynomials $\bar{P}_n \in \mathcal{P}_n \setminus \{0\}$ satisfying the above stated equation for $L^2([-1,\ 1])$-orthogonality.

Theorem: Existence of n-point quadrature formulas of order $2n$

Let $\{\bar{P}_n\}_{n \in \mathbb{N}_0}$ be a family of non-zero polynomials that satisfies

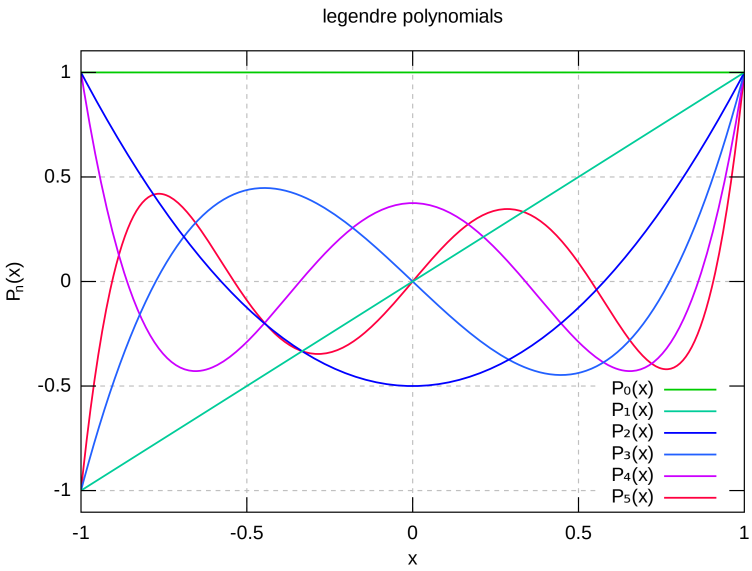
- $\bar{P}_n \in \mathcal{P}_n$,
- $\int_{-1}^{1} q(t)\bar{P}_n(t)\ \mathrm{d}t = 0$ for all $q \in \mathcal{P}_{n-1}$ ($L^2([-1,\ 1])$-orthogonality),
- The set $\{c_j^{(n)}\}_{j=1}^{m}$, $m \le n$, of real zeros of $\bar{P}_n$ is contained in $[-1,\ 1]$.

Then the quadrature rule $Q_n(f) := \sum_{j=1}^{m} w_j^{(n)} f(c_j^{(n)})$ with weights chosen according to the theorem of section 7.4.1 provides a quadrature formula of order $2n$ on $[-1,\ 1]$.

Definition: The n-th ***Legendre polynomial*** P_n is defined by

- $P_n \in \mathcal{P}_n$,
- $\int_{-1}^{1} P_n(t)q(t)\ \mathrm{d}t = 0\ \ \forall q \in \mathcal{P}_{n-1}$,
- $P_n(1) = 1$.

The Legendre polynomials $P_0, ..., P_5$ look as follows:



The ***Gauss points*** $\zeta_j^{(n)}$ are the zeros of the Legendre polynomial P_n.

Lemma: P_n has n distinct zeros, also called Gauss points, in $]-1,\, 1[$.

Definition: The n-point quadrature formulas whose nodes, the Gauss points, are given by the zeros of the n-th Legendre polynomial, and whose weights are chosen according to the theorem from section 7.4.1, are called ***Gauss-Legendre quadrature formulas.***

The famous ***3-term recursion*** for Legendre polynomials is given by

$$P_{n+1}(t) := \frac{2n+1}{n+1} t P_n(t) - \frac{n}{n+1} P_{n-1}(t), \;\; P_0 := 1, \; P_1(t) := t.$$

```
void legendre(const unsigned n, const Vectorxd& x, MatrixXd& L) {
  L = MatrixXd::Ones(n, n);
  L.col(1) = x;
  for(unsigned j = 1; j < n - 1; ++j) {
    L.col(j + 1) = (2. * j + 1) / (1. + j) + L.col(j - 1).cwiseProduct(x)
    - j / (1. + j) * L.col(j - 1);
  }
}
```

7.4.3 Quadrature Error Estimates

The Gauss-Legendre quadrature formulas do not only enjoy maximal order, but another key property that can be regarded as essential for viable families of quadrature rules.

Lemma: The weights of the Gauss-Legendre quadrature formulas are positive.

Theorem: Quadrature error estimate for quadrature rules with positive weights

For every n-point quadrature rule Q_n of order $q \in \mathbb{N}$ with weights $w_j \geq 0,\, j = 1, \ldots, n$ the quadrature error satisfies

$$E_n(f) := \left| \int_a^b f(t)\,\mathrm{d}t - Q_n(f) \right| \leq 2|b-a| \inf\nolimits_{p \in \mathcal{P}_{q-1}} \underbrace{\|f - p\|_{L^\infty([a,\,b])}}_{\text{best approximation error}} \quad \forall f \in C^0([a,\,b]).$$

Lemma: For every n-point quadrature rule Q_n of order $q \in \mathbb{N}$ with weights $w_j \geq 0,\, j = 1, \ldots, n$ we find that the quadrature error $E_n(f)$ for an integrand $f \in C^r([a,\,b]),\, r \in \mathbb{N}_0$, satisfies

$$\begin{cases} \text{if } q \geq r, & E_n(f) \leq C q^{-r} |b-a|^{r+1} \left\| f^{(r)} \right\|_{L^\infty([a,\,b])}, \\ \text{else} & E_n(f) \leq \frac{|b-a|^{q+1}}{q!} \left\| f^{(q)} \right\|_{L^\infty([a,\,b])} \end{cases},$$

with a constant $C > 0$ independent of n, f and $[a, b]$.

7.5 Composite Quadrature

The so-called composite quadrature rules on an interval $[a, b]$ follows a similar idea to approximation by piecewise polynomial interpolants. Analogously, they start from a grid/mesh

$$\mathcal{M} := \{a = x_0 < x_1 < \cdots x_{m-1} < x_m = b\}$$

and appeal to the trivial identity

$$\int_a^b f(t)\, \mathrm{d}t = \sum_{j=1}^{m} \int_{x_{j-1}}^{x_j} f(t)\, \mathrm{d}t.$$

On each mesh interval $[x_{j-1}, x_j]$ we then use a ***local quadrature rule.***

General construction of composite quadrature rules

- Partition integration domain $[a, b]$ by a ***mesh/grid*** $\mathcal{M}$
- Apply quadrature formulas from section 7.3 and 7.4 *locally* oh the mesh interval $I_j := [x_{j-1}, x_j]$ and sum up

---

Lemma: The n-point quidistant trapezoidal quadrature rule

$$\int_a^b f(t)\, \mathrm{d}t \simeq T_n(f) := h\Big(\frac{1}{2}f(a) + \sum_{k=1}^{n-1} f(kh) + \frac{1}{2}f(b)\Big),\ h := \frac{b-a}{n}$$

is exact for trigonometric polynomials of degree $\leq 2n - 2$.

7.6 Adaptive Quadrature

The policy of ***adaptive quadrature*** approximates $\int_a^b f(t)\, \mathrm{d}t$ by a quadrature formula, whose nodes c_j^n are chosen depending on the integrand f.

We distinguish:

1. ***a priori*** adaptive quadrature: the nodes are fixed before the evaluation of the quadrature formula, taking into account external information about f
2. ***a posteriori*** adaptive quadrature: the node positions are chosen or improved based on information gained during the computation inside the a loop. It termiantes when sufficient accuracy has been reached

Lemma: Let $f : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ be a convex function with $f(0) = 0$ and $x > 0$. Then the constrained minimization problem: seek $\zeta_1, \ldots, \zeta_m \in \mathbb{R}_0^+$ such that

$$\sum_{k=1}^{m} f(\zeta_k) \to \min \text{ and } \sum_{k=1}^{m} \zeta_k = x,$$

has the solution $\zeta_1 = \zeta_2 = \cdots = \zeta_m = \frac{x}{m}$.

7.7 Learning Outcomes

- You should know what is a quadrature formula and terminology connected with it
- You should be able to transform quadrature formulas to arbitrary intervals
- You should understand how an interpolation and approximation schemes spawn quadrature formulas and how quadrature errors are connected to interpolation/approximation errors
- You should be able to compute the weights of polynomial quadrature formulas
- You should know the concept of order of a quadrature rule and why it is invariant under affine transformation
- You should remember the maximal and minimal order of polynomial quadrature rules
- You should know the order of the n-point gauss-Legendre quadrature rule

- You should understand why Gauss-Legendre quadrature converges exponentially for integrands that can be extended analytically and algebraically for integrands with limited smoothness
- You should be able to apply regularizing transformations to integrals with non-smooth integrands
- You should know about asymptotic convergence of the h-version of composite quadrature
- You should know the principles of adaptive composite quadrature