Universität Duisburg-Essen
Computational Mechanics
Campus Essen

Dr. S. Vanis

Fall Term 2010/11

21. October 2010

Introduction to Numerical Methods
Tutorial 2

Exercise 1:
Let the following table of measurements be given.

i	x_i	y_i
0	-2	31
1	-1	5
2	1	1
3	2	11

Calculate the interpolating polynomial $p_3 \in \mathcal{P}_3$ using

(i) the method of Lagrange,

(ii) the method of Newton,

(iii) the standard form of the interpolation problem.

(*)-Exercise 2: (10 points)
Show that the linear system of equations arising from the standard form of the interpolation problem, cf. (2.1) in the lecture notes, can be solved uniquely for every $n \in \mathbb{N}$ when the nodes x_i are distinct, i.e., $x_i \neq x_j$ if $i \neq j \ \ \forall i, j = 0, \dots n$. This is to show that

$$\begin{pmatrix} 1 & 1 & \dots & \dots & 1 \\ x_0 & x_1 & \dots & \dots & x_n \\ x_0^2 & x_1^2 & \ddots & & x_n^2 \\ \vdots & \vdots & & \ddots & \vdots \\ x_0^n & x_1^n & \dots & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ y_n \end{pmatrix}$$

is uniquely solvable for all ordered sets of distinct nodes $(x_i)_{i=0,\dots,n}$ and associated values $(y_i)_{i=0,\dots,n}$.

Hint: You need some knowledge from Linear Algebra to show that.

Start with the special cases $n = 1, 2, 3$ and then generalize it to $n \in \mathbb{N}$.

Programming-Exercise 1: (delivery date: 4. November 2010, 15 points)
Program the method of Newton to solve

(i) the problem given in exercise 1,

(ii) the example of Runge, i.e., $f(x) = \frac{1}{1+25x^2}$, $x \in [-1,1] \subset \mathbb{R}$, with equidistant nodes for any arbitrary $n \in \mathbb{N}$. To test your source code try $n = 2, 4, 8, 10$ and compare the results with those presented in the lecture.

Your program should compute the **coefficients b_i** of the polynomial

$$\begin{aligned} p_n(x) &= b_0 + b_1(x-x_0) + b_2(x-x_0)(x-x_1) + \dots b_n(x-x_0)\dots(x-x_{n-1}) \\ &= \sum_{i=0}^{n} \left(b_i \prod_{j=0}^{i-1} (x-x_j) \right) \end{aligned}$$

and **plot the solution** on the interval $[-3,3] \subset \mathbb{R}$ for (i) and $[-1,1] \subset \mathbb{R}$ for (ii).
In both cases the **nodes** have to be **marked in the plots**.
For the example of Runge, the **original function** $f(x)$ has to be plotted in the diagram together with the approximation and the nodes.

Hint: For the example of Runge it would be best to write an m-file taking n as input variable.

Use the first part to see how to program the method of Newton and then generalize it for the second part.

Important: The **source code** which you have to send by e-Mail for all programming exercises has to be an **executable m-file**.

You have to send in a source code for **every programming exercise**. 50% of the points will not be sufficient if you have not worked on all of the programming exercises, cf. revised requirements on the homepage.

Delivery: Thursday, 28. October 2010, before the tutorial, for Exercise 1 and (∗)-Exercise 2.