Technische Universität Braunschweig
Institute for Partial Differential Equations
Prof. Dr. Carmen Gräßle
Jannis Marquardt

Summer term 2022

Numerical Methods for Differential Equations

Assignment 1

Upload solutions until 2 May 2022, 3pm

Task 1.1 (Interpolation, Runge function) **1+2+3=6 points**

Consider the Runge function $f : \mathbb{R} \to \mathbb{R}$, defined by

$$f(x) = \frac{1}{x^2+1},$$

from the first tutorial sheet.

(a) Create a new Matlab script-file and save it as `interpolation.m`.
Put the command `clear all` in the first line of your script in order to clear all variables from previous executions.

(b) Plot the Runge function for all $x \in [-3, 3]$. Add the title *"Runge function"* to the plot.

(c) Get familiar with the commands `polyfit` and `polyval`. Use them to find an interpolation polynomial of the Runge function for the following nodes:

1) `n=linspace(-2,2,3)` 2) `n=linspace(-2,2,5)` 3) `n=linspace(-2,2,7)`

Use polynomials of maximal possible degree such that they are uniquely defined. Plot the Runge function, together with these three interpolation polynomials, in a second figure with the title *"Polynomial interpolation of Runge function"*. Restrict the values of your plot in `x`-direction to the interval $[-3, 3]$ and in `y`-direction to the interval $[-1, 1.5]$.

Task 1.2 (Lagrange interpolation) **8 points**

Create a new file `lagrange_coeffs.m` and write a function which calculates the coefficients of the Lagrange interpolation polynomial for given data points `(xi,yi)`. The function should have the structure

```
function alfa = lagrange_coeffs(x,y)
```

where `x=(x0,...,xN)`, `y=(y0,...,yN)` are the inputs and `alfa = (alfa0,...,alfaN)` are the coefficients of the Lagrange interpolation polynomial.

Task 1.3 (Test of Lagrange interpolation) **3+3=6 points**

Think about two functions $f_1, f_2 : \mathbb{R} \to \mathbb{R}$ of your choice, for which you have calculated the Lagrange interpolation polynomials $p_1, p_2 : \mathbb{R} \to \mathbb{R}$ by hand (hint: you may use the examples from the lecture and tutorial class).

(a) Write a script called `test_lagrange.m` which compares the manually computed coefficients for $p_1(x)$ and $p_2(x)$ with the coefficients computed with `lagrange_coeffs(x,y)` and let the script print out whether the solutions coincide.

(b) Use `lagrange_coeffs(x,y)` to compute and plot the Lagrange polynomials of f_1 and f_2. This time, consider ten nodes which divide the interval $[0, 5]$ equidistantly.

Remark for all tasks: Make sure that your implementations are well structured and readable. A good way to ensure this is to use comments ("`%...`") in order to briefly explain the important steps in your code.