Prof. Ch. Schwab
B. Fitzpatrick
J. Zech

Spring Term 2016

Numerische Mathematik I

ETH Zürich
D-MATH

Homework Problem Sheet 5

For some problems, parts of the solution are already given. Fill in the gaps and complete the proofs where you see a red band at the left margin.
Introduction. Interpolation.

Problem 5.1 Neville's Algorithm

We want to interpolate the function $f(x) = \sqrt{x}$ at supporting points $x_0 = \frac{1}{4}$, $x_1 = 1$ and $x_2 = 4$ with a polynomial and evaluate it at $x^\star = 2$.

(5.1a) Compute the interpolating polynomial of minimum degree using the barycentric interpolation formula. Evaluate the polynomial at $x^\star$.

Solution: We first compute the coefficients λ_i, namely

Inserting those into the barycentric interpolation formula yields

Since there are many calculations needed to get to the result, one does usually not apply the barycentric interpolation formula. However, an evaluation at $x^\star = 2$ gives $p_n(2) = \frac{68}{45} \approx 1.5111$.

(5.1b) Evaluate the interpolating polynomial directly at $x^\star$ using Aitken's algorithm. Is there a difference compared to the value from subproblem (5.1a)? If yes, why?

Solution: Neville's algorithm can be used to evaluate p_n directly in $x^\star$:

Problem 5.2 Evaluating the Derivatives of Interpolating Polynomials

This problem is dedicated to fundamental algorithms for polynomial interpolation as discussed in Section 3.1 of the lecture. We will also learn about some of their extensions and applications. This problem involves substantial implementation in MATLAB.

(5.2a) Write a MATLAB function

```
p = AitNevpoleval(x,y,t)
```

that, using the Aitken-Neville scheme from [NMI, Thm. 3.7], evaluates the polynomial $p \in \mathbb{P}_n$ interpolating the data points (x_i, y_i), $i = 0, \ldots, n$, for pairwise different $x_i \in \mathbb{R}$ and data values $y_i \in \mathbb{R}$, in $t \in \mathbb{R}$. The data points are passed through the vectors `x` and `y`.

Solution: We implement the recursion formula for the Aitken-Neville scheme. The Algorithm reads:

(5.2b) Write an efficient MATLAB function

```
dp = dipoleval(x,y,t)
```

that returns the value $p'(t)$ of the *derivative* of the polynomial $p \in \mathbb{P}_n$ interpolating the data points (x_i, y_i), $i = 0, \ldots, n$, for pairwise different $x_i \in \mathbb{R}$ and data values $y_i \in \mathbb{R}$, evaluated at $t \in \mathbb{R}$.

HINT: Use the recursion formula for the Aitken-Neville scheme from [NMI, Thm. 3.7] and differentiate it.

Solution: Differentiating the recursion formula for the Aitken-Neville scheme, we obtain

(5.2c) Test the implementation of `dipoleval` in subproblem (5.2b) by comparing with the result obtained using simple *difference quotients* applied to the point values computed in subproblem (5.2a) through the routine `AitNevpoleval`. That is, we approximate

$$p'\left(\tfrac{1}{2}(t_i + t_{i+1})\right) \approx \frac{p(t_{i+1}) - p(t_i)}{t_{i+1} - t_i} \ .$$

Use $m = n + 1 = 10$ interpolation points `x = linspace(0,1,m)`, `y = rand(1,m)` and evaluation points `t = linspace(0,1,N)` for $N = 100$. Plot the results for the two implementations.

HINT: You may use the MATLAB command `diff`.

Problem 5.3 Hermite Interpolation

(5.3a) Let $p_n \in \mathbb{P}_n$ be the interpolating polynomial of degree at most n for the data points $\{(x_i, y_i)\}_{i=0}^{n} \subset \mathbb{R}^2$ with $x_i = x_j \Rightarrow i = j$, so in particular, we have $p_n(x_i) = y_i$ for $i = 0, \dots, n$. Show that p_n is given by

$$p_n(x) = \sum_{i=0}^{n} \frac{\omega_{n+1}(x) \cdot y_i}{(x - x_i) \cdot \omega'_{n+1}(x_i)}, \quad \text{where} \quad \omega_{n+1}(x) = \prod_{i=0}^{n} (x - x_i). \tag{5.3.1}$$

Solution: We know that the interpolation problem with $n + 1$ distinct points in the plane has a unique polynomial solution of degree at most n, so it is sufficient to show that the polynomial p_n given in 5.3.1 is of degree at most n and interpolates the given data points.

Looking at the degrees first, we see that

$$\frac{\omega_{n+1}(x)}{(x - x_i)} =$$

We now want to prove $p_n(x_j) = y_j$ for all $j \in \{0, \dots, n$. Using the product rule, we get the derivative

$$\omega'_{n+1}(x) =$$

(5.3b) Show that the interpolation mapping $I_n : \mathcal{C}^0([a,b]) \to \mathbb{P}_n,\ f \mapsto p_n$ on data points $a \leqslant x_0 < x_1 < \cdots < x_n \leqslant b$ is a linear map.

Solution: Consider two functions $f, g \in \mathcal{C}^0([a,b])$ and their interpolating polynomials $p_n = I_n(f)$ and $q_n = I_n(g)$ in $\mathbb{P}_n$. We want to show that for any $\alpha, \beta \in \mathbb{R}$,

$$I_n(\alpha f + \beta g) = \alpha p_n + \beta q_n.$$

(5.3c) Let $\mathbf{x} = (x_i)_{i=0}^n \in \mathbb{R}^{n+1}$. Computing the interpolating polynomial in the monomial basis $1, x, x^2, \ldots$, one will be confronted with the Vandermonde matrix $\mathbf{A} \in \mathbb{R}^{(n+1)\times(n+1)}$ given by $\mathbf{A}_{ij} = x_i^j$ for $i, j \in \{1, \ldots, n+1\}$. Show that

$$\det \mathbf{A} = \prod_{0 \leqslant i < k \leqslant n} (x_k - x_i).$$

When is the matrix $\mathbf{A}$ singular?

Solution:

Problem 5.4 Basis of Polynomials for Hermite Interpolation

Let $p_3 \in \mathbb{P}_3$ an interpolating polynomial that interpolates f and its first derivative at double data points $x_0 = x_1$ and $x_2 = x_3$ with values $f(x_0) = y_0$, $f'(x_0) = y_0'$ and $f(x_2) = y_2$, $f'(x_2) = y_2'$. From the theorem in [NMI, Sect. 3.5], we know that p_3 is unique. Determine four polynomials $h_i \in \mathbb{P}_3$, $i = 0, \ldots, 3$, such that p_3 can be written in the form

$$p_3(x) = y_0 h_0(x) + y_0' h_1(x) + y_2 h_2(x) + y_2' h_3(x).$$

Provide a general form for the polynomials h_i and sketch the graphs for $x_0 = 0$, $x_2 = 1$.

HINT: Think about conditions that have to hold true for the h_i. You can think of this as finding interpolation conditions for the h_i and solving the interpolation problem afterwards.

Problem 5.5 MATLAB: Lebesgue constant for interpolation

We want to study the influence of the interpolation points on the Lebesgue constant and on the error of the interpolation.

(5.5a) Write a MATLAB function `calcLambda(x)` that takes the interpolation points as a vector $\mathbf{x} = (x_0, \ldots, x_n)$ and returns an approximation of the Lebesgue constant Λ_n (see [NMI, Eq. 3.33]).

(5.5b) Compute an approximation of Λ_n as a function of $n = 1, \ldots, 20$ for

- equidistant data points in the interval $[-1, 1]$ using `linspace(-1,1,n+1)`,
- with Chebyshev nodes (see [NMI, Thm. 3.41]) as data points in the interval $[-1, 1]$,

and plot it in a single diagram with `semilogy`. Discuss your results.

(5.5c) Compute the approximate interpolation error $\|f - I_n[f]\|_\infty$ for $f(x) = (1 + a^2x^2)^{-1}$, $x \in [-1, 1]$ and plot it as a function of $n = 1, \ldots, 19$ for $a = 1, 5, 10$,

- in a diagram for equidistant data points,

- in a second diagram using Chebyshev nodes as data points.

In which case does the decrease of the best approximation error $\inf_{q \in \mathbb{P}_n} \|f - q\|_\infty$ compensate the increase of the Lebesgue constant with n? (see [NMI, Thm. 3.19])

Published on April 1, 2016.

To be submitted on April 12, 2016.

MATLAB: Submit all file in the online system. Include the files that generate the plots. Label all your plots. Include commands to run your functions. Comment on your results.

References

[NMI] Lecture Notes for the course "Numerische Mathematik I".

Last modified on April 1, 2016