Python Assignment 2

Polynomial Interpolation | Newton's 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:
"HW2_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).

Polynomial Interpolation

Given a set of distinct numbers $x_0, x_1, \dots, x_{n+1}$ and the values $f(x_0),\ f(x_1), \dots, f(x_{n+1})$ we want to compute the interpolating Lagrange polynomial P_n of degree n (page 108 Theorem 3.2).

Implement the following function in Python:

interpolate(points)

The function gets a sequence of points which represent the set of $x_i, f(x_i)$ pairs. The function computes and returns the corresponding Lagrange polynomial. The polynomial must be defined using symbolic variables.

Test your function to find the unique interpolating polynomial defined by the points:

$\boldsymbol{x}$	0	1	3	4	5
$\boldsymbol{f(x)}$	-1	-0.5	1	2	3

Display the resulting polynomial in the report.

Newton's Method

Implement the following function in Python:

newton(p0, f, tol, max_iter)

The function gets the following parameters as input:

- Parameter p0 is the initial guess.
- Some function f (defined symbolically).
- Convergence tolerance - tol.
- Maximum number of iterations - max_iter.

The function searches for a root of f(x) using Newton's method (page 83 eq. (2.13)). The function will output the solution for which $f(x)$ = 0 (if found) and the number of iterations used.

Use this function to find a real root of the polynomial found in the previous section. Add your results to the report.

Note:

1. Use tolerance of 10^{-5}.
2. Decide on an appropriate initial guess p_0 for the method.
3. You may use SymPy to compute derivatives as instructed in Python Appendix 2.

Deliverables:

1. PDF report including your code. Attach LARGE images.
2. ZIP file with your Python code.