Name: ____________________

Date: ____________________

Question:	1	2	3	4	5	6	Total
Points:	10	20	20	20	20	10	100
Score:							

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. The inverse of a matrix A can be calculated by finding the matrix whose columns x_j are solutions of the equations

$$Ax_j = e_j, \qquad j = 1, \ldots, n, \qquad (A)$$

where e_j is the jth column of the identity matrix.

(a) (5 points) **Estimate** the number of operations that are required to inverse $n \times n$ matrix, if (a) you solve n equations above n times using gaussian elimination without pivoting; (b) you LU-decompose once the matrix A; next two *triangular* systems of equations of order n are solved n times.

Assume that $n \gg 1$ and keep only the leading in n terms within the logarithmic precision.

(b) (5 points) It took about 0.2 seconds (on a slow computer) to inverse a 20×20 matrix using the method (b). **Estimate** how long it would take to invert a 100×100 matrix (on the same computer, using the same method).

Numerical derivatives

2. (a) (10 points) Determine the weights in the following *asymmetric* formula for the second derivative of a function:

$$\frac{\mathrm{d}^2 f}{\mathrm{d}x^2} = a\,f(x-h) + b\,f(x) + c\,f(x+h) + d\,f(x+2h) + e\,f(x+3f) + O(h^{\alpha}).$$

Use the Fornberg method. Show your work below. Use a computer algebra system to calculate the needed Taylor series coefficients.

(b) (10 points) Conduct numerical experiment to determine leading error term of your formula $\delta \sim h^{\alpha}$: Write a matlab script, **finp2** that for $h = 1, \frac{1}{\sqrt{2}}, \frac{1}{2}, \ldots, \frac{1}{2^7}$ calculates the absolute value of the error for the second derivative of $f(x) = log(x)$ at $x = 1$ when using your formula. On the same graph, in double logarithmic axes, plot the graph of the error vs. h. In addition, as a guide, plot the graphs of $y(h) = h^k$, for $k = 1, 2, 3, 4$. Analyze your figure and by visual inspection determine the constant α. Describe your results in gitlab's README.md file.

Gaussian quadrature

3. (a) (10 points) Find the nodes and the weights for the three point gaussian quadrature,

$$\int_{-1}^{1} \frac{f(x)\,\mathrm{d}x}{\sqrt{1-x^2}} = w_1 f(x_1) + w_2 f(x_2) + w_3 f(x_3), \qquad (B)$$

using **only** the requirements that your formula must produce exact answers for $f(x) = x^n$, $n = 0,\ 1,\ 2,\ 3,\ 4,$ and 5. Use the symmetry properties of the nodes and the weights of the gaussian quadrature.

You will need the values of the following integrals:

$$\int_{-1}^{1} \left(1-x^2\right)^{-\frac{1}{2}} \mathrm{d}x = \pi, \qquad \int_{-1}^{1} x^2 \left(1-x^2\right)^{-\frac{1}{2}} \mathrm{d}x = \frac{1}{2}\pi, \qquad \int_{-1}^{1} x^4 \left(1-x^2\right)^{-\frac{1}{2}} \mathrm{d}x = \frac{3}{8}\pi.$$

Show all your work below.

(b) (10 points) Write a matlab function, `fingaus3pt(f)`, that accepts a pointer to a function as the argument and implements your formula Eq. (B). Write a matlab script, **finp3** that verifies that your integration formula produces the exact answer for

$$f(x) = (1-x)^5.$$

Hint: use matlab function `quadgk` to find the "exact" value of the integral.

4. This problem deals with the investigation of errors of three different second order Runge Kutta methods for solving initial value problem:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t,y), \quad y(t_0) = y_0.$$

The classic Runge Kutta method, RK2, is given by the formula

$$y_{n+1} = y_n + \frac{k_1}{2} + \frac{k_2}{2}, \quad \text{where} \quad k_1 = hf(t_n, y_n), \quad k_2 = hf(t_n + h, y_n + k_1).$$

The midpoint method is given by the formula

$$y_{n+1} = y_n + k_2, \quad \text{where} \quad k_1 = hf(t_n, y_n), \quad k_2 = hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right).$$

The Ralston's method is given by the formula

$$y_{n+1} = y_n + \frac{k_1}{3} + \frac{2k_2}{3}, \quad \text{where} \quad k_1 = hf(t_n, y_n), \quad k_2 = hf\left(t_n + \frac{3h}{4}, y_n + \frac{3k_1}{4}\right).$$

(a) (10 points) Write three matlab functions implementing the stepping algorithms above, `classic((fun,h,t,y)`, `midpoint(fun,h,t,y)`, and `ralston(fun,h,t,y)`. Your functions accept the following input parameters: the pointer to the function $f(t,y)$ from the right hand side of the differential equation, the value of the step h, and the current values of t and $y(t)$.

(b) (10 points) For your numerical experiments use the following initial value problem:

$$\frac{\mathrm{d}y}{\mathrm{d}t} f = \frac{1}{y} + \frac{y}{2t}, \quad y(1) = 1, \quad 1 \le t \le 2,$$

that has the following exact analytic solution:

$$y(t) = \sqrt{t(2\log(t) + 1)}.$$

Write a matlab script, **finp4**, that for steps $h = (b-a)/2^{n+3}$, where $n = 1, 2, \ldots 8$, calculates the absolute value of the error of the solution, $y(2)$, produced by the three different stepping algorithms. On the same figure, in double logarithmic axes, plot the three graphs of the errors vs. h. Verify that the global error for all three algorithms is $O(h^2)$. Determine whether one of the algorithms is always better than the other two. Describe your results in gitlab's README.md file.

5. Boole's rule is a method of numerical integration that approximates an integral $\int_{x_1}^{x_5} f(x)\,\mathrm{d}x$ by using the values of the integrand at five equally spaced points x_1, $x_2 = x_1 + h$, $x_3 = x_1 + 2h$, $x_4 = x_1 + 3h$, $x_5 = x_1 + 4h$.

$$\int_{x_1}^{x_5} f(x)\,\mathrm{d}x = \frac{2h}{45}\Big(7f(x_1) + 32f(x_2) + 12f(x_3) + 32f(x_4) + 7f(x_5)\Big).$$

(a) (5 points) Derive the composite Boole's integration rule for n ($n = 4k + 1$, $k = 2, 3, \ldots$), equally spaced data points:

(b) (5 points) Write a matlab function, `finboo(h, f)`, that accept the value of the discretization step h and the array of function values $f(a)$, $f(a + h)$, $f(a + 2h)$, $\ldots$ $f(b)$ and return the approximate value of the integral using the composite Boole's rule.

(c) (10 points) Write a matlab script, **finp5**, that for $h = (b - a)/2^{n+1}$, where $n = 1, 2, \ldots 5$, calculates the absolute value of the error of the Boole's composite rule for the integral $\int_1^2 \sin(x)\,\mathrm{d}x$. Plot the graph of the error vs. h in double logarithmic axes. In addition, as guides, plot the graphs of $y(h) = h^k$, for $k = 4, 5, 6, 7$. Analyze your figure and by visual inspection determine order of the error of the composite Boole's. Describe your results in gitlab's README.md file.

Programming hints:

- To divide a segment $[a, b]$ into k subintervals of equal size, use the matlab function `linspace(a, b, k+1)`.
- To sum elements of an array `v`, use `sum()`. E.g., to sum every third element starting with the 4th element, use `sum(v(4:3:end))`.

Git and Gitlab

6. (10 points) Upload the code you wrote for this exam:

   1. Use gitlab web interface to create a new project called **final** (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.