MA2501

NUMERICAL METHODS

NTNU, SPRING 2022

Project 2

- Project sets are to be handed in individually by each student using the OVSYS system[1]
- This part of the project counts for 8% of the final mark. The project is not compulsory.
- For the numerical part (Problem 1), you will submit a report in form of a Jupiter notebook file. This file will contain the code as well as explanations and comments in text of your findings and of your numerical results. Make sure that your code contains a meaningful documentation, in particular a reasonable number of comments to explain what is done. Also test the programs before submission and comment on all strange behaviour of the programs in your report.
- For the theoretical part (Problem 2), you will submit a report in form of a PDF file.
- When you present numerical results, these should be reproducible. This means that you will have to provide all relevant parameters.

You are asked to pay attention to the quality of presentation, in particular, the correctness of mathematical notation and formalism.

Deadline: 18-03-2022, 18h00 (OVSYS)

Problem 1. *We consider a continuous function $f : [-1, 1] \to \mathbb{R}$ which we want to approximate with different methods. Implement the Lagrange interpolation polynomial for an arbitrary set of distinct nodes $x_0, \ldots, x_n$ and for values $y_0, \ldots, y_n$. Test your code on equidistant nodes and on Chebyshev nodes considering first a function $f \in C^\infty$ defined on the interval $[-1, 1]$.*

Write your code as a Python function.

Input*:*

- *The data that we want to interpolate: x_i and y_i, $i = 0, \ldots, n$.*
- *The values of $z \in [-1, 1]$ where we want to evaluate the interpolation polynomial p_n (not interpolation points), this z can be an array.*

Output*: the value of the interpolation polynomial at z.*

Your answer to this problem should include a plot of the interpolation of the function $f(x) = \frac{\sin(\pi x)}{1+25x^2}$ with equidistant nodes and Chebishev nodes on the interval $[-1, 1]$ with $n = 15$. Try to calculate the error $\|f - p_n\|_\infty$ numerically by $\max_{j=0,\ldots,M} |f(z_j) - p_n(z_j)|$ using many points z_j, $n << M$, and plot how the error behaves against increasing n.

Date: February 25, 2022.

[1]A link can be found here: `https://wiki.math.ntnu.no/ma2501/2022v/start`

Problem 2. *In this problem we want to prove that the method of divided differences actually gives us the correct Newton form. Consider the Lagrange interpolation problem with given interpolation points $x_i, i = 0, \dots, n$ and we want to construct a polynomial $p_n \in P_n$ such that*

$$p_n(x_i) = f(x_i), \quad \text{for } i = 0, \dots, n,$$

for a given function f. The Newton form of polynomial p_n is given by the following form:

$$\begin{aligned} p_n(x) &= a_0 + a_1 (x - x_0) + a_2 (x - x_0)(x - x_1) + \cdots + a_n (x - x_0) \cdots (x - x_{n-1}) \\ &= \sum_{k=0}^{n-1} a_k \omega_k(x), \end{aligned} \tag{1}$$

where $\omega_k(x) = \prod_{j=0}^{k-1}(x - x_j)$. These coefficients a_k need to be determined from values of x_i and $f(x_i)$.

The following way of determining the coefficients a_k is called "divided differences". Define an operation $[x_0]f := f(x_0)$ and the following operations recursively for $k = 1, \dots, n$:

$$[x_0, x_1, \dots, x_k]\, f := \frac{[x_1, x_2, \dots, x_k]\, f - [x_0, x_1, \dots, x_{k-1}]\, f}{x_k - x_0}.$$

Note that on the left hand side, the number of arguments in the bracket $[\cdots]$ is $k + 1$ whereas on the right hand side the number of arguments is k. Then the coefficients are given by

$$a_k = [x_0, x_1, \dots, x_k]\, f, \text{ for } k = 0, \dots, n - 1. \tag{2}$$

2-a) We are going to prove the above way of calculating a_k gives us the correct Newton form, by induction. Assume that for some fixed $n = m$, the Newton form (1) with coefficients (2) solves the interpolation problem for any distinct interpolation point sets $(x_0, \dots, x_m)$ and any given function f, i.e., $p_m(x_i) = f(x_i)$ for $i = 0, \dots, m$. Using this induction hypothesis, prove the following: if we add one more point x_{m+1} for the Lagrange interpolation problem, then the solution p_{m+1} can be written as

$$p_{m+1}(x) = \frac{x_{m+1} - x}{x_{m+1} - x_0} p_m(x) + \frac{x - x_0}{x_{m+1} - x_0} q_m(x),$$

where $q_m \in P_m$ is the solution of the Lagrange interpolation problem with the interpolation points $(x_1, \dots, x_{m+1})$, and the leading coefficient[2] of p_{m+1} is given by

$$\frac{[x_1, x_2, \dots, x_{m+1}]\, f - [x_0, x_1, \dots, x_m]\, f}{x_m - x_0}.$$

2-b) Check if the induction hypothesis for $m = 0$ is true, and by the above argument, conclude that divided difference gives us the correct Newton form. Thus, the Newton form is useful when one wants to add interpolation points one by one recursively.

2-c) Prove that for any n the interpolation error can be written by

$$f(x) - p_n(x) = [x_0, \dots, x_n, x]\, f \omega_{n+1}(x).$$

Problem 3. *A Householder transformation is a matrix P of size $m \times m$ defined in terms of the vector $w \in \mathbb{R}^m$ with Euclidean norm $||w||_2 = 1$ and the $m \times m$ identity matrix I:*

$$P := I - 2ww^T.$$

Householder transformations can be used to compute the QR-decomposition of a matrix A by finding $P_1, \dots, P_k$ such that $P_k \cdots P_1 A$ is upper triangular. The successive multiplication with the P_is results in eliminating entries in the lower triangular part of A.

[2]The coefficient of the largest degree monomial in the polynomial.

Note: Problems 3-1 and 3-2 below are of theoretical nature and must be done by hand. Problem 3-3 should be submited in form of a Jupiter notebook file. This file will contain the code as well as explanations and comments in text of your findings and of your results. The code should be accompanied by a meaningful documentation including comments that explain what you are doing. Also test the program before submission and comment on all strange behaviour in your report (e.g., compare with numpy.linalg.qr).

3-1) Compute the eigenvalues, eigenvectors as well as the determinant of a Householder transformation. Provide detailed computations and arguments.

3-2) Use Householder transformation to compute the QR-decomposition of the matrix

$$\begin{pmatrix} 4 & -2 & 7 \\ 6 & 2 & -3 \\ 3 & 4 & 4 \end{pmatrix}$$

Provide detailed step-by-step computations.

3-3) The QR-decomposition can be used on non-square matrices. It results in a decomposition of an $m \times n$ matrix into an $m \times m$ orthogonal matrix and an $m \times n$ upper triangular matrix. Implement QR-decomposition for arbitrary matrices.