104283 – Introduction to Numerical Analysis

Spring Semester, 2023

Assignment 3

Polynomial Fit | Least Squares Method

Instructions:

1. Assignment is done in singles only and submitted through Moodle.
2. Submit the following two files ONLY:
   2.1. Report – including your code, results, derivations, explanations (if required), etc.
   Saved as a single *.pdf file: **report.pdf**
   2.2. Single .ZIP file with your Python codes.
3. Python files should be stored as a *.py files inside the zip file. Pay attention - your code should be included in the report as well (as image or text).
4. The zip file name format is as follows:
   4.1. HW
   4.2. Assignment number
   4.3. Underscore
   4.4. Student name (first and last name with no spaces, last name in CAPITAL letters)
   4.5. Underscore
   4.6. Student number

Example:
"HW3_harryPOTTER_999333666.zip"

5. The submission should include only the required submission files and nothing else. No subfolders or any unnecessary files should appear in the zip file. Do not use rar file or anything other than zip.
6. Submissions not following this format will not be accepted.
7. Late submissions policy – given n days late submission, $2n$ points reduction penalty.
8. Make sure you adhere to all the principles learned in class. Using specialized external Python libraries is not allowed (unless otherwise specified).

Least Squares Approximation

Write the following function in Python:

LS(xs, ys, degree)

Where xs is a list of nodes and ys is a list of functional values for these nodes.
Meaning, ys[k] = f(xs[k]), hence the length of these lists must be the same. Parameter degree specifies the degree of the least-squares polynomial to be computed.

The function constructs the least squares polynomial and returns the following:

- An n-array with the coefficients of the polynomial (of length degree+1)
- The total error.

The coefficients are computed from a linear system that comes from the so-called *normal equations*. To solve this linear system, you may use a Python library function or any other method.

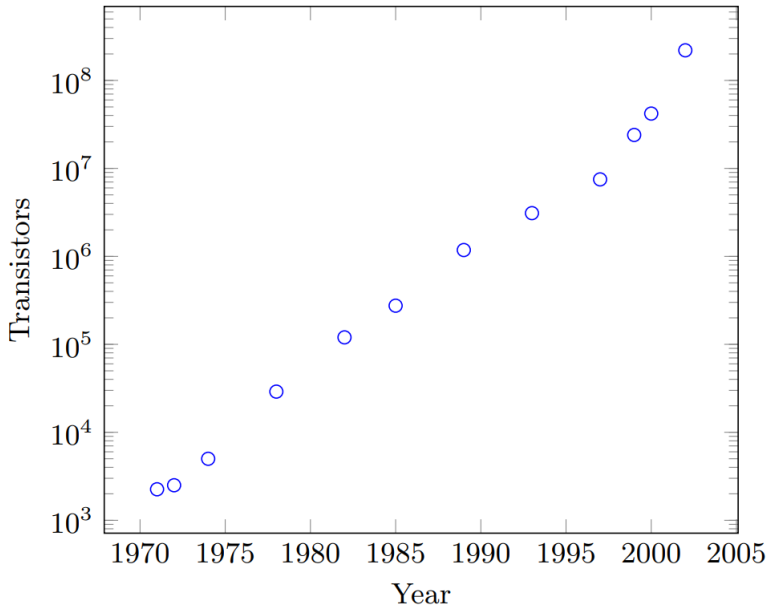
Test your function on the sample data set given in the file: *data_points.txt*.

Plot the given data points and the resulting polynomials with degrees 1, 2 and 3 on the same figure. Add a legend and appropriate title and axis labels to the figure.

Moore's Law Prediction

Moore's law was formulated by computer scientist and Intel co-founder Gordon Moore (1929-2023) in 1965. The law states that the number of transistors on microchips will roughly double every one and a half to two years.

The figure below shows the number of transistors N in 13 microprocessors, and the year of their introduction. Notice the plot gives the number of transistors on a **logarithmic** scale.



The attached file *moore.txt* includes this raw data. Find the least squares linear model for the data. Use your model to predict the number of transistors in a microprocessor introduced in the year 2023. Plot the results (raw data and model) and compare your calculated model to Moore's law.

Note

1. Python Appendix 3 covers all the necessary tools required for this assignment.
2. The so-called report is used mostly to present your code, results and plots. The answers to the additional questions should be brief, up to 1-2 paragraphs.

Deliverables:

1. PDF report including your code, plots and brief explanations. Attach LARGE images of the plots and code in the report.
2. ZIP file with your Python code.