Question:	1	2	3	4	5	6	7	8	Total
Points:	15	10	10	10	10	25	10	10	100

Answer the questions in the spaces provided. If you need more space for your answer, continue on the back of the page. Show all your work and indicate your reasoning in order to receive the credit. Present your answers in *low-entropy* form.

1. Estimate the number of operations that are required to calculate a single value of an interpolating polynomial that passes through n points. The polynomial is given as a set of its coefficients $c_1, c_2, \ldots$. Do not assume that $n \gg 1$.

   (a) (10 points) Use Horner's method.

   (b) (5 points) Use the worst case "naive" method of calculations. For the purpose of this assignment assume that the "naive" calculation of x^n requires $n-1$ multiplications.

A polynomial that interpolates n points has the order $(n-1)$: $P(x) = C_1 x^{n-1} + C_2 x^{n-2} + \ldots + C_{n-1} x + C_n$.

(a) $P(x) = (\ldots((C_1 \cdot x + C_2)\cdot x + C_3)\cdot x + \ldots) x + C_n$; operations are (multiply by x, add) repeated $(n-1)$ times. Thus Horner's method requires $(n-1)$ multiplications and $(n-1)$ additions; $\boxed{2(n-1)}$ operations altogether.

(b) Calculation of $C_k x^{n-k}$ term requires $(n-k)$ multiplications. Total number of multiplications: $(n-1)+(n-2)+\ldots+1+0$. This is arithmetic progression. Its sum is $n(n-1)/2$. There is also $(n-1)$ additions:

$\frac{n(n-1)}{2}$ multiplication $+ (n-1)$ additions $-$ $\boxed{\frac{(n-1)(n+2)}{2}}$ operations in total

2. (10 points) Estimate the number of operations that are required to calculate the determinant of an $n \times n$ matrix using LU decomposition without pivoting. Assume that $n \gg 1$ and keep only the leading in n term in logarithmic precision.

Number of operations: $\sim n^3$ to factor a matrix plus $(n-1)$ operation to multiply the diagonal elements of U. In the large n limit $\sim \boxed{n^3}$ number of operations.

3. (10 points) You are solving a nonlinear equation $f(x) = 0$ using **bisection** and starting from the initial interval $[1, 2]$. Estimate the number of function evaluations that is required to achieve the error less than 2^{-24}.

The length of the initial interval is 1.
After the first bisection step the length is $\frac{1}{2}$
— // — second — // — $1/4$
— // — third — // — $1/8$
. . . .
— // — n-th — // — $1/2^n = 2^{-n}$
To have the error less than 2^{-24} we need more than 24 iterations

4. (10 points) A hypothetical algorithm for solution of nonlinear equation achieves the **cubic** convergence rate. During the testing you noticed that after the three initial iterations the error of the solution was 10^{-2}. What are the expected errors for the fourth and the fifth iterations? Explain.

Cubic convergence rate means that

$\frac{\varepsilon_{n+1}}{\varepsilon_n^3} \approx const$, where ε_n is the error after n iteration steps

We are given than $\varepsilon_3 = 10^{-2}$. Hence (ignoring the constant)

$\varepsilon_4 \approx \varepsilon_3^3 = \boxed{10^{-6}}$; $\varepsilon_5 \approx \varepsilon_4^3 = \boxed{10^{-18}}$

5. (10 points) Determine the interpolating polynomial, $P(x)$, that passes through the points $(0,1)$, $(1,4)$, $(2,9)$. Construct the Lagrange polynomials $L_i(x)$ and write down $P(x)$.

$$L_1(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} = \frac{(x-1)(x-2)}{(-1)(-2)} = \frac{1}{2}(x^2-3x+2)$$

$$L_2(x) = \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} = \frac{x(x-2)}{1\cdot(-1)} = 2x-x^2$$

$$L_3(x) = \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} = \frac{x(x-1)}{2\cdot 1} = \frac{1}{2}(x^2-x)$$

verify tha $L_i(x_i)=1, i=1,2,3$; $L_i(x_j)=0$, $j\neq i$

$$P(x) = L_1(x) + 4L_2(x) + 9L_3(x) = \frac{1}{2}(x^2-3x+2) - 4x^2 + 8x + \frac{9}{2}(x^2-x)$$

$$= \boxed{x^2+2x+1}$$ ← verify that $P(x_i)=y_i$

Matlab

6. The Airy function (or Airy function of the first kind), $Ai(x)$, is a special function named after the British astronomer George Airy (1801–1892). The function Ai(x) is a solutions to the differential equation

$$\frac{d^2y}{dx^2} - xy = 0.$$

The Airy function has many important applications: it provides uniform semiclassical approximations near a turning point of Schrdinger's equation; the Airy function describes the intensity near an optical directional caustic, such as that of the rainbow; the Airy function is also important in microscopy and astronomy: it describes the pattern, due to diffraction and interference, produced by a point source of light (the one which is much smaller than the resolution limit of a microscope or telescope).

Matlab's function `airy(x)` returns the value $Ai(x)$; the function `airy(1,x)` returns the value of the first derivative of Airy function.

Write a matlab script, **m2p6.m**, that produces the following:

(a) (5 points) Plot a graph of Airy function for $-10 \leq x \leq 0$. Provide the grid, a title, and axis labels. Use at least 200 points to produce the graph.

(b) (15 points) Using the matlab code that was developed in class for the bisection, the secant, and the Newton's methods, find the locations of two closest to zero negative roots of Airy function. Add your solutions to the plot. When finding the roots, use the tolerance `tol = 10*eps`.

(c) (5 points) Compare the performance of the three methods in terms of the number of function evaluations. Describe your observations in gitlab's README file.

7. (10 points) Write a matlab script, **m2p7.m**, that interpolates the function

$$f(x) = \frac{2}{1+16x^2}$$

using 16 points on the interval [-2, 2]. On the same graph plot the original function (using 100 points, graph as a solid blue line), your data points (as disconnected black symbols) and your interpolating polynomial (using 100 points, as a solid red line). Use the code for lagrange interpolation that we developed in class. Chose your data points to avoid the Runge phenomenon. Discuss your choice of the data points in your README.md file.

Git and Gitlab

8. (10 points) Upload the code you wrote for this exam:

   1. Use gitlab web interface to create a new project called **midterm2-sample** (the name is case sensitive, must be exactly as shown)
   2. Use gitlab web interface to create *README.md* file
   3. Use gitlab web interface to upload your matlab code to your project
   4. Use gitlab web interface to grant the access to your project (with the permission of the *reporter*) to the instructor.