Instructions:

- You must solve all the porblems.
- You can use library functions only when permitted.
- Your code will be subject to peer-reviews and AI based plagarism and AI generated content check.
- You can you C/C++, Python or Java.
- You must turn in the solutions via MS Teams within due date.
- The deadline will be followed very strictly.
- There should be 9 different source code files for all 9 problems.
- You can code the solutions of subproblems inside one single file.
- Do not zip all the files into one folder, but turn them in seperately
- You must click "*turn in"* after you have uploaded your solution, otherwise we wont grade your submission.
- Finally, please follow these instructions very strictly.

a) Suppose that you have a noisy speech signal $x[n]$ as follows

$$x[n] = 2\cos(2\pi n/100) + \eta[n], n = 0,1, \dots 99$$

Where $\eta[n]$ is random noise. You need to create a new speech signal $y[n]$ such that the noise is removed as effectively as possible. (You can use the library function eig, svd, inv, pinv, solve)

1) Generate and plot the noisy signal using scatter plot
2) Fit a 3rd degree polynomial curve to x using least squares.
3) Use a regularized least square with $\lambda = 100$.
4) Filter and plot the filtered signal $y[n]$ using continuous plot

b) The acceleration of a moving object with respect to time is given below (you cannot use any library other than Numpy. For all the problems bellow, you cannot hard code solution, that is, your solution should be able to handle larger data and segments)

t	0.5	1	1.5
$a(t)$	45	26	40

1) Fit a Lagrange polynomial to the data in the table above and plot the result.
2) Find the average velocity from $t_0 = 0.5$ to $t_2 = 1.5$ by integrating the data using a 2nd order polynomial with necessary number of segments.
3) Compare the accuracy of the results from 2 with an $n = 2$ segment 1st order integration technique.

c) The derivative of a function can be approximated as ($\Delta x = 0.05$)

$$f'(x) \approx \frac{-f(x+2\Delta x) + 8f(x+\Delta x) - 8f(x-\Delta x) + f(x-2\Delta x)}{12\Delta x}$$

With very high accuracy ($E = \mathcal{O}(\Delta x^4)$).

1) Compute the derivative of $f(x) = e^{-x^2}$ at $x = 1.75$ using the formula given above.
2) Minimize $f(x)$ with respect to x, using *gradient descent* as follows:

$$x_{k+1} = x_k - \frac{1}{k+1} f'(x_k)$$

Iterate for maximum 500 iterations or until $|f'(x_k)| < 0.005$.
Approximate $f'(x)$ using the formula given above.
3) Find the true percent relative error of approximating the derivative

d) The *Gaussian* function is defined as following

$$g(x) = e^{-x^2}$$

1) Scatter plot $g(x)$ for $x = -10, -9, -8, \ldots 50$
2) Fit a 3rd order polynomial regression curve. Use pinv (pseudoinverse)
3) Use a regularized least square with $\lambda = 100$. Use SVD
4) Plot the regression curve as a continuous plot.

e) The velocity of a moving object with respect to time is given as

$$v(t) = \ln(1+t)$$

(You cannot use any library other than Numpy. Your solution should be able to handle larger data and segments)

1) Fit a 4th degree Newton's divided difference polynomial to the data points
$\mathcal{D} = \{(0.5, v(0.5)), (1, v(1)), (1.5, v(1.5)), (2, v(2)), (2.5, v(2.5))\}$
2) Find distance traveled from $t_0 = 0.5$ to $t_4 = 2.5$ by integrating $v(t)$ using a 2nd order polynomial with necessary number of segments.
3) Compare the results from 2 with an $n = 4$ segments 1st order integration technique.

f) The partial derivative of a function can be approximated as ($\Delta x = 0.05, \Delta y = 0.075$)

$$\frac{\partial f}{\partial x} \approx \frac{f(x+\Delta x, y) - f(x,y)}{\Delta x}$$
$$\frac{\partial f}{\partial y} \approx \frac{f(x, y+\Delta y) - f(x,y)}{\Delta y}$$

1) Compute the partial derivatives of $f(x,y) = \exp(-\frac{x^2+y^2}{2})$ at $x = 1.75$ and $y = 0.27$ using the formula given above.
2) Minimize $f(x,y)$ with respect to x and y, using *gradient descent* as follows:

$$x_{k+1} = x_k - \frac{\partial f}{\partial x_k}$$
$$y_{k+1} = y_k - \frac{\partial f}{\partial y_k}$$

Iterate for maximum 500 iterations or the absolute value of gradients are less then 0.05. Start with $x_0 = y_0 = 0$. For approximating the partial derivatives, use the above-mentioned formula.

g) Suppose that you have a noisy speech signal $x[n]$ as follows

$$x[n] = 2\log(1+n) + \eta[n], n = 0,1, \dots 99$$

Where $\eta[n]$ is random noise. You need to create a new speech signal $y[n]$ such that the noise is removed as effectively as possible. (You can use the library function eig, svd, inv, pinv, solve)

1) Generate and plot the noisy signal using scatter plot
2) Fit a least squares regression curve using the basis expansion function $\phi(x) = (x^3, x^2, x, 1)$. Use SVD.
3) Use a regularized least square with $\lambda = 100$. Use SVD
4) Filter and plot the filtered signal $y[n]$ using continuous plot

h) The velocity of a moving object with respect to time is given as

$$v(t) = e^{0.05t}$$

(You cannot use any library other than Numpy. Your solution should be able to handle larger data and segments)

1) Fit a 4th degree Lagrange polynomial to the data points
   $\mathcal{D} = \{(0.5, v(0.5)), (1, v(1)), (1.5, v(1.5)), (2, v(2)), (2.5, v(2.5))\}$
2) Find distance traveled from $t_0 = 0.5$ to $t_4 = 2.5$ by integrating $v(t)$ using a 2nd order polynomial with $n = 3$ segments.
3) Compare the accuracy of the results from 2 with an $n = 2$ segment 1st order integration technique.

i) The derivative of a function can be approximated as ($\Delta x = 0.05$)

$$f'(x) \approx \frac{-f(x+2\Delta x) + 8f(x+\Delta x) - 8f(x-\Delta x) + f(x-2\Delta x)}{12\Delta x}$$

With very high accuracy ($E = \mathcal{O}(\Delta x^4)$).

1) Compute the derivative of $f(x) = e^{-x^2}$ at $x = 1.75$ using the formula given above.
2) Minimize $f(x)$ with respect to x, using *gradient descent* as follows:

$$x_{k+1} = x_k - \frac{1}{k+1} f'(x_k)$$

   Iterate for maximum 500 iterations or until $|f'(x_k)| < 0.005$.
   Approximate $f'(x)$ using the formula given above.
3) Find the true percent relative error of approximating the derivative