Aschaffenburg University of Applied Sciences
Faculty for Engineering
Prof. Dr. Michael Möckel

Email: Michael.Moeckel@th-ab.de
Telephone: +49 (0)6021 / 4206 - 507

Numerical Methods and Simulation
Module MT 03
Winter term 2022/23

`https://moodle.th-ab.de/course/view.php?id=624`

Problem Set 7

Interpolation and Regression

Coursework

Exercise 1: Polynomial interpolation

Consider the following data points: (0,1), (1,2), (2,5), (3,10).

(1) Create two vectors x and y and plot the data points in MATLAB as circles using `plot()` with the option `'or'`.

(2) Assume that simple polynomials $g_i(x) = x^i$ form the set of basis functions. Set up the Vandermonde matrix by hand and determine the coefficients of the interpolating polynomial. Solve the system of coupled linear equations in MATLAB. Result: (1,0,1,0).

(3) How do you represent the corresponding polynomial $f(x)$ in MATLAB? Use the command `polyval(f, x)` to calculate an interpolated value at $x = 0.5$.

(4) Create the Vandermonde determinant using the `A = vander(x)` command in MATLAB and compare.

(5) Now create the interpolating polynomial based on the MATLAB routine `p = polyfit(x, y, n)`. n is the maximal order of the polynomial. Use `y = polyval(p,x)` to evaluate the polynomial fit. Plot the result.

(6) Write your own MATLAB function `Interpolation.m` which receives two vectors x and y and returns the coefficients of the interpolating polynomial. You can use the \ operator but no other high-level MATLAB routines. Test the routine!

Exercise 2: Interpolation based on high order polynomials

It can be a great joy for parents to see a baby growing. A young couple has documented the following growth of their child:

Month	0	1	2	3	4	5	6	7	8	9	10	11	12
Length	54	55	58	60	62	64	67	71	74	76	77	79	80

(1) What is a good time scale for creating a polynomial fit to the data? Hint: Create the Vandermonde matrix for different time units (days, months, years) and compare the condition number of the resulting matrices. Discuss the result.

(2) Write a MATLAB function `Extrapolation.m` which calls your polynomial interpolation `Interpolation.m` from exercise 1 but also calculates and returns an extrapolation to other given values outside of the data range.

(3) Test your routine for an extrapolation to the expected length at the age 15 months. Compare with the direct route based on the MATLAB commands for polynomial interpolation `p = polyfit(month, length, n)` and polynomial evalutaion `polyval(p,X)` for maximum n. Comment on the result!

(4) Try a polynomial extrapolation of lower order (e.g. n=2). What happens?

Exercise 3: Linear regression

Often, measured data points (x_i, y_i) come with some measurement error. A precise fitting of a complicated curve to the collected data is therefore meaningless (so-called overfitting). In such a case, *linear regression* provides a reasonable approach: A linear curve $y = a_1x + a_0$ is fitted by optimization on the parameters $a_0,\ a_1$ such that the error

$$E = \sum_{i=1}^{n} \left[y_i - (a_1x_i + a_0)\right]^2$$

defined as the sum of the squared difference between the data points and the curve at corresponding points becomes minimal (*least square* method).
Optimization implies that the partial derivatives w.r.t. the parameters a_0 and a_1 must vanish (minimum). This gives a system of two linear equations for the unknown parameters a_0 and a_1.

(1) Derive the coupled linear equations for the parameters a_1 and a_0.

(2) Introduce reasonable notation for the sums over data coordinates and solve the system for the optimal parameters.

(3) Write a MATLAB function `[a1, a0] = LinearRegression(x,y)` which accepts the vectors of the x and y coordinates of the data points as inputs. Implement the required sums.

Now the pressure (P) inside a bottle of Helium (He) gas is measured w.r.t. the temperature of the environment. The bottle is placed in a heat bath and different values for the temperature (T) are set. Assume that Helium is an ideal gas, that the volume (V) of the bottle does not change and that the ideal gas law can be applied:

$$\frac{PV}{T} = \text{const}$$

The following measurements have been taken:

T	0	10	20	30	40	50	60	70	80	90	100
P	0.94	0.96	1.0	1.05	1.07	1.09	1.14	1.17	1.21	1.24	1.28

(4) How is this problem related to linear regression?

(5) Plot the data points in a suitable plot using circles.

(6) Run your MATLAB function to obtain the linear regression parameters and write down the approximate functional dependence $P(T)$ for the Helium gas.

(7) Create a new plot for the data points and the linear function in the interval [-300, 100]. Study the plot and find a way for calculating the point of absolute zero temperature. What would happen there? Result: Absolute zero is at -274.15 Celsius. How close does your result get?

Exercise 4: MATLAB basic fitting user interface

MATLAB provides a basic user interface for data analysis. It can be called from a plot of data points.

(1) Create 11 datapoints with `x=0:10` and `y = exp(x./10) + 0.1*rand(1,11)` and plot the data.

(2) Go to the plot window, choose menu *Tools* → *Basic Fitting* and try different fittings. Plot the residua (errors) in a subplot.

(3) Expand the window twice using the arrow symbol at the bottom of the fitting interface. Export the coefficients to the workspace and copy them to a vector p.

(4) Return to the plot window and go to the menu *File* → *GenerateCode*.... Export the procedures which you have performed manually as MATLAB code and save it as a MATLAB function. Then close the plot and reproduce it by calling the function. This procedure can be used to design fitting frameworks which are applied to various data sets lateron.

(5) In this example the data was already sorted. As the fitting routines work more efficiently with sorted data it is recommended to sort larger data arrays before plotting them. Note that the sorting should be w.r.t. the x values but that the correspondance between pairs (x,y) must not get lost. This can be done by sorting x `[xsorted, perm] = sort(x)` and re-arranging y w.r.t. the permutation vector `perm` (or any other name) `ysorted = y(perm)`.