Lecturer: Barbara Re **TAs:** Davide Torlo, Stefano Colombo

Exercise session 4, 09 October, 2019

Numerical Methods in Informatics (L + E)

Exercise 1 DIY

Given the Runge's function $f(x) = \frac{1}{1+25x^2}$, for $-1 \leq x \leq 1$, compare the Lagrange interpolation of degree at most 4, 9, and 19 using $n+1$ equidistant points, and using $n+1$ Chebyshev nodes. Plot

- the three polynomials obtained with equidistant points and the function in a single figure,
- the three polynomials obtained with Chebyshev points and the function in a single figure,
- a figure for each degree $n = 4, 9, 19$ displaying the polynomial of degree n obtained with equidistant points, the one obtained with Chebyshev points, and the function

Remind: Chebyshev nodes on $x \in [-1, 1]$:

$$t_i = \cos\left(\frac{2i-1}{2n}\pi\right), \quad i = 1, \dots, n$$

Exercise 2

The price of a certain product has been recorded once per week for 5 weeks. With the abscissae measured in days, the following data were acquired: (0, 100), (7, 98), (14, 80), (21, 92),(28, 75), (35,70). Now, you want to predict the price for next week (e.g., for days 42). Plot the data and

- find a suitable polynomial interpolant to predict the required value (hint: look at the data behavior).
- predict also the value through the least-squares approximations of degree $m = 1$ and $m = 2$ for the given data set.
- compare the results.

Exercise 3

Implement and study the piecewise linear interpolation, a.k.a. first order splines.

- Build a function that performs the piecewise linear interpolation. It takes as inputs the points $\{(x_i, y_i)\}_{i=1}^N$ and some test points x and returns the piecewise reconstruction evaluated in points x, i.e., $\Pi_1(x)$.
- Test the algorithm on points generated by a function $f(x) = \texttt{sinc}(x)$ on the interval $[-5, 5]$, with different number N of equispaced points $\{x_i\}_{i=1}^N$, i.e., $\{(x_i, \texttt{sinc}(x_i))\}_{i=1}^N$.
- Find the order of convergence ($e \approx Ch^p$) of the algorithm computing the error of the reconstruction with respect to the exact function on a test set of points.

Exercise 4

The function `cubicspline.m` compute the cubic spline with a number of operations equal to the dimension of the system itself. The input parameters are the vectors `x` and `y` of the nodes and the data to interpolate, plus the vector `zi` of the abscissae where we want the spline s_3 to be evaluated. Unless otherwise specified,it computes the natural interpolation cubic spline. The optional parameters `type` and `der` (a vector with two components) serve the purpose of selecting other types of splines. With `type=0`, it computes the interpolating cubic spline whose first derivative is given by `der(1)` at x_0 and `der(2)` at x_n. With `type=1` we obtain the interpolating cubic spline whose values of the second derivative at the endpoints is given by `der(1)` at x_0 and `der(2)` at x_n.

- Test the different types of spline.
- Explore also the options offered by the MATLAB functions `spline` and `pchip`[1].

Exercise 5

Compare the order of convergence of the cubic splines (ex 4) with the linear splines (ex 3).

[1] See documentation: `https://ch.mathworks.com/help/matlab/ref/pchip.html`